Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 7513717, 15 pages
https://doi.org/10.1155/2022/7513717

Research Article

A Network Security Situation Prediction Method through the Use of Improved TCN and BiDLSTM

Chengpeng Yao, Yu Yang, Jinwei Yang, and Kun Yin

School of Information Engineering, Engineering University of PAP, Xi'an, Shaanxi 710000, China

Correspondence should be addressed to Yu Yang; miaoyude@163.com

Received 18 July 2022; Revised 12 September 2022; Accepted 23 September 2022; Published 5 October 2022

Academic Editor: Gengxin Sun

The rapid development of information technology has brought much convenience to human life, but more network threats have also come one after another. Network security situation prediction technology is an effective means to protect against network threats. Currently, the network environment is characterized by high data traffic and complex features, making it difficult to maintain the accuracy of the situation prediction. In this study, a network security situation prediction model based on attention mechanism (AM) improved temporal convolutional network (ATCN) combined with bidirectional long short-term memory (BiDLSTM) network is proposed. The TCN is improved by AM to extract the input temporal features, which has a more stable feature extraction capability compared with the traditional TCN and BiDLSTM, which is more capable of processing temporal data, and is used to perform the situation prediction. Finally, by validating on a real network traffic dataset, the proposed method has better performance on multiple loss functions and has more accurate and stable prediction results than TCN, BiDLSTM, TCN-LSTM, and other time-series prediction methods.

1. Introduction

The development of information technology has consistently promoted the progress of human society. With the deep development of artificial intelligence, big data, fifth-generation mobile communications, and other information technology, more network applications have played an essential role in the economic development of society, and the network is closely related to the national economy. However, the rich network applications also bring more opportunities for network threats to invade. Recently, network attacks with its hidden, fast, and automated characteristics, so that the network ecosystem suffered a severe impact. Various high-threat attacks such as distributed denial of service (DDoS) and ransomware attacks are more frequent. Although network security defense measures are progressing and developing, various security loopholes are being continuously investigated. Additionally, the existence of network attacks in the shadows is always defensible. Therefore, the network security issue has become an urgent problem in today's society, indicating the need to maintain network security effectively.

Network security situation awareness [1] was proposed in 1999 to reflect the overall network security situation by integrating data from network security protection devices, such as intrusion detection systems, firewalls, and virus detection systems (VDS) [2]. Compared with the traditional means of defense against network threats, network security situation awareness has the characteristics of more comprehensive detection, more active protection, and a faster response. Network security situation awareness is divided into situation element extraction, understanding, and prediction. Situation prediction is the last step of network security situation awareness and is also the ultimate purpose of situation awareness, and effective situation prediction is an essential means to prevent network threats.

There are many methods for network security situation prediction. The main research focuses on two aspects based on time-series prediction [3] and graph theory-based prediction [4]. The time-series prediction method is to take

advantage of the characteristics of network attacks with a certain periodicity (for example, more frequent attacks in certain periods). The periodic attacks make the network security situation with a certain periodical change consistent with the characteristics of time series. However, this method is more applicable to short-term situation prediction because the regularity of long-term posture is difficult to capture. The graph theory-based situation prediction method uses vulnerability information in the network environment to generate a state transfer graph to determine future attacks from the intruder's perspective [2]. However, this method suffers from a severe false alarm rate and insufficient prediction accuracy. In this study, based on the characteristics of the abovementioned methods, we employ a time-series prediction method to make short-term predictions of the network security situation. There are various approaches to time-series-based network security situation prediction. For instance, in parameter-based modeling, Yang et al. used adaptive cubic exponential smoothing for situation prediction [5], which is simple to model but unstable in prediction. Based on machine learning (ML) [6], Xing et al. and Wang Jian et al. used a support vector machine (SVM) for situation prediction [7, 8], which has a fast response time and a small model memory but a relatively low prediction accuracy. Based on deep learning (DL) [6], Wei et al. used gated recurrent unit (GRU) for situation prediction [9]; Chen et al. used long short-term memory (LSTM) for situation prediction [10]; and Guosheng et al. used backpropagation (BP) neural network for situation prediction [11]. Situation prediction using DL is relatively more complex and computationally intensive; however, it has higher accuracy. With the development of information technology, such as big data and cloud computing, a good platform for DL has been created. More data training and greater computing power support using DL for situation prediction have gradually become mainstream.

In the previous time series prediction methods, most of them are the prediction of a single model. In the face of complex and long-term time series characteristics, the prediction ability is insufficient. At this stage in the research study of situation prediction, more methods are used to combine the techniques of feature extraction and time series prediction. For instance, Shen and Wen [12] used a network security situation prediction method combining gray theory and BP neural network to enhance feature extraction. Liu et al. [13] proposed a network security situation prediction method combining TCN and LSTM to extract temporal features by TCN, while situation prediction is performed by LSTM later. The technology of combining feature extraction with time-series prediction has been studied to some extent, which makes up for the insufficient prediction ability of a single model. But nowadays the network environment is complex and changeable, and the network traffic is updated all the time, so the above research study needs to improve its feature extraction ability when dealing with temporal features, and there are more advanced prediction methods for prediction. To address the above issues, this study proposes a network security situation prediction method based on an attention mechanism (AM) improved temporal convolutional network (ATCN) combined with bidirectional long short-term memory (BiDLSTM) network. TCN is a variant of a convolutional neural network (CNN) [14]. Compared with the traditional convolution process, it has greater advantages in processing time series and AM is used to enhance its ability to extract important features of images when it is proposed. Similarly, it can find more important features in sequences. BiDLSTM is composed of two layers of LSTM with different input directions. Compared with LSTM, it has a stronger long-term and short-term prediction ability by combining the three models of AM, TCN, and BiDLSTM to achieve better situation prediction. Finally, the proposed method is validated on a real network traffic dataset. This study has the following contributions:

(1) Given the insufficient prediction ability of a single model on the network security situation, in this study, we propose a model integrating ATCN and BiDLSTM for network security situation prediction. It is an end-to-end model. ATCN is used as a feature extraction tool and BiDLSTM is used as a prediction tool. The prediction is carried out by combining the two models. By combining the feature extraction model with the prediction model, the model has more advantages compared with the single model in feature extraction and prediction ability of sequence data. By using better models for combination and by the use of AM, the hybrid model has a better prediction effect than other hybrid models such as TCN-LSTM.

(2) The improved TCN is used to extract the feature of time series, and the AM is used before each dilated causal convolutional layer in the TCN structure, which has a more stable feature extraction ability.

(3) Through BiDLSTM for situation prediction: BiDLSTM has excellent long-distance feature extraction ability, and its prediction ability is stronger than LSTM, GRU, and TCN models.

(4) By validating the model on China Internet Emergency Response Center's Cybersecurity Information and Dynamics Weekly Report Dataset, the proposed model has more accurate and stable prediction results compared to other single models and hybrid models time-series prediction methods, and has better performance in the root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

This study consists of the following sections. Section 2 describes the related work on time-series prediction. Section 3 describes TCN, AM, BiDLSTM, and overall model structure. It also presents a brief description of datasets and evaluation metrics. Section 4 conducts experiments and analyzes the prediction results for the proposed model. Finally, Section 5 provides the summary and outlook for future work.

2. Related Work

As the name implies, time-series prediction is based on chronological order by learning information from a past

period to make predictions about future periods. Because time series has a backward and forward time causality, it has strict requirements on the backward and forward order of the inputs. The current time-series prediction covers a wide range of fields, such as energy wind speed prediction [15], infectious disease prediction [16], water quantity prediction [17], population prediction [18], and stock prediction [19].

Time-series prediction methods have evolved from traditional parametric modeling prediction and time regression prediction to ML and DL. However, most traditional methods have simple models and cannot balance spatial and temporal correlation [20]. At this stage, time-series prediction methods mainly focus on ML and DL. The time-series prediction methods based on ML include SVM [7, 8], random forest [21], and LightGBM [22]. The random forest and LightGBM methods are derived based on the regression tree [23] algorithm. As a classical ML algorithm, the regression tree algorithm has the advantages of easy construction and fast speed. However, as the volume of data becomes larger and the number of data dimensions increases at this stage, the regression tree also begins to be less stable, and the prediction effect in some complex situations becomes less satisfactory. Time-series prediction methods based on DL have developed rapidly in recent years. The most common ones that deal with time-series problems are recurrent neural networks (RNNs) [24] and their variants LSTM and GRU. RNNs, LSTMs, and GRUs have a memory of previously processed sequences when processing sequences, a feature that makes them well-suitable for applications in time-series prediction. BiDLSTM is obtained by improving on LSTM, and temporal prediction by BiDLSTM has been studied in many aspects, such as Mikhailov and Kashevnik [25] predicted car tourist trajectory by BiDLSTM; Mao et al. [26] predicted depression level by BiDLSTM and time distributed CNN; and Kang et al. [27] performed sewage flow prediction by BiDLSTM. It has been demonstrated through experimental studies that BiDLSTM has a more stable prediction effect on timing prediction compared to LSTM. TCN was proposed in 2018, which has a more flexible perceptual field mechanism with more stable gradients than RNN, a traditional method for processing time series, and it combines the features of CNN and RNN, which is well suited for feature extraction of time-series data. TCN has been widely used in time-series prediction in recent years. For instance, Wang et al. [28] used TCN and LightGBM for electrical load predictions, and feature extraction of multiple long-term sequences was performed by TCN. Menegozzo et al. [29] used an improved TCN to enhance the feature extraction capability for food production prediction. In this study, we take advantage of the excellent feature extraction capability of TCN to facilitate model building.

