EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Network security situation prediction based on feature separation and dual attention mechanism

Zhijian Li[1,2], Dongmei Zhao[1,2*], Xinghua Li[3] and Hongbin Zhang[4]

*Correspondence: zhaodongmei666@126.com
[2] Hebei Key Laboratory of Network and Information Security, Hebei Normal University, Shijiazhuang 050024, China
Full list of author information is available at the end of the article

Abstract

With the development of smart cities, network security has become more and more important. In order to improve the safety of smart cities, a situation prediction method based on feature separation and dual attention mechanism is presented in this paper. Firstly, according to the fact that the intrusion activity is a time series event, recurrent neural network (RNN) or RNN variant is used to stack the model. Then, we propose a feature separation method, which can alleviate the overfitting problem and reduce cost of model training by keeping the dimension unchanged. Finally, limited attention is proposed according to global attention. We sum the outputs of the two attention modules to form a dual attention mechanism, which can improve feature representation. Experiments have proved that compared with other existing prediction algorithms, the method has higher accuracy in network security situation prediction. In other words, the technology can help smart cities predict network attacks more accurately.

Keywords: Situation prediction, Smart cities, Feature separation, Global attention, Limited attention

1 Introduction

Smart city mainly uses new generation information technologies to plan, manage and build city. Currently, many cities are building smart cities, and network security is a problem that must be faced in the construction of smart cities. If smart cities do not have a good network security environment, then various technologies used in smart cities may be used by hackers, thus affecting the normal operation of smart cities. In order to realize and maintain the security and privacy of smart cities, we intend to use the technology of network security situation prediction.

In 1988, Endsley defined situation awareness in the paper [1]. In addition, Endsley also discussed the construction of situation awareness and divided situation awareness into three levels consisting of perception of elements in current situation, comprehension of current situation, and projection of future status. Bass et al. [2] believed that the data flow on the Internet is similar to the airplane flying in the air, and the future network management is also similar to the air traffic control (ATC). Affected by the situation awareness needed in ATC, they believed that the next generation of IDSes can combine short-term sensor data with long-term knowledge databases to create cyberspace situation awareness.

Later researchers proposed more than a dozen network security situation awareness models, but they all have three basic functions [3]. The first is the extraction of network security situation elements, the second is the evaluation of the network security situation, and the third is the prediction of the network security situation. The main work of this paper is network security situation prediction. The main contributions of this paper are as follows:

1. We propose a feature separation (FS) method. Compared with the one-hot method, the method can alleviate the overfitting problem and reduce cost of model training by keeping the dimension unchanged.
2. In this paper, according to global attention, we propose limited attention mechanism. We sum the outputs of the two attention modules to form a dual attention mechanism (DAM), which can improve feature representation.
3. We use RNN, LSTM and GRU to validate our method. In addition, two data sets are used for experiments in the paper. It is also worth referring to the feature selection and preprocessing of the AWID data set.

The rest of this paper is arranged as follows. Section 2 roughly introduces the existing work. In Sect. 3, RNN, RNN variants and global attention mechanism are introduced. Section 4 introduces our work in detail. Section 5 introduces the results and experiments. Finally, Sect. 6 draws the conclusion of the paper, and the future direction of the work is explained.

2 Related work

There are many researches on network security situation prediction. Some researchers use traditional machine learning methods, while others combine intelligent optimization algorithm and machine learning. Currently, many researchers use deep learning-related algorithms to study network security situation prediction.

Wang et al. [4] compared several network security situation prediction methods. The results show that the radial basis neural network optimized by particle swarm optimization algorithm had better performance, but the number of data samples used in the experiment was too small. Zhang et al. [5] proposed a network security situation prediction method based on BP neural network. The authors used seeker optimization algorithm to find the best weight and introduced simulated annealing algorithm to improve the global search ability of the algorithm, but BP neural network is not suitable for time series data. Hu et al. [6] proposed a situation prediction algorithm of SVM based on MapReduce to solve the problem that the training time of SVM is long when there are a large number of samples. Compared with the traditional model, this prediction model improves the prediction accuracy and reduces the training time. Aiming at the slow convergence of the adaptive genetic algorithm in wavelet neural network, Zhang et al. [7] proposed an improved niche genetic algorithm for parameter optimization of wavelet neural network. The results show that the improved algorithm has faster convergence speed and higher accuracy.

As network security data are time series data, many scholars use RNN or its variants for situation prediction. Zhu et al. [8] proposed an improved LSTM algorithm for situation awareness. In order to speed up the convergence speed of the improved LSTM and reduce the learning time, based on the Nadam optimization algorithm, they proposed the Nadam with Look-ahead (NAWL) algorithm. Experiments show that the algorithm can get a relatively small mean square error. If the algorithm is added to the attention mechanism, it can be expected to have better results. Shang et al. [9] proposed a simplified bidirectional LSTM. Compared with the traditional LSTM, this algorithm has better prediction effect, but the authors did not consider the extraction of training set and test set. Hu et al. [10] proposed a network security situation prediction method based on RNN. The experiment shows that the method is feasible and has good prediction ability, but the number of samples used is too small. Li et al. [11] used LSTM to build a three-layer model with ReLU as the activation function. On the KDD CUP 99 data set, the experiments were also conducted with accuracy reaching beyond 90.56%. However, the accuracy is not satisfactory.

3 Basic theory

3.1 Variants of RNN

Recurrent neural network is an extension of traditional feedforward neural network and has the ability to process variable-length sequence input [12]. Long short-term memory (LSTM) is a variant of recurrent neural network, which is used to solve the problem of gradient vanishing, and gated recurrent unit (GRU) is a simplification of LSTM. The next two sections will briefly review LSTM and GRU.

3.1.1 Long short-term memory

The long short-term memory is a variant of RNN proposed by Hochreiter and Schmidhuber. In the section, the description of LSTM follows the description of LSTM in Chung et al. [12]. Given the input $X = (x_1, x_2, \ldots, x_{T'})$, we can describe the LSTM as follows.

Firstly, LSTM has three gates, which are forget gate f_t, input gate i_t, and output gate o_t.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1}) \tag{1}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1}) \tag{2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t) \tag{3}$$

where σ is the sigmoid activation function and $t \in [1, T^{'}]$. $T^{'}$ is the time step. W_f, W_i, W_o, U_f, U_i and U_o are weight matrices. V_f, V_i and V_o are diagonal matrices. Then, the memory cell c_t can be obtained by forget gate f_t and input gate i_t.

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \tag{4}$$

where forget gate f_t means forgetting part of the information of the previous memory cell c_{t-1}, and input gate i_t indicates that part of the information of a new memory cell $\tilde{c}_t$ is added to the memory cell c_t. The calculation formula of the new memory cell $\tilde{c}_t$ is shown as follows.

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1}) \tag{5}$$

In this formulation, W_c and U_c are weight matrices. *tanh* is a activation function:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{6}$$

Finally, the output h_t can be obtained through the output gate o_t and the memory cell c_t.

$$h_t = o_t tanh(c_t) \tag{7}$$

The function of the output gate o_t is to select appropriate information from memory cell for target prediction.

3.1.2 Gated recurrent unit

Gate current unit was proposed by Cho et al. [13] under the stimulation of LSTM. Compared with LSTM, gate current unit is simpler in calculation and implementation. Firstly, GRU has two gates, which are reset gate r_t and update gate z_t.

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{8}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{9}$$

where σ is the sigmoid activation function. W_r, W_z, U_r and U_z are weight matrices. We can get a new hidden state $\tilde{h}_t$ through reset gate r_t.

$$\tilde{h}_t = tanh(W x_t + U(r_t \odot h_{t-1})) \tag{10}$$

In this formulation, W and U are weight matrices. *tanh* is the activation function shown in (6). $\odot$ stands for the Hadamard Product. Here, reset gate r_t indicates how much information to obtain from the previous hidden state h_{t-1}. If the reset gate r_t is 0, no information will be obtained from the previous hidden state.

Then, we can get the current hidden state h_t through the update gate z_t.

$$h_t = z_t h_{t-1} + (1 - z_t)\tilde{h}_t \tag{11}$$

Update gate z_t represents how much information is carried from the previous hidden state to the current hidden state. The $(1 - z_t)$ represents how much information is carried from the new hidden state $\tilde{h}_t$ to the current hidden state.

3.2 Global attention

Since Bahdanau et al. [14] introduced the attention mechanism to machine translation, the attention mechanism has become more and more popular and plays an increasingly important role in neural networks. Global attention is proposed by Luong et al. [15]. In this paper, given the input X=(x_1,x_2, . . . , $x_{T'}$,β), we use the global attention shown in Fig. 1.