The AM has been a hot research topic in recent years, and the combination of AM and neural networks is also the mainstream of research studies. For instance, Pei et al. [30] combined AM and RNN to predict health records. Majid et al. [31] combined AM and convolutional neural networks (CNNs) for fire detection. The combination of AM and neural networks has achieved good results. However, in practice, it is found that a single AM is unstable in helping sequence feature extraction. Therefore, this study improves the TCN by using the AM both inside and outside the structure of the TCN, so that the improved TCN has a stronger feature extraction ability to help the model learn the features of the time series.

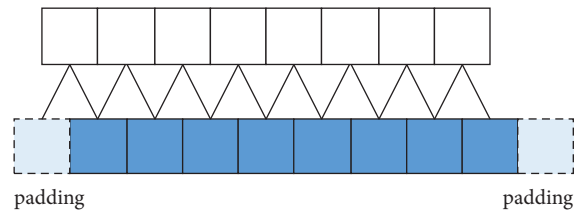

Figure 1: Sequence filling.

3. Methodology

3.1. Temporal Convolutional Network. Temporal convolutional network (TCN) [32] was proposed by Shaojie Bai et al., which is based on CNN and is designed to deal with time-series problems. It implements the processing of time-series problems through three structures: causal convolution, dilated convolution, and residual connections. Each part has the following structure.

3.2. Sequence Model. For time-series problems, the output sequence (y_0, y_1, ..., and y_T) must have the same length when the sequence (x_0, x_1, ..., and x_T) is the input, and the TCN is implemented using a one-dimensional full convolutional network (FCN) [33]. FCN ensures that each convolutional layer has the same time step length by employing the padding method for each layer of sequence padding. Figure 1 shows that when kernel size is 2, at a padding of 1, a padding is added to each end of the sequence, and the right padding is removed, making the length between sequences the same by using the following padding formula in which dilation has also described.

$$\text{Padding} = (\text{Kernel\_size} - 1) * \text{dilation}. \quad (1)$$

3.3. Causal Convolution. The input of traditional CNN has no time order, and the information before and after is acquired simultaneously, leading to future information leakage for time series. Meanwhile, causal convolution can be used to solve this problem. Figure 2 shows the causal convolution. As shown in this figure, the design of the causal convolution is unidirectional. The output y_t is only related to the inputs (x_0, x_1, ..., and x_t) at a moment t and before a moment t by an unidirectional design according to the temporal order. The output of the next layer at a moment t is obtained from the input of the previous layer at a moment t and the input before a moment t. This design makes the increase in the perceptual field very slow. When dealing with time-series problems, a large field of view is often required to learn the information of a long-time period. This can only be achieved

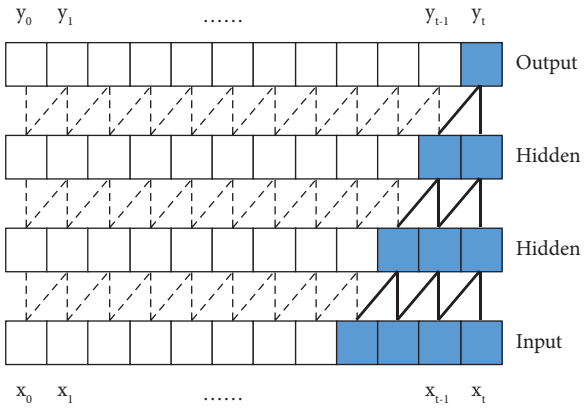

Figure 2: Causal convolution structure.

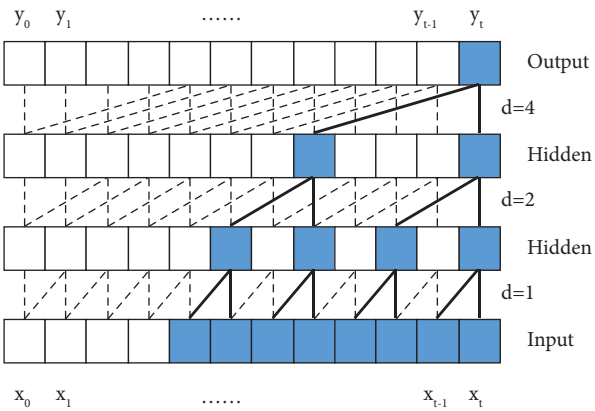

Figure 3: Dilated convolution structure.

by accumulating the number of hidden layers or a larger filter and by making the training more complicated. The accumulation of layers will also bring the hidden problem of gradient disappearance. Therefore, to solve these problems, the dilated convolution method is introduced.

3.4. Dilated Convolution. The dilated convolution method is used to solve the problem of the restricted field of view of causal convolution. Compared with causal convolution, dilated convolution introduces the concept of a dilation factor. Dilated convolution takes interval sampling in each layer for convolution sampling. The size of the interval is determined by the dilation factor d, as shown in Figure 3, where the kernel size is 2 and dilations are [1, 2, 4]. The size of each layer d of the dilated convolution grows exponentially, where the first layer is 1 (1 means the interval is 0). As shown in the figure, the field of view is increased from 4 to 8 in Figure 1 by three convolutions. By the exponential growth of the dilation factor d, an inflated convolutional network with stacking can operate over a larger field of view without loss of resolution or coverage [34]. For an input sequence, the expansion convolution of the sequence element O is given by the following equation, where k is the kernel size, and $O - d \cdot n$ represents the past direction.

$$F(O) = (x *_d f)(O) = \sum_{n=0}^{k-1} f(n) \cdot x_{O-d \cdot n}. \tag{2}$$

3.5. Residual Connections. In practical applications, the number of hidden layers is deepened to make the model more expressive. However, the gradient disappears for too deep networks. Thus, to solve this problem, residual connections are introduced. Residual connections avoid the problem of gradient disappearance by carrying short paths of gradients over a very deep network range [35]. In other words, information from the bottom layer can be passed directly to the top layer to avoid degradation of the model's learning ability and thus this makes the model more generalizable.

Figure 4 shows the output of the residual block which is obtained by adding $F(x)$ after a series of transformations and by making a convolutional mapping of the input x. The equation is given as follows.The residual block consists of the dilated causal convolution layer, normalization layer, activation layer, and dropout. The normalization layer is used to limit the distribution of the inputs, inorder to avoid gradient saturation with faster convergence. Then, the activation function allows the model to learn more nonlinear features. Meanwhile, the linear ReLU activation function does not have the problem of gradient explosion and is suitable for multilayer network structures. Finally, dropout is used to prevent overfitting.

$$o = \text{Activation}(x + F(x)). \tag{3}$$

3.6. Attention Mechanism. The AM [36] was first applied in computer vision [37]. Its essence is derived from the attention of human vision, which finds more important parts of a picture by paying more attention to it. When humans scan a group of things visually, they usually find the most noteworthy point after the first observation, devote more attention resources to it, and ignore the other information; thus, improving the efficiency and accuracy of observing things. The AM of the computer takes advantage of the characteristics of human attention by adding different weights to different features according to their degree of importance after observing the desired information; thus, achieving more attention to important features.

There are various categories of AM, such as Bahdanau attention [38] and Luong attention [39]. Although there are many variants of AM, the main difference is their locations and uses. Therefore, this study uses the AM before each dilated causal convolution layer to calculate all time steps of the input. The weights of each time step are generated using the softmax function, and the weights are matrix multiplied by the time steps to obtain the input of the next layer, as shown in Figure 5. The degree of importance of each time step is determined by calculating the magnitude of the weights for each time step for feature learning in the dilated causal convolution layer.

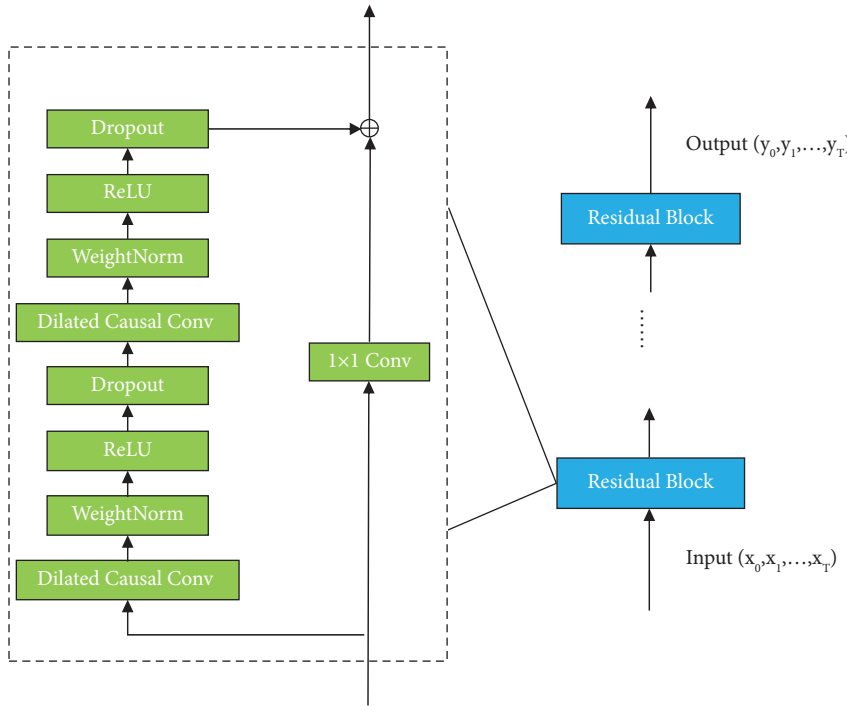

Figure 4: Residual connections structure.

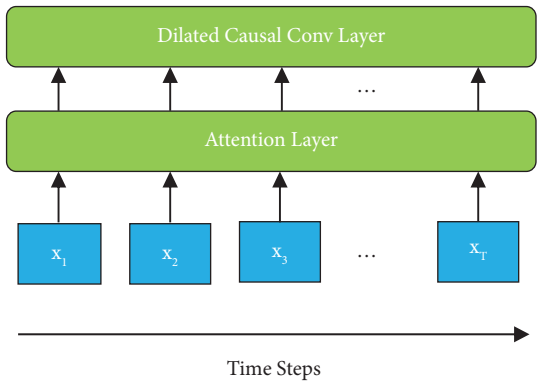

Figure 5: Structure of AM.

3.7. ATCN. Figure 6 shows the model based on AM and TCN. The main idea is to help the dilated causal convolution layer to better extract features by introducing AM. An attention layer is used before each dilated causal convolution layer in the residual structure to find the important difference between the input data through the attention layer. Sequences are weighted once by the AM before being input into the dilated causal convolution layer, which enables the AM to differentiate the importance of sequence data after each hidden layer processing so that the dilated causal convolution layer can perform better feature learning. The improved TCN module performs feature extraction of the time-series data and inputs the learned feature relationships between sequences into the next layer. ATCN is equivalent to the function of the encoder as a whole. After effectively learning the sequence features, it is input into BiDLSTM for decoding prediction.

3.8. Bidirectional Long Short-Term Memory. Bidirectional long short-term memory (BiDLSTM) [40] is generated based on LSTM [41], which consists of two layers of LSTM, one layer processing the original forward input data and one layer processing the reverse input data, and finally, the output data is obtained by combining the data of the two layers. BiDLSTM can effectively solve the problem of gradient disappearance in standard RNN by bidirectional design [42], and the bidirectional design is also more helpful for the extraction of input features.

Figures 7 and 8 show the structural diagrams of LSTM and BiDLSTM. LSTM is composed of three gate structures: forgetting, input, and output. The forgetting gate deletes the information that does not continue to be transmitted, the input gate inputs current information, and the output gate outputs the current phase information and hidden information passed onto the next phase, and the long-distance memory function is realized by the three gate structures. BiDLSTM is composed of two layers of LSTM with different input directions, the original direction of the lower input time series and the opposite direction of the upper input time series, to better extract the temporal features of the series and to obtain better prediction results.

3.9. Model Structure. Figure 9 shows the overall structure of the model. It consists of an input layer, an ATCN layer, a BiDLSTM layer, a fully-connected layer, and an output layer. This is an end-to-end model, which input data through the input layer, the ATCN layer, the BiDLSTM layer, and the fully-connected layer for data processing, and the final output layer output prediction results. The ATCN layer is the feature extraction module and the BiDLSTM layer is the

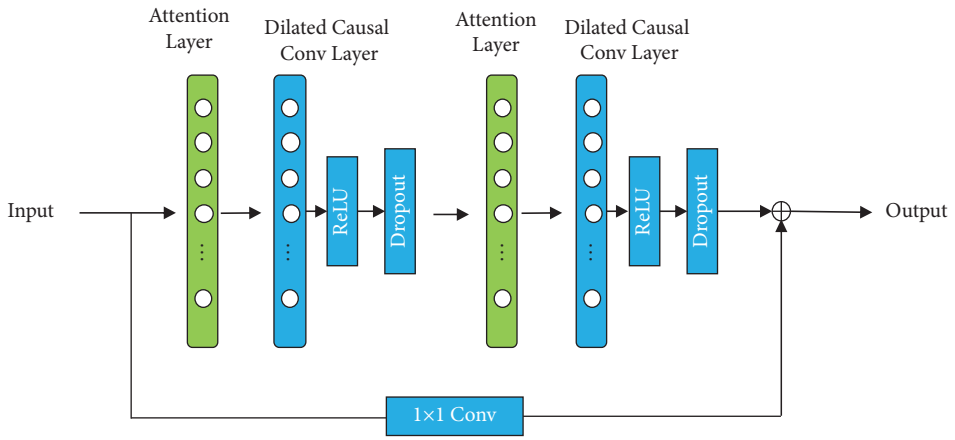

Figure 6: ATCN model structure.

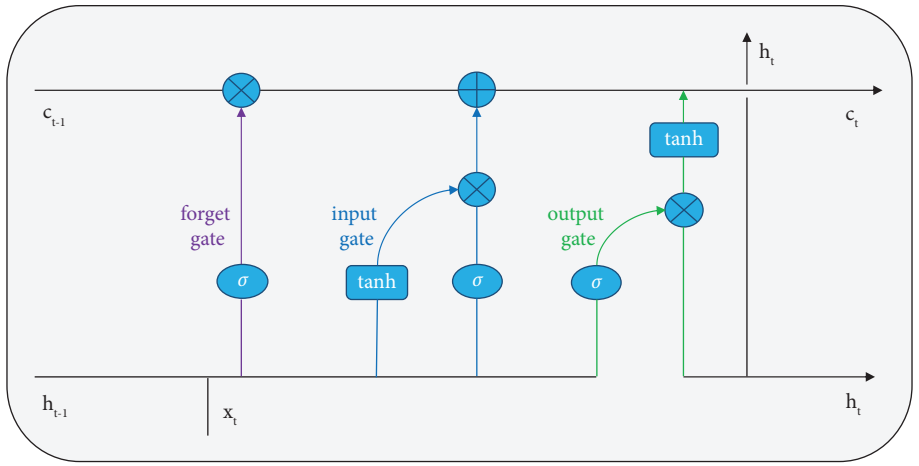

Figure 7: LSTM structure.

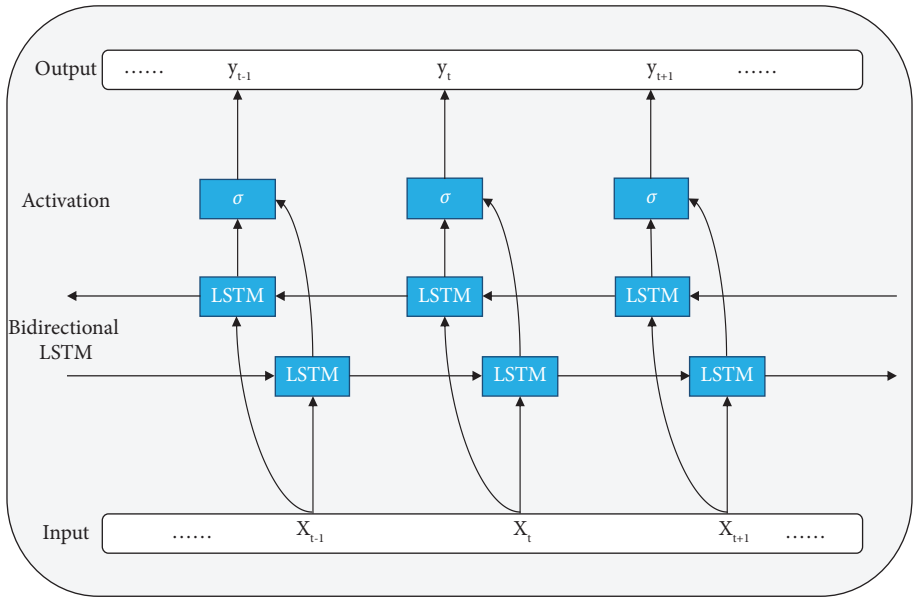

Figure 8: BiDLSTM structure.