In Fig. 1, $\bar{h}_s$ is the s-th source hidden state, h is the target hidden state, and α is the alignment weight vector. Here, β marks the end of the input. It is a feature vector. Based on $\bar{h}_s$ and h, α can be obtained by an *align* function.

$$\begin{aligned}\alpha_s &= align(h, \bar{h}_s) \\ &= \frac{exp(score(h, \bar{h}_s))}{\sum_{s'=1}^{T'} exp(score(h, \bar{h}_{s'}))}\end{aligned} \tag{12}$$

In this formulation, α_s is the weight of $\bar{h}_s$, T' is the time step, and *score* is a dot product function.

$$score(h, \bar{h}_s) = h^T \bar{h}_s \tag{13}$$

A global context vector c is then computed as the weighted average, according to α, over all the source hidden states.

$$c = \sum_{s'=1}^{T'} \alpha_{s'} \bar{h}_{s'} \tag{14}$$

Finally, we employ a simple concatenation layer to combine the information from both vectors to get an attentional hidden state.

$$\tilde{h} = [c; h] \tag{15}$$

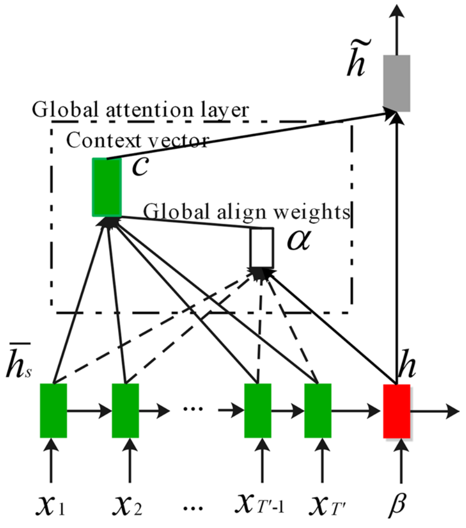


Fig. 1 Global attention model

Table 1 The samples we created

Sample	**Numerical data**	**Categorical data**	
	1	**2**	**3**
x_1	0.04	rsvp	INT
x_2	0.65	tcp	FIN
x_3	0.25	tcp	FIN
x_4	0.11	udp	CON

Table 2 The samples according to one-hot encoding method

Sample	**Numerical data**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
x_1	0.04	1	0	0	0	0	1
x_2	0.65	0	1	0	0	1	0
x_3	0.25	0	1	0	0	1	0
x_4	0.11	0	0	1	1	0	0

4 Methods

In this section, we introduce our proposed feature separation (FS) and limited attention firstly, and then, we propose a network security situation prediction model based on feature separation and dual attention mechanism (FSDAM) according to global attention, feature separation and limited attention.

4.1 Feature separation

Since categorical data exist in the data set and cannot be fed directly into the model like the numerical data, many researchers use one-hot encoding method. However, the one-hot method will make the dimension of features larger, which will cause the model to train more parameters and overfit. In this paper, in order to keep the dimension of features unchanged, we propose a feature separation method based on word embedding. Since categorical data and numerical data are processed separately, we call this technique feature separation, or FS for short.

In order to clearly explain one-hot, word embedding and feature separation (FS), we artificially created several samples as shown in Table 1. ‘rsvp,’ ‘tcp’ and ‘udp’ are transaction protocols. ‘INT,’ ‘FIN’ and ‘CON’ indicate to the state and its dependent protocols.

The one-hot encoding method uses a vector to represent a word. Length of the vector is number of words in the vocabulary. Each word has a unique vector. If index of the word in the vocabulary is i, the vector is 1 at i, and the remaining positions are 0. For example, in Table 1, ‘rsvp’ can be represented by ‘1 0 0.’ According to one-hot encoding method, Table 2 can be obtained. We can see that the dimension has changed from 3 to 7.

The word embedding generates vector representation of each word, the length of which is a hyperparameter. If embedding size is 3, in Table 1, ‘rsvp’ can be represented by ‘1.8 0.4 0.1.’ According to word embedding method, Table 3 can be obtained.

Table 3 The samples according to word embedding method

Sample	1	2	3
x_1	0.04	(1.8, 0.4, 0.1)	(−0.5, −1.3, 0.9)
x_2	0.65	(−1.9, −0.3, −0.4)	(0.9, 1.7, 0.1)
x_3	0.25	(−1.9, −0.3, −0.4)	(0.9, 1.7, 0.1)
x_4	0.11	(−0.1, −0.6, −0.1)	(−0.4, −0.5, −1.5)

Word embedding is only one phase of feature separation. Figure 2 shows the working progress of feature separation (FS). Firstly, x_1 is divided into $\tilde{x}_1$ and $\bar{x}_1$.Then, $\tilde{x}_1$ performs a linear transformation and sums by rows to get h_1. Finally, we employ a simple concatenation layer to combine the information from both vectors to get $x_1^{'}$. The dimension of $x_1^{'}$ is the same as that of x_1.

The architecture of the feature separation is shown in Fig. 3. Firstly, x_i is divided into $\tilde{x}_i$ and $\bar{x}_i$. $\tilde{x}_i$ is a matrix.

$$\tilde{x}_i = \begin{pmatrix} \tilde{x}_{i1} \\ \vdots \\ \tilde{x}_{ij} \\ \vdots \\ \tilde{x}_{in} \end{pmatrix}, (1 \leq j \leq n) \tag{16}$$

where $\tilde{x}_{ij}$ is the j-th feature vector in the categorical data, and n means there are n features in the categorical data. In order to get n features, $\tilde{x}_i$ performs a linear transformation with *tanh* as the activation function.

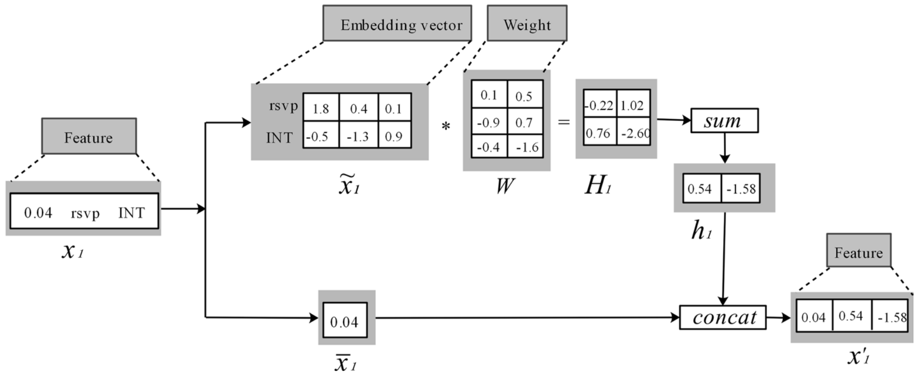


Fig. 2 Working progress of feature separation

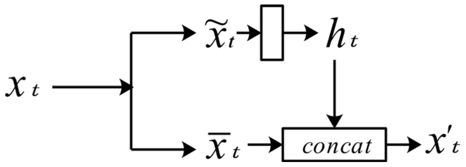


Fig. 3 Architecture of feature separation

$$H_i = tanh(\tilde{x}_i W + b) \tag{17}$$

where W is a weight matrix and b is a bias term; $H_i \in \mathbb{R}^{n \times n}$ is a matrix. Then, *sum* or *mean* is performed. In the work of this paper, *sum* is used by default.

$$h_i = \begin{cases} \sum_{j=1}^{n} H_{ij} & sum \\ \frac{1}{n} \sum_{j=1}^{n} H_{ij} & mean \end{cases} \tag{18}$$

In this formulation, h_i is the numerical representation of categorical data. Finally, we employ a simple concatenation layer to combine the information from both vectors to get $x_i^{'}$.

$$x_i^{'} = [\bar{x}_i; h_i] \tag{19}$$

4.2 Limited attention

In Section 3.2, we introduced the global attention mechanism. In this section, based on the global attention mechanism, we propose a limited attention mechanism as shown in Fig. 4. When model is a deep network, limited attention can alleviate the gradient vanishing problem by using the final output of each layer.

Given a network with $L = 5$ hidden layers, we can define the following attention mechanism.

Firstly, we employ a *slice* function to process the layer hidden state $\bar{h}_j$.

$$\bar{h}_j^{'} = slice(\bar{h}_j, k) \tag{20}$$

In this formulation, $\bar{h}_j$ is the hidden state of the j-th hidden layer and k is the dimension of the hidden state $h^{'}$ in the last hidden layer. The *slice* function has two cases: (1) Hidden

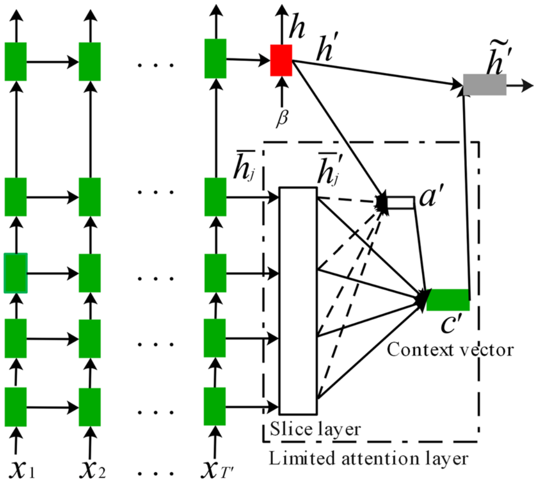


Fig. 4 Limited attention model

states have different dimensions. (2) Hidden states have same dimension. Figure 5 shows the working progress of *slice* function.