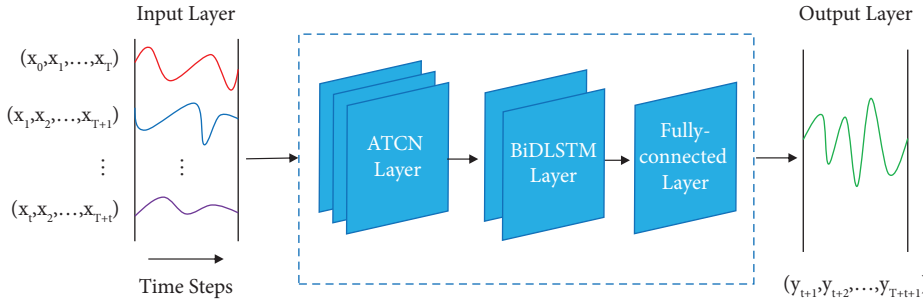

Figure 9: Overall model structure.

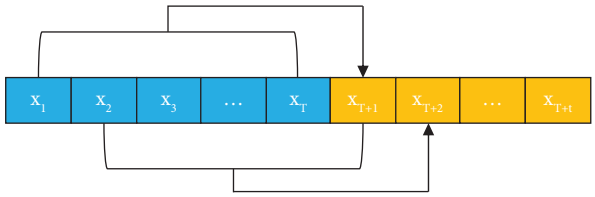

Figure 10: Prediction method.

prediction module. This combined feature extraction module and prediction module have stronger feature extraction and prediction ability than a single model, and the use of AM further strengthens the temporal feature extraction ability of TCN. The input layer inputs continuous time-series data with a fixed period T. The time-series data are a three-dimensional array consisting of sample size, time step length, and feature dimension. The sample size represents the number of input samples; the time step indicates the number of time steps through which the prediction is performed; and the feature dimension indicates the number of features for each time step. For univariate time-series prediction, the feature dimension is 1, and only the values before this variable are used to predict the values after it. The ATCN layer is responsible for extracting and learning the temporal features of the input sequence and feeding the learned features to the next layer. The BiDLSTM layer is responsible for carrying out the prediction work. A single-step prediction method is used, and only the situation value of the previous period is used to predict the next situation value each time. We use the form of a sliding window for sliding prediction, as shown in Figure 10. By setting the sliding time window to 1, the value of the first time step to the T time step is used to predict the value of the $T+1$ time step, and the value of the second time step to the $T+1$ time step is used to predict the value of the $T+2$ time step, and then pushing it down in turn until all the situation values are predicted.

3.10. Dataset Description. In this study, we validate the model of China's Internet Emergency Response Center's Cybersecurity Information and Dynamics Weekly Report Dataset [43]. The dataset was divided into two segments, dataset 1 and dataset 2. Dataset 1 was selected from the 1st issue of 2010 to the 13th issue of 2012, for a total of 115 weeks. Dataset 2 was selected from the 32nd issue of 2017 to the 1st issue of 2022, for a total of 231 weeks. There were three characteristic indicators in dataset 1, which are the number of hosts controlled by Trojan or bot programs in the territory, the number of government websites tampered within the territory, and the number of new security vulnerabilities. Table 1 presents the data values for five weeks from 9nd to 13th issues in 2012. There are five characteristic indicators in dataset 2, which are the number of hosts infected with malicious computer programs in the territory, the total number of URLs tampered within the territory, the total number of websites implanted with backdoors in the territory, the number of counterfeit pages targeting websites in the territory, and the number of new information security vulnerabilities. Table 2 presents the data values for five weeks from 32nd to 36th issues in 2017.

The weekly situation values were calculated from feature indicators using the situation assessment method in the literature [44]. Each featured category was assigned a different weight according to the threat level, as presented in Tables 1 and 2. Then, according to the following equation the weekly posture values were calculated. Here i is the feature category; n is the number of features; M_i is the value of this feature; $M_{i\max}$ is the maximum value of this feature in all weeks; and ω_i is the feature weights. In this calculation, due to the lack of the characteristic indicator for the number of new security vulnerabilities from the 1st issue of 2010 to the 22nd issue of 2010, we used the average of the 23rd issue of 2010 to the 48th issue of 2010 for filling.

$$\mathrm{SV} = \sum_{i=1}^{n} \frac{M_i}{M_{i\max}} \cdot \omega_i. \tag{4}$$

The calculated situation values are shown in Figures 11 and 12. Dataset 1 takes the first 92 weeks as training data and 93–115 weeks as testing data. Dataset 2 takes the first 184 weeks as training data and 185–231 weeks as testing data. It can be seen that the situation values of dataset 1 show the cyclic movement of up-and-down with the characteristics of a time series. After one large fluctuation at 100 weeks, the overall situation values of dataset 2 show a cyclic movement with the characteristics of a time series.

3.11. Evaluation Metrics. In the experiments, the mean squared error (MSE) loss function is used to evaluate the prediction results in training. Three-loss functions, RMSE,

Table 1: Indicator Characteristics for dataset 1.

Weeks	Characteristics		
	Trojans and bots (million)	Tampered government websites	Security vulnerabilities
1	29.9	39	107
2	21.7	28	127
3	15.7	105	162
4	14.8	48	165
5	12.8	32	160
Weights	0.5	0.3	0.2

Table 2: Indicator Characteristics for dataset 2.

Weeks	Characteristics				
	Malicious programs (million)	Tampered URL	Implanted backdoor websites	Counterfeit pages	Security vulnerabilities
1	41.1	2094	1527	351	212
2	54.6	2692	927	117	302
3	56.8	2338	1032	440	397
4	57.3	2184	1109	325	292
5	56.1	2444	961	306	348
Weights	0.3	0.25	0.15	0.15	0.15

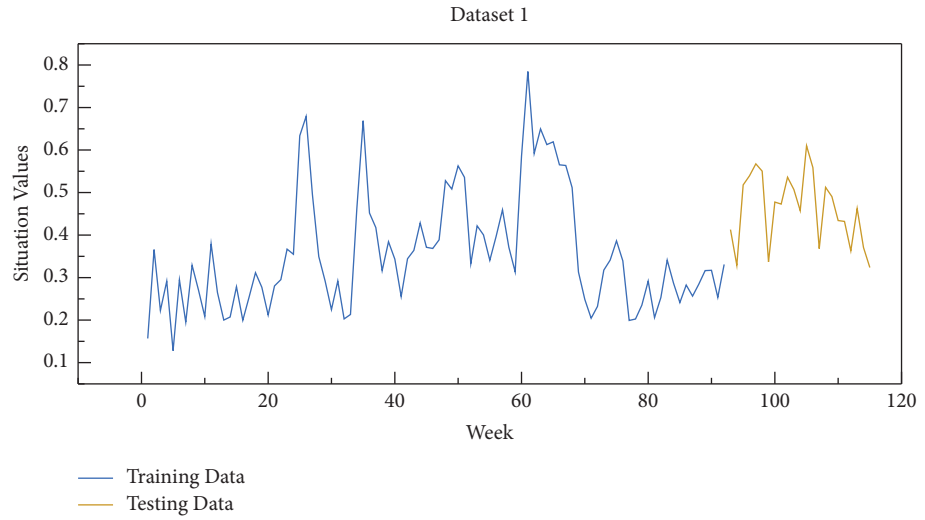

Figure 11: Dataset 1 situation values.