Based on the hidden state $h^{'}$, we can get the weight $\alpha_j^{'}$ of $\bar{h}_j^{'}$ through an *align* function.

$$\begin{aligned}\alpha_j^{'} &= align(h^{'}, \bar{h}_j^{'}) \\ &= \frac{exp(score(h^{'}, \bar{h}_j^{'}))}{\sum_{j^{'}=1}^{L-1} exp(score(h^{'}, \bar{h}_{j^{'}}^{'}))}\end{aligned} \tag{21}$$

where *score* is a dot product function.

$$score(h^{'}, \bar{h}_j^{'}) = h^{'T} \bar{h}_j^{'} \tag{22}$$

A context vector $c^{'}$ is then computed as the weighted average, according to $\alpha^{'}$, over all the layer hidden states.

$$c^{'} = \sum_{j^{'}=1}^{L-1} \alpha_{j^{'}}^{'} \bar{h}_{j^{'}}^{'} \tag{23}$$

Finally, we employ a simple concatenation layer to combine the information from both vectors to get an attentional hidden state.

$$\tilde{h}^{'} = [h^{'}; c^{'}] \tag{24}$$

Compared with global attention, limited attention has a *slice* function. Only a part of each hidden state is used when the dimensions are different, so we call it limited attention. Our limited attention uses the hidden state of the final output of each layer, focusing on the impact of the network layer. The work in this paper is the first case in Fig. 5.

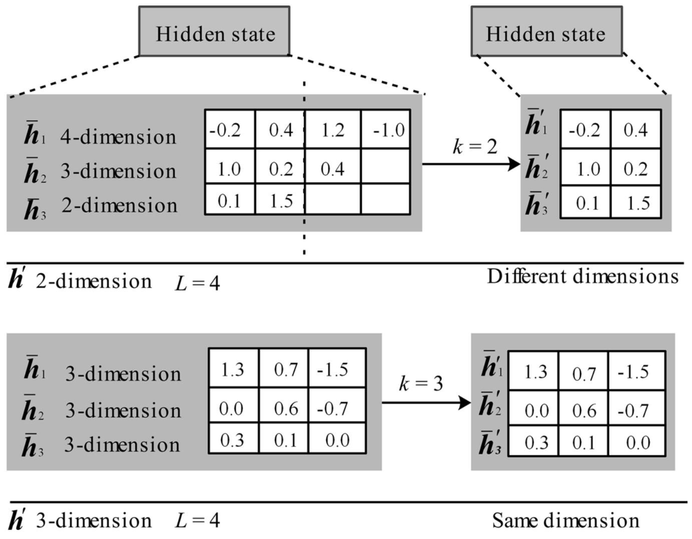


Fig. 5 Working progress of *slice* function

4.3 Design of the FSDAM+GRU model

Before this section, we have introduced gated recurrent unit, global attention, feature separation, and limited attention. According to these methods, we design the FSDAM+GRU model shown in Fig. 6. The model is mainly composed of feature separation (FS), gated recurrent unit (GRU) and dual attention mechanism (DAM). The dual attention mechanism is composed of global attention and limited attention. Inside the model, we also use shortcut connections [16].

Firstly, given the input X=(x_1,x_2,...,$x_{T'}$,β), we can use the feature separation (FS) layer to get X'=(x'_1,x'_2,...,$x'_{T'}$,β). Then, X' is input to the five-layer GRU network for training. The outputs of GRU are input to the global attention and the limited attention. Two attentional hidden states $\tilde{h}$ and $\tilde{h}'$ are obtained.

Finally, we sum the outputs of the two attention modules. The output of fusion is connected to the full connection layer to realize network security situation prediction.

5 Results and discussion

5.1 Description of data sets

In this paper, we are going to use two data sets, one is the UNSW-NB15 data set [17], and the other is the AWID data set created by Kolias et al. [18] in 2015. The following two subsections will describe the UNSW-NB15 and AWID data sets in detail.

5.1.1 UNSW-NB15 data set

To overcome the shortcomings of the KDD CUP 99 data set, Moustafa et al. [17] created the UNSW-NB15 data set in 2015. There are ten traffic data types in this data set, which are Normal, Dos, Fuzzers, Analysis, Exploits, Reconnaissance, Worm, Backdoors, Generic and Shellcode. Each sample contains 49 attributes, two of which are labels, one label is used for multi-classification, and the other is used for two-class classification. In addition, the data set has four files, which are UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv and UNSW-NB15_4.csv. The first three CSV files each file contains 700000 records, and the fourth file contains 440044 records. All records are sorted according to the last time attribute. Note that, during actual use, we found that the first three CSV files each file actually contains 700001 records.