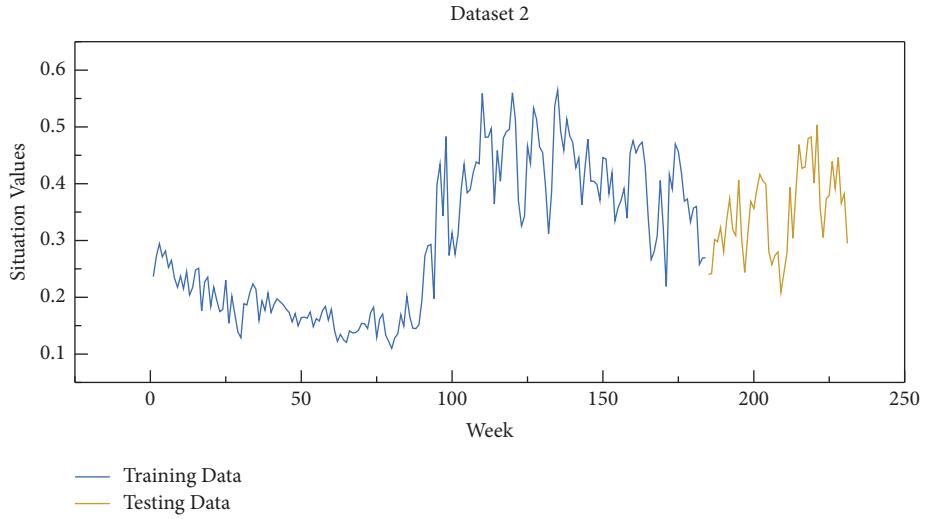

Figure 12: Dataset 2 situation values.

MAE, and MAPE, are used to evaluate the prediction results in testing. In the following equations,n is the total number of experiments, $\hat{v}_i$ is the predicted value, and v_i is the true value.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{v}_i - v_i\right)^2, \tag{5}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{v}_i - v_i\right)^2}, \tag{6}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{v}_i - v_i\right|, \tag{7}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{v}_i - v_i}{v_i}\right|. \tag{8}$$

In the test, three metrics are used to balance the advantages and disadvantages between them. RMSE evaluates smooth results but is more sensitive to outliers, and its value is influenced by a single outlier. MAE solves the outlier sensitivity problem, but the function may not be derivable at some points because of the existence of absolute values. MAPE is robust, but the prediction is more biased to models with positive errors and its evaluation index will be worse for negative errors, especially where the predicted value is higher than the true value.

4. Experiment and Results

4.1. Implementation. The experiment was implemented on a personal host with Intel core i5 10600 KF CPU and NVIDIA RTX2060 GPU, using python programming language and building models for implementation using TensorFlow and Keras methods. The detailed data are presented in Table 3.

We selected eleven methods, support vector regression (SVR), BiDLSTM, LSTM, TCN, GRU, TCN-LSTM, TCN-GRU, TCN-BiDLSTM, TCN-BiDGRU, ATCN, and ATCN-LSTM, for comparison experiments with ATCN-BiDLSTM. These include ML models, single DL models, and hybrid DL models. At the same time, the average value of the five experiments is taken to avoid the influence of error.

4.2. Metrics Analysis

4.2.1. Dataset 1. Dataset 1 was selected from the 1st issue of 2010 to the 13th issue of 2012, for a total of 115 weeks. For the selection of time steps, 6 weeks were taken as one cycle. Each time, the sliding prediction was achieved by predicting the next week's situation value by the previous 6 weeks' situation value. The parameters of each model are adjusted by many experiments, as follows.

(1) SVR: use the linear kernel function and set the penalty factor to 1.

(2) BiDLSTM: the two hidden layers have 32 nodes, respectively. The four fully-connected layers have 64, 32, 16, and 1 node, respectively.

Table 3: Experimental operating environment.

Project	Environmental parameters
CPU	Intel core i5 10600 KF
GPU	NVIDIA RTX2060
Python version	3.6
TensorFlow version	1.14
Keras version	2.2.5

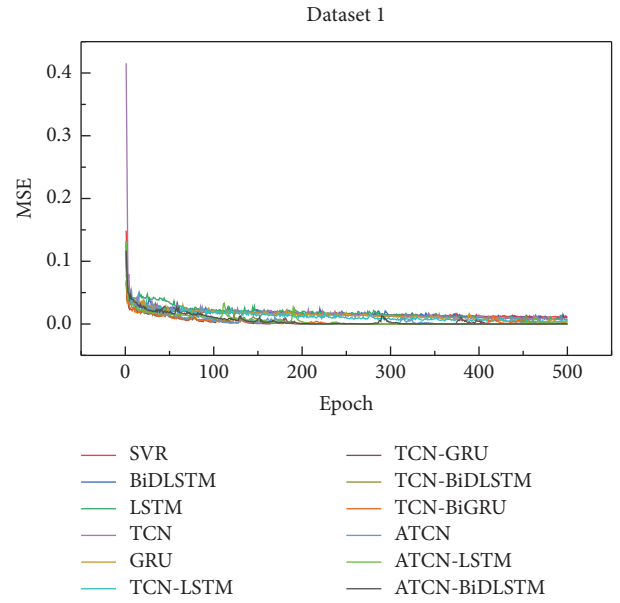

Figure 13: Training convergence curves for each model on dataset 1.

(3) LSTM and GRU: the three hidden layers have 32 nodes, respectively. The four fully-connected layers have 64, 32, 16, and 1 node, respectively.

(4) TCN and ATCN: the number of filters is 4 and the size is 3, the dilation factor is (1, 2, 4, and 8), and the residual connection layers is 1, the four fully-connected layers have 64, 32, 16, and 1 node, respectively.

(5) TCN-LSTM, TCN-GRU, and ATCN-LSTM: the number of filters is 4 and the size is 3, the dilation factor is (1, 2, 4, and 8), and the residual connection layers is 1. LSTM and GRU's hidden layer is 1, and the number of nodes is 16. The last four fully-connected layers have 64, 32, 16, and 1 node, respectively.

(6) ATCN-BiDLSTM, TCN-BiDLSTM, and TCN-BiDGRU: the number of filters is 4 and the size is 3, the dilation factor is (1, 2, 4, and 8), and the residual connection layers is 1. BiDLSTM and BiDGRU's hidden layer are 1, and the number of nodes is 8. The last four fully-connected layers have 64, 32, 16, and 1 node, respectively.

Each model is well trained after reasonable parameter configuration, and the decrease in training loss is shown in Figure 13. The number of training cycles is set to 500 and it can be seen that the loss of each model tends to be stable in

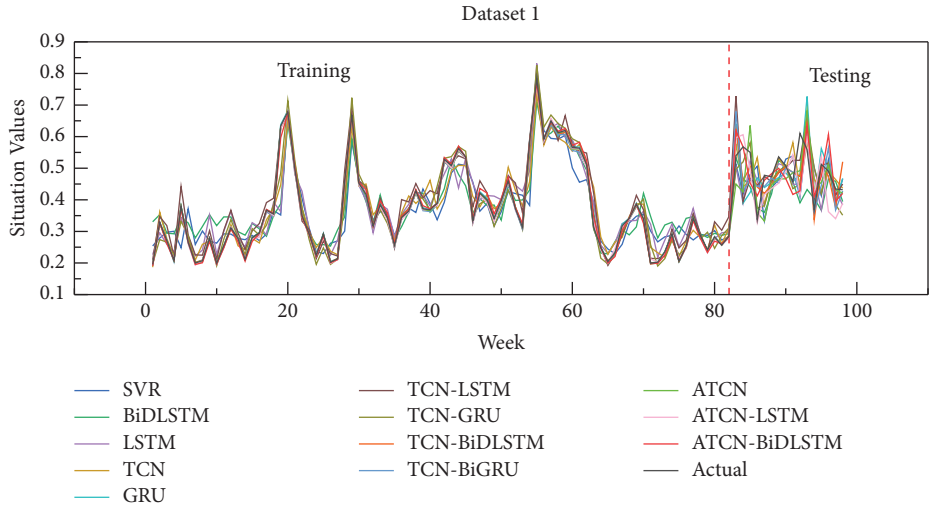

Figure 14: Fitting curves of each model on dataset 1.

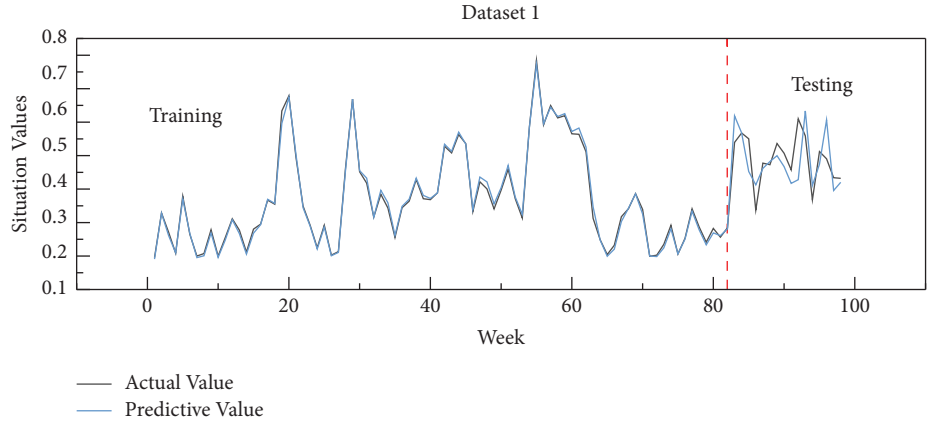

Figure 15: Fitting curves of the proposed model on dataset 1.

100 cycles, indicating that dataset 1 of each model can converge quickly and can achieve a good training effect.

Figure 14 shows the fitting curves of each model, and the fitting curves of the proposed model are shown in Figure 15. The wide range fluctuation of the situation value brings challenges to the prediction. In the training stage, each model can better fit the trend of the curve and can learn effectively. But it can also be seen that in the 0 to 20 time period, 40 to 50 time period, and 70 to 80 time period, the fitting effect of the single model is not as good as that of the mixed model, indicating that the difficulty of feature learning in this period becomes larger, and the feature learning ability of the single model shows limitations. In the test stage, each model has a certain deviation, but it can be seen that although the model proposed in this study still has a certain deviation in the predicted value, it can well predict the trend of the situation, and can more effectively capture the subtle changes of the situation value, indicating that the proposed model has stronger feature extraction ability and prediction ability than other hybrid models.

Table 4 shows the loss evaluation metrics of each model on dataset 1. It can be seen that the evaluation metrics of the SVR model are the worst, indicating that the DL model has advantages over traditional ML methods. The overall metrics of the hybrid model are better than that of the single model, indicating that the method combining feature extraction and prediction tools has greater advantages. Compared with the

Table 4: Comparison of metrics across models on dataset 1

Model	Metrics		
	MAE	MAPE	RMSE
SVR	0.0865	18.3413	0.1021
BiDLSTM	0.0658	12.7806	0.0898
LSTM	0.0671	13.9057	0.0893
TCN	0.0685	14.8139	0.0920
GRU	0.0700	13.9189	0.0909
TCN-LSTM	0.0616	12.3945	0.0837
TCN-GRU	0.0602	12.5181	0.0740
TCN-BiDLSTM	0.0578	11.9516	0.0695
TCN-BiGRU	0.0698	13.7644	0.0885
ATCN	0.0653	12.7507	0.0800
ATCN-LSTM	0.0639	12.7629	0.0777
ATCN-BiDLSTM	**0.0561**	**11.3181**	**0.0722**

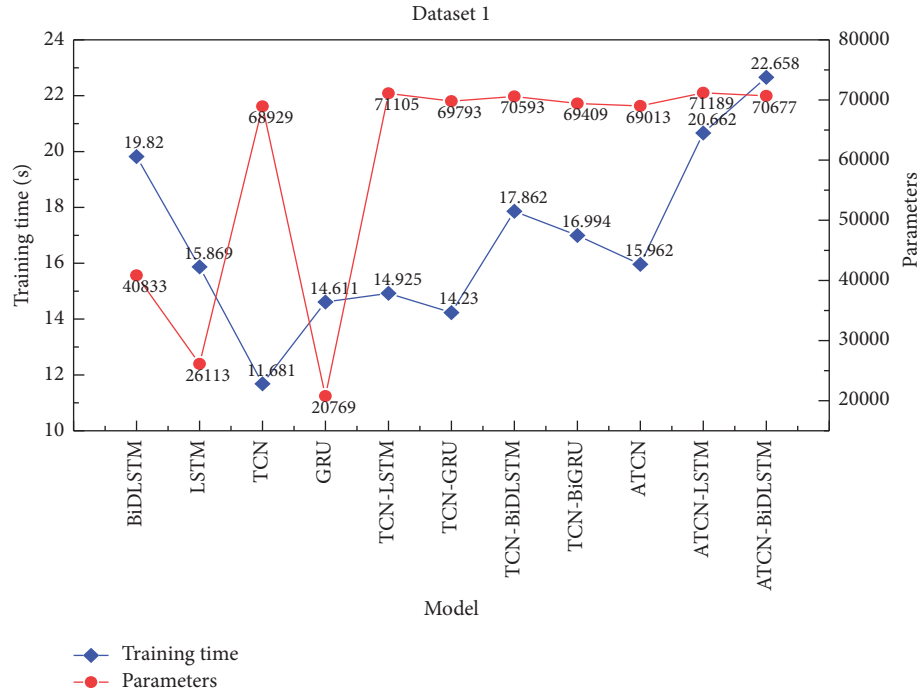

Figure 16: Training time and parameters for each model on dataset 1.

model without AM, the loss of the model with AM is reduced to a certain extent, indicating that AM improves the feature extraction ability of the model. It can also be seen that BiDLSTM has less loss than LSTM whether it is a single model or a hybrid model, indicating that BiDLSTM has better prediction ability. At the same time, the proposed hybrid model has better loss results than other hybrid models, indicating that the proposed hybrid model is more advanced and has higher prediction accuracy.

Figure 16 shows the comparison of training time and model parameters for DL models. It can be seen that TCNs spend the least time in training because TCNs can process data in parallel and has higher efficiency. Compared with LSTM, GRU merges input gates and forgetting gates, and its parameters are minimal. Due to the combination of multiple models, the parameters and training time of the hybrid model are generally greater than that of the single model. The number of model parameters and training time proposed in this study are both high, which is due to the AM. The combination of TCN and BiDLSTM structure improves model complexity. Moreover, to fully carry out feature learning, the training time also becomes longer.

4.2.2. Dataset 2. Dataset 2 was selected from the 32nd issue of 2017 to the 1st issue of 2022, for a total of 231 weeks. For the selection of time steps, 12 weeks were taken as one cycle. The parameters for each model are as follows.

(1) SVR: Use the linear kernel function and set the penalty factor to 1.

(2) BiDLSTM: The hidden layer has 32 nodes. The four fully-connected layers have 64, 32, 16, and 1 node, respectively.

(3) LSTM and GRU: The three hidden layers have 16, 32, and 32 nodes, respectively. The four fully-connected layers have 64, 32, 16, and 1 node, respectively.

(4) TCN and ATCN: The number of filters is 4 and the size is 3, the dilation factor is (1, 2, 4, and 8), and the residual connection layers is 1, and the four fully-connected layers have 64, 32, 16, and 1 node, respectively.

(5) TCN-LSTM, TCN-GRU, and ATCN-LSTM: The number of filters is 4 and the size is 3, the dilation factor is (1, 2, 4, and 8), and the residual connection layers is 1. LSTM and GRU's hidden layer is 1, and the number of nodes is 16. The last four fully-connected layers have 64, 32, 16, and 1 node, respectively.

(6) ATCN-BiDLSTM, TCN-BiDLSTM, and TCN-BiDGRU: The number of filters is 4 and the size is 3, the dilation factor is (1, 2, 4, and 8), and the residual connection layers is 1. BiDLSTM and BiDGRU's hidden layers are 1, and the number of nodes is 16. The last four fully-connected layers have 64, 32, 16, and 1 node, respectively.

The decrease in training loss is shown in Figure 17, and the number of training cycles is set to 500. It can be seen that compared with the single model, the overall loss of the hybrid model decreases faster and has a faster feature learning ability.

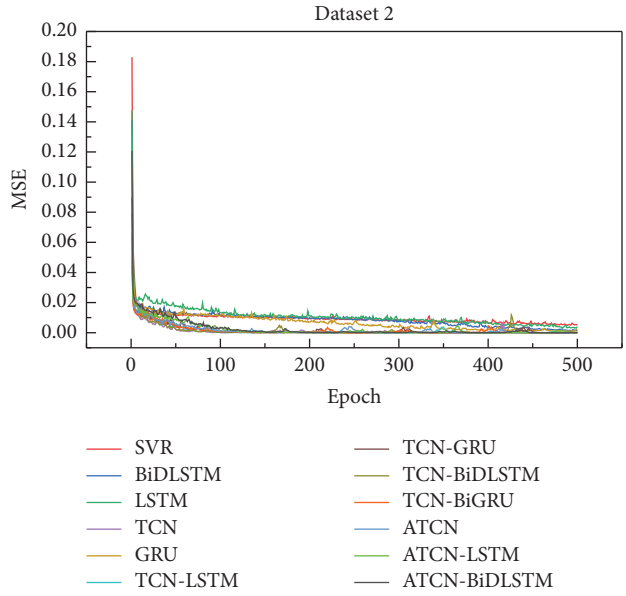

Figure 17: Training convergence curves for each model on dataset 2.