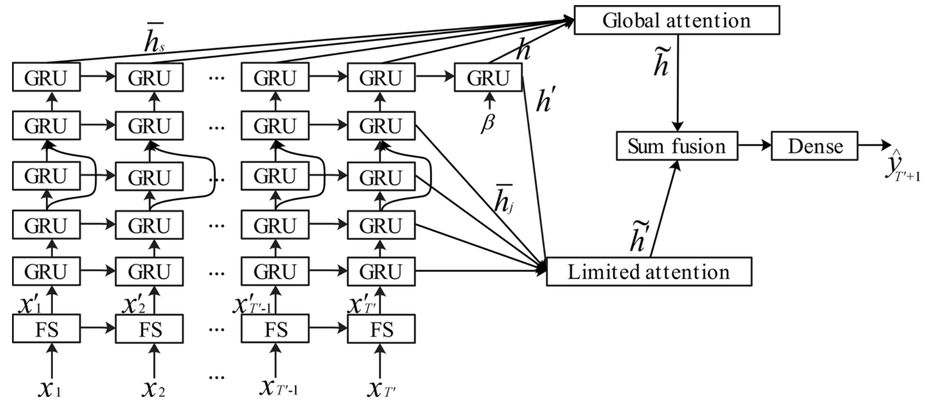


Fig. 6 FSDAM+GRU model

Table 4 Information about the samples of UNSW-NB15

Class	Training set	Test set
Normal	542,576	351,150
Generic	118,198	61,878
Exploits	16,574	11,439
Fuzzers	9137	5390
DoS	5642	4907
Reconnaissance	5582	3530
Analysis	873	670
Backdoor	759	666
Shellcode	593	371
Worms	67	43
Total	700,001	440,044

Table 5 Information about the samples of AWID

Class	Training set	Test set
Normal	1,633,190	530,785
Injection	118,198	61,878
Impersonation	16,574	11,439
Flooding	9137	5390
Total	1,795,575	575,643

In our experiments, we use the files UNSW-NB15_3.csv and UNSW-NB15_4.csv as our training set and test set. Detailed information about the training set and test set can be seen in Table 4.

5.1.2 AWID data set

The Aegean WiFi Intrusion Dataset (AWID) [18] is extracted from a dedicated WEP protected 802.11 network with real network traffic, not manual traffic. It contains a lot of normal traffic and attack traffic for 802.11 network. Compared with KDD CUP 99, AWID is a relatively new data set. In the data set, each sample has 154 features. According to the number of samples, the data set is divided into two versions, which are full subset (AWID-ATK-F and AWID-CLS-F) and reduced subset (AWID-ATK-R and AWID-CLS-R). The two versions are extracted from the real network.

In the experiments, we use AWID-CLS-R-Trn and AWID-CLS-R-Tst as our training set and test set, respectively. The AWID-CLS-R-Trn data set is obtained by monitoring the network for 1 h. The first 45 min is normal traffic, and the next 15 min contains attack traffic. In the training set, there are a total of 1,795,575 records, including 1,633,190 normal records and 162,385 attack records. In the test set, there are a total of 575,643 records, including 530,785 normal records and 44,858 attack records. Table 5 shows detailed information about the training set and test set.

5.2 Data set preprocessing

In order to meet the requirements of input format, data set preprocessing is necessary. For AWID data set, there are many missing values and features, which cannot be directly used. In order to improve the prediction accuracy, how to deal with missing values and how to select useful features for training also need to be studied in detail. In this section, preprocessing includes data filtration and feature selection, feature separation and standardization.

5.2.1 Data filtration and feature selection

Due to the problems of missing values and many attributes in the AWID data set, we provide our solution to solve these problems. Our solution is as follows:

1. In phase one, we discard the features that account for more than 90% of missing values in AWID-CLS-R-Trn. In the phase, 46 features are discarded.
2. In phase two, we discard eight useless features, which are **frame.time_epoch(#4)**, **wlan.ra (#76)**, **wlan.da (#77)**, **wlan.ta (#78)**, **wlan.sa (#79)**, **wlan.bssid (#80)**, **wlan.wep.iv (#140)** and **wlan.wep.icv (#142)**. For example, **wlan.ra (#76)**, **wlan.da (#77)**, **wlan.ta (#78)** and **wlan.sa (#79)** only represent the hardware MAC address.
3. In phase three, it is mainly to fill in the missing values of the data set. Numerical data fill in the mean at the missing places, and categorical data fill in $miss_i$ at the missing places, where i represents the i-th feature of categorical data to be filled.
4. In phase four, it is mainly to convert categorical data into numerical data.
5. In phase five, since the constant value has a limited effect on classification, we discard the features that are constant values. In the phase, 50 features are discarded.

According to the above five phases, there are only 50 features left in the 154 features of the AWID data set. Based on the previous work, this paragraph mainly focuses on feature selection. The main purpose of feature selection is to reduce dimension, eliminate irrelevant or redundant features, and then select important features. In [19], Chen et al. used extra trees for feature selection and selected 20 important features from the 154 features for intrusion detection. The results show that the method is effective for improving accuracy. We ran the algorithm and extracted the 38 most important features from the AWID data set for our experiments. A detailed description of the data features is provided in Table 6.

For the UNSW-NB15 data set, first of all, we discard six useless features, which are **srcip (#1)**, **sport (#2)**, **dstip (#3)**, **dsport (#4)**, **stime (#29)** and **ltime (#30)**, where **srcip (#1)** and **dstip (#3)** only represent IP address. Then, because **ct_flw_http_mthd (#38)**, **is_ftp_login (#39)** and **ct_ftp_cmd (#40)** have a large number of missing values, the three features are discarded. In the end, there are 38 features left in addition to two labels attributes. Table 7 shows the 38 features used in our experiments.

Table 6 Features extracted by extra trees classifier

No	Feature name	No	Feature name	No	Feature name	No	Feature name
5	.time_delta	64	.type_subtype	90	.ess	101	.dsss_ofdm
6	_displayed	66	wlan.fc.type	91	.ibss	102	.del_blk_ack
7	.time_relative	67	wlan.fc.subtype	92	.cfpoll.ap	103	.imm_bl_ack
8	frame.len	68	wlan.fc.ds	93	.privacy	110	.fixed.reason_code
9	.cap_len	70	wlan.fc.retry	94	.preamble	141	wlan.wep.key
38	.mactime	71	wlan.fc.pwrmgt	95	.pbcc	145	wlan.qos.tid
47	.datarate	73	fc.protected	96	.agility	148	wlan.qos.ack
50	.type.cck	75	wlan.duration	98	_slot_time	154	data.len
51	.type.ofdm	81	wlan.frag	99	.apsd		
61	.dbm_antsignal	82	wlan.seq	100	_measurement		

Table 7 Features of UNSW-NB15 data set after data filtration

No	Feature name	No	Feature name	No	Feature name	No	Feature name
5	proto	15	sload	25	trans_depth	37	ct_state_ttl
6	state	16	dload	26	res_bdy_len	41	ct_srv_src
7	dur	17	spkts	27	sjit	42	ct_srv_dst
8	sbytes	18	dpkts	28	djit	43	ct_dst_ltm
9	dbytes	19	swin	31	sintpkt	44	ct_src_ ltm
10	sttl	20	dwin	32	dintpkt	45	ct_src_dport_ltm
11	dttl	21	stcpb	33	tcprtt	46	ct_dst_sport_ltm
12	sloss	22	dtcpb	34	synack	47	ct_dst_src_ltm
13	dloss	23	smeansz	35	ackdat		
14	service	24	dmeansz	36	is_sm_ips_ports		

5.2.2 Standardization and feature separation

After obtaining 38 features, we redo some preprocessing. In the training set and test set of the AWID data set, numerical data fill in the mean at the missing place, and categorical data fill in the $miss_i$ at the missing place.

For each data set, we divide their features into two types, one is categorical data, and the other is numerical data. For example, in the UNSW-NB15 data set, **proto (#5)**, **state (#6)** and **service (#14)** are categorical data, and the rest are numerical data. In the AWID data set, **.type_subtype (#64)**, **wlan.fc.ds (#68)**, **.cfpoll.ap (#92)**, **.fixed.reason_code (#110)** and **wlan.qos.ack (#148)** are categorical data, and the rest are numerical data. We use $\bar{x}$ to represent numerical data and $\tilde{x}$ to represent categorical data. For numerical data, each feature is standardized. Standardization is as shown in (25).

$$\bar{x}_{ij}^{'} = \frac{\bar{x}_{ij} - \mu_j}{\sigma_j} \tag{25}$$

where $\bar{x}_{ij}$ is the j-th feature value of the i-th sample, μ_j is the mean value of the j-th feature, and σ_j is the standard deviation of the j-th feature.

For categorical data, we use word embedding technology which is widely used in the field of natural language processing.