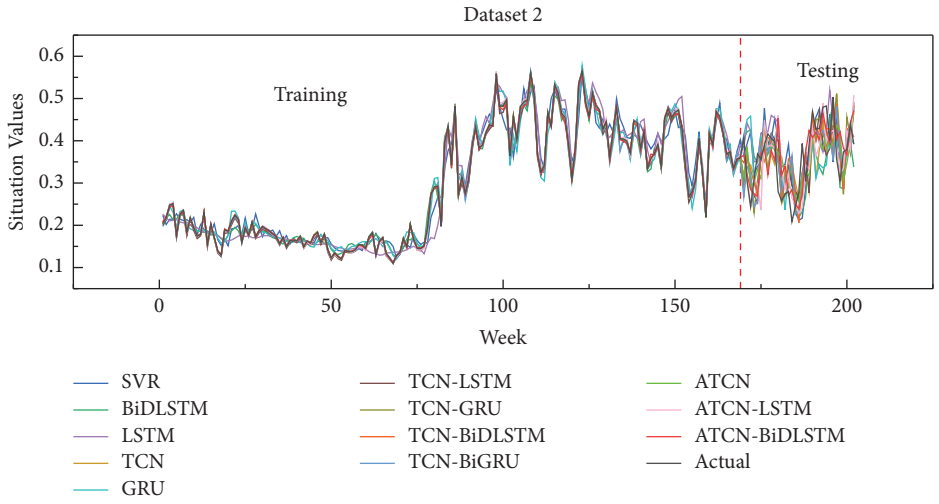

Figure 18: Fitting curves of each model on dataset 2.

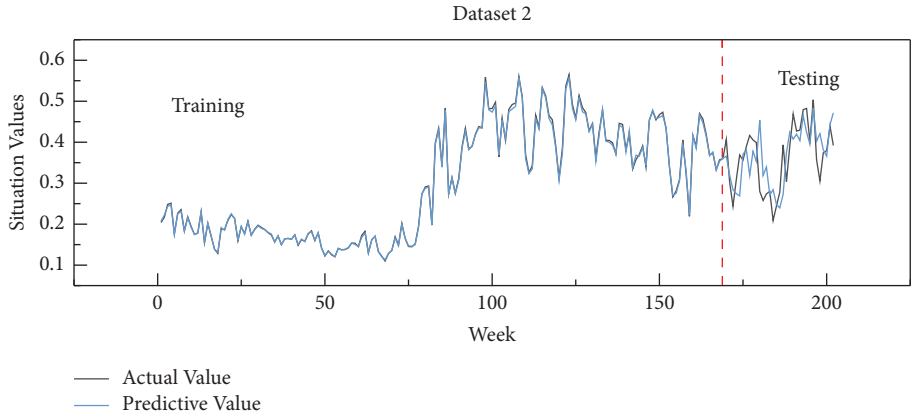

Figure 19: Fitting curves of the proposed model on dataset 2.

Table 5: Comparison of metrics across models on dataset 2.

Model	Metrics		
	MAE	MAPE	RMSE
SVR	0.0682	20.2076	0.0820
BiDLSTM	0.0575	16.9473	0.0753
LSTM	0.0579	17.8927	0.0741
TCN	0.0626	17.8601	0.0769
GRU	0.0612	18.6871	0.0814
TCN-LSTM	0.0572	16.4657	0.0662
TCN-GRU	0.0575	16.9201	0.0690
TCN-BiDLSTM	0.0529	16.1642	0.0653
TCN-BiGRU	0.0556	16.4503	0.0696
ATCN	0.0539	15.5705	0.0633
ATCN-LSTM	0.0504	14.3002	0.0656
ATCN-BiDLSTM	**0.0466**	**14.0611**	**0.0613**

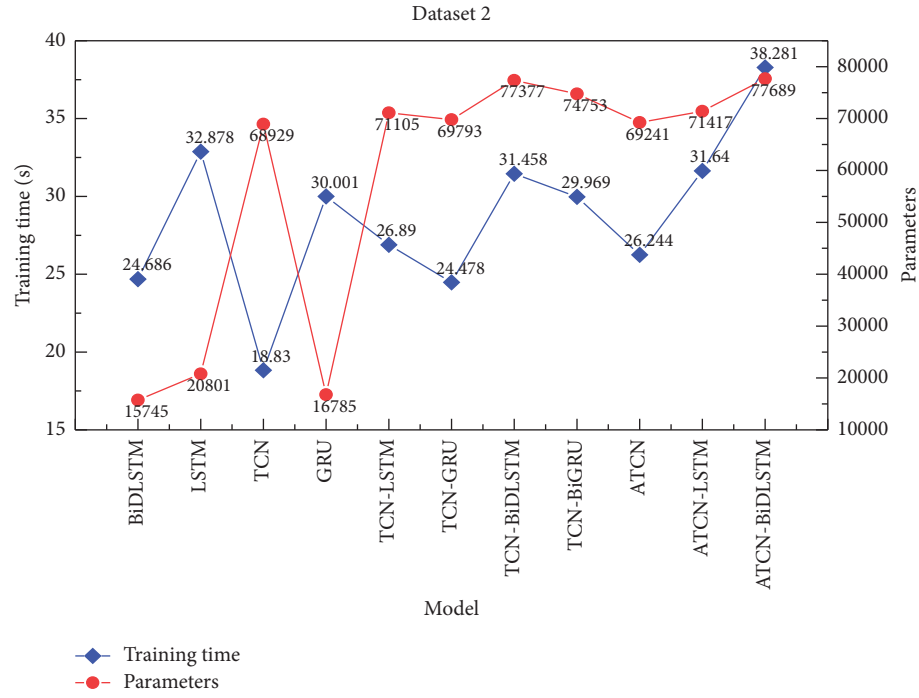

Figure 20: Training time and parameters for each model on dataset 2.

Figure 18 shows the fitting curves of each model, and the fitting curves of the proposed model are shown in Figure 19. Compared with dataset 1, dataset 2 has a larger fluctuation, and the situation value has a steep rise from week 75 to week 100, indicating that the overall network threat has increased in recent years. The situation prediction results are similar to dataset 1. In the training stage, each model can accurately capture the trend of situation change except for different fitting degrees. Compared with other models, the model proposed in the test phase can capture the subtle change trend of the situation value more accurately and is also relatively accurate in the prediction of the situation value.

Table 5 shows the loss evaluation metrics of each model in dataset 2. The overall effect is consistent with dataset 1, which proves that the hybrid model is better than the single model, and the use of AM helps TCN to learn features, indicating that the proposed model is more accurate than other hybrid models for network security situation prediction.

Figure 20 shows the comparison of training time and the model parameters of each DL model. The results are consistent with dataset 1. The model proposed in this study has a higher model complexity.

4.3. Discussion and Analysis. This study predicts the situation at two different time stages. The results show that the model proposed in this study has better performance than other models in situation prediction. In addition, the following information can be obtained.

(1) Compared with the single DL model, the hybrid DL model has a faster rate of decrease in training loss, and the prediction accuracy and fitting are also better, indicating that the overall prediction effect of the hybrid model is better than that of the single model.

(2) The prediction accuracy of the model using AM is improved compared with the original model, indicating that AM is helpful for feature extraction.

(3) The hybrid model proposed in this study is superior to other hybrid models, indicating that the model combining AM with TCN has a stronger feature extraction ability. At the same time, BiDLSTM performs better in time series prediction than that of LSTM and GRU.

(4) Although the hybrid model has better prediction accuracy, the complex model structure and longer training time make the model's performance limited.

5. Conclusion and Future Works

In this study, we propose a network security situation prediction method based on AM improved TCN combined with the BiDLSTM network. First, the TCN is improved by the AM, and the improved TCN has a stronger time-series feature extraction ability, which can learn the trend of the historical period of the situation values well. Second, the excellent time-series prediction ability of BiDLSTM is then used for the situation prediction. The experimental results show that compared with a variety of single and hybrid DL models, the proposed model has better results in RMSE, MAE, and MAPE. The proposed model has more effective feature extraction ability and prediction ability, so it has higher prediction accuracy and stability. In addition, in the fitting of the predictive value and the actual value, the model can also achieve a good fitting effect and can capture the subtle trend change. At the same time, the proposed model has a complex structure and many parameters, which has certain limitations. However, for situation prediction, a higher prediction accuracy is more important, so this model can be used as an effective network security situation prediction tool. In future work, we aim to apply the model to other time-series prediction scenarios inorder to validate the long-range prediction capability of the model. Other advanced prediction methods are combined with feature extraction for more effective prediction. In addition, the structural design of the model can focus on lightweight design, such as reducing the number of model layers and appropriately discarding the full-connection layer to reduce the model parameters, or selecting a more efficient and lightweight prediction model to meet different scene requirements. At the same time, a multistep forecasting method can be used to meet the needs of more long-term forecasting.

Data Availability

The data supporting the current study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like thank the Fundamental Research Fund of the School of Information Engineering, Engineering University of PAP (number WJY202130) for funding this research.

References

[1] T Bass and Tim, "Intrusion detection systems and multisensor data fusion," *Communications of the ACM*, vol. 43, no. 4, pp. 99–105, 2000.

[2] H Hu, HQ Zhang, YL Liu, and YW Wang, "Quantitative Method for Network Security Situation Based on Attack Prediction," *Security And Communication Networks*, vol. 2017, pp. 1–19, 2017.

[3] Zhongyang Han, Jun Zhao, Henry Leung, K F Ma, and Wei Wang, "A Review of Deep Learning Models for Time Series Prediction," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 7833–7848, 2021.

[4] Chaoxian Dong and Lixin Zhao, "Sensor network security defense strategy based on attack graph and improved binary PSO," *Safety Science*, vol. 117, pp. 81–87, 2019.

[5] Hongyu Yang, Le Zhang, Xugao Zhang, and Jiyong Zhang, "An Adaptive IoT Network Security Situation Prediction Model," *Mobile Networks & Applications*, pp. 1–11, 2021.

[6] Christian Janiesch, Patrick Zschech, and Kai Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.

[7] Jingyu Xing and Zheng Zhang, "Prediction model of network security situation based on genetic algorithm and support vector machine," *Journal of Intelligent & Fuzzy Systems*, pp. 1–9, 2021.

[8] W Jian, L Ke, and Z Guosheng, "Network security situation automatic prediction model based on accumulative CMA-ES optimization," *The Journal of China Universities of Posts and Telecommunications*, vol. 24, no. 3, pp. 33–43, 2017.

[9] Wei Feng, Yuqin Wu, and Yexian Fan, "A new method for the prediction of network security situations based on recurrent neural network with gated recurrent unit," *International Journal of Intelligent Computing and Cybernetics*, vol. 13, no. 1, pp. 25–39, 2020.

[10] Liqiong Chen, Guoqing Fan, Kun Guo, and Zhao, "Junyan. Security Situation Prediction of Network Based on Lstm Neural Network," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12639, pp. 140–144, 2021.

[11] Zhao Guosheng, Liu Dongmei, and Wang Jian, "Cloud security situation prediction method based on grey wolf optimization and bp neural network," *Journal of China Universities of Posts and Telecommunications*, vol. 27, no. 6, pp. 30–41, 2020.

[12] Liang Shen and Zhicheng Wen, "Network security situation prediction in the cloud environment based on grey neural network1," *Journal of Computational Methods in Sciences and Engineering*, vol. 19, no. 1, pp. 153–167, 2019.

[13] Dongmei Liu, Jie Cheng, Zilong Yuan et al., "Prediction Methods for Energy Internet Security Situation Based on Hybrid Neural Network," *IOP Conference Series: Earth and*

Environmental Science, vol. 645, no. 1, Article ID 012085, 2021.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] Junkai Ji, Minhui Dong, Qiuzhen Lin, and Kay Chen Tan, "Forecasting Wind Speed Time Series Via Dendritic Neural Regression," *IEEE Computational Intelligence Magazine*, vol. 16, no. 3, pp. 50–66, 2021.

[16] Mohsen Mousavi, Rohit Salgotra, Damien Holloway, and A H Gandomi, "COVID-19 Time Series Forecast Using Transmission Rate and Meteorological Parameters as Features," *IEEE Computational Intelligence Magazine*, vol. 15, no. 4, pp. 34–50, 2020.

[17] Tony Salloom, Okyay Kaynak, Xinbo Yu, and Wei He, "Proportional integral derivative booster for neural networks-based time-series prediction: Case of water demand prediction," *Engineering Applications of Artificial Intelligence*, vol. 108, p. 104570, 2022.

[18] K H Poon, P K Y Wong, and J. C. Cheng, "Long-time gap crowd prediction using time series deep learning models with two-dimensional single attribute inputs," *Advanced Engineering Informatics*, vol. 51, Article ID 101482, 2022.

[19] Anup Majumder, Rahman, Md Mahbubur et al., "Stock Market Prediction: A Time Series Analysis," *Smart Innovation, Systems and Technologies*, vol. 235, pp. 389–401, 2022.

[20] Jin Fan, Ke Zhang, Yipan Huang, Yifei Zhu, and Baiping Chen, "Parallel spatio-temporal attention-based TCN for multivariate time series prediction," *Neural Computing & Applications*, pp. 1–10, 2021.

[21] P J Moore, T J Lyons, and J Gallacher, "Random forest prediction of Alzheimer's disease using pairwise selection from time series data," *PLoS One*, vol. 14, no. 2, Article ID e0211558, 2019.

[22] Jinwoong Park and Eenjun Hwang, "A Two-Stage Multistep-Ahead Electricity Load ForecastingScheme Based on LightGBM and Attention-BiLSTM," *Sensors (Basel, Switzerland)*, vol. 21, no. 22, p. 7697, 2021.

[23] Jennifer Hill, Antonio Linero, and Jared Murray, "Bayesian Additive Regression Trees: A Review and Look Forward," *Annual Review of Statistics and Its Application*, vol. 7, no. 1, pp. 251–278, 2020.

[24] S T Noor, S T Asad, M M Khan, G S Gaba, J F Al-Amri, and M Masud, "Predicting the Risk of Depression Based on ECG Using RNN," *Computational intelligence and neuroscience*, vol. 2021, pp. 1–12, 2021.

[25] Sergei Mikhailov and Alexey Kashevnik, "Car Tourist Trajectory Prediction Based on Bidirectional LSTM Neural Network," *Electronics*, vol. 10, no. 12, p. 1390, 2021.

[26] Kaining Mao, Wei Zhang, Deborah Baofeng Wang et al., "Prediction of Depression Severity Based on the Prosodic and Semantic Features with Bidirectional LSTM and Time Distributed CNN," *IEEE Transactions on Affective Computing*, p. 1, 2022.

[27] Hoon Kang, Seunghyeok Yang, Jianying Huang, and Jeill Oh, "Time Series Prediction of Wastewater Flow Rate by Bidirectional LSTM Deep Learning," *International Journal of Control, Automation and Systems*, vol. 18, no. 12, pp. 3023–3030, 2020.

[28] Yuanyuan Wang, Jun Chen, Xiaoqiao Chen et al., "Short-Term Load Forecasting for Industrial Customers Based on TCN-LightGBM," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 1984–1997, 2021.

[29] Giovanni Menegozzo, Diego Dall'Alba, and Paolo Fiorini, "Industrial Time Series Modeling With Causal Precursors and Separable Temporal Convolutions," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6939–6946, 2021.

[30] Su Pei, Ke Niu, Xueping Peng, and Zeng, "Jingni. Readmission Prediction with Knowledge Graph Attention and RNN-Based Ordinary Differential Equations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12817, pp. 559–570, 2021.

[31] Saima Majid, Fayadh Alenezi, Sarfaraz Masood, Musheer Ahmad, E S Gündüz, and Kemal Polat, "Attention based CNN model for fire detection and localization in real-world images," *Expert Systems with Applications*, vol. 189, Article ID 116114, 2022.

[32] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *Learning*, 2018.

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully Convolutional Networks for Semantic Segmentation," *Computer Science*, 2014.

[34] M. Nauta, D. Bucur, and C. Seifert, "Causal discovery with attention-based convolutional neural networks," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 312–340, 2019.

[35] Andreas Veit, Michael Wilber, and Serge Belongie, "Residual Networks Behave Like Ensembles of Relatively Shallow Networks," *30th Annual Conference on Neural Information Processing Systems*, 2016.

[36] D Bahdanau, K Cho, and Y Bengio, "Neural machine translation by jointly learning to align and translate," 2014.

[37] Yan-Lin He, Lei Chen, Yanlu Gao, Jia-Hui Ma, Yuan Xu, and Qun-Xiong Zhu, "Novel double-layer bidirectional LSTM network with improved attention mechanism for predicting energy consumption," *ISA Transactions*, vol. 127, pp. 350–360, 2022.

[38] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *Statistics*, vol. 3, 2014.

[39] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," *Computer Science*, 2015.

[40] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[41] S Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] Yakubu Imrana, Yanping Xiang, Liaqat Ali, Z Abdul-Rauf, and Zaharawu, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, Article ID 115524, 2021.

[43] CNCERT, "Network Security Information and Dynamics Weekly Report," 2022.

[44] W F. JIANG, *Research on network security situation prediction based on multi model weight extraction and fusion*, Lanzhou University of Technology, 2016.