Table 8 The accuracy of models with different hyperparameters

No	Recurrent neurons	Accuracy (Tst)		
		Mean (%)	Max (%)	Min (%)
HyPa1	[30,20,20,20,10]	92.927	92.939	92.913
HyPa2	[35,30,30,20,20]	92.936	92.942	92.928
HyPa3	[40,30,30,20,10]	92.934	92.947	92.906
HyPa4	[40,30,30,30,20]	92.940	92.952	92.923
HyPa5	[45,40,40,30,30]	92.944	92.956	92.933
HyPa6	[50,40,40,30,20]	92.942	92.947	92.936

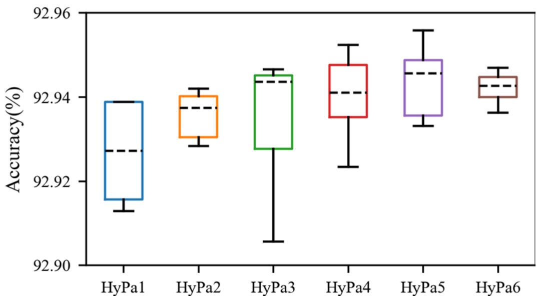


Fig. 7 The box plot of accuracy according to hyperparameters

5.3 Model configuration and training

The section mainly describes the configuration used by the model. In the paper, keras-gpu-2.3.1 and tensorflow-gpu-1.15.0 are used to build the structure of the model. In our model, we use softplus as the activation function.

In the training phase of the model, batch is set to 3072 , and time-step is set to 4. We use the mean, maximum and minimum values of accuracy obtained from five runs to measure our model. In addition, Adam [20] is used as the optimizer, and the parameters of Adam are set to lr=0.001, beta_1=0.9 and beta_2=0.999.

5.4 Experiments

5.4.1 Selection of hyper parameters

In order to investigate the impact of the output size of each layer on the model performance, we designed six different hyper parameter configurations based on FSDAM+GRU model. UNSW-NB15 was used as the data set in the experiment. The results are shown in Table 8 and Fig. 7. Figure 7 shows boxplot of five results.

From the boxplot, we can see that the trend upward in terms of median classification accuracy (black dashed line on the box) from HyPa1 to HyPa5. There also appears to be a decrease at HyPa6. Perhaps HyPa5 would be a good configuration. According to the mean and maximum values in the table, it can also be clearly seen that HyPa5 is superior to others. Therefore, we will use configurations of HyPa5 for next experiments, as it will be efficient for improving accuracy.

5.4.2 Comparison of one-hot and FS

In this section, we made a detailed comparison between FS and one-hot. The results are shown in Tables 9, 10 and Fig. 8. 'Params' in the tables represents the total number of parameters for model training.

From results, we can see that different methods have a significant effect on performances of DAM+GRU. According to the mean, maximum and minimum values in each table, DAM+GRU using FS is superior to DAM+GRU using one-hot, which shows that FS can improve accuracy. According to the 'Params' in each table, we can also see that the number of parameters using FS is lower than that using one-hot. Due to the decrease in the number of parameters, FS can improve the training efficiency of DAM+GRU. Figure 8 can further illustrate that FS can improve accuracy, which is particularly obvious on the AWID data set.

The main reason is that the one-hot method will make the dimension of features larger, and the increase in dimension will lead to the redundancy of internal parameters. The model may overfit due to redundancy of parameters. On the UNSW-NB15 data set, the input dimension using FS is 38, while the input dimension using one-hot is 196. On the AWID data set, the input dimension using FS is 38, while the input dimension using one-hot is 95.

5.4.3 Comparison of RNN, LSTM and GRU

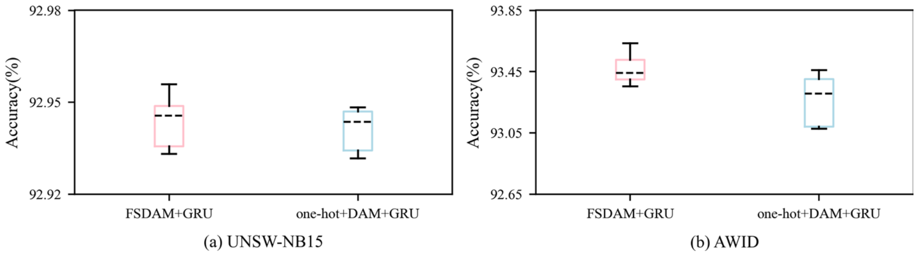
In this section, we measured the performance according to three different recurrent units. The experimental results based on the UNSW-NB15 data set are presented

Table 9 Accuracy and parameters of different methods on UNSW-NB15

Method	Accuracy (Tst)			Params
	Mean (%)	Max (%)	Min (%)	
FSDAM + GRU	92.944	92.956	92.933	50,213
one-hot + DAM + GRU	92.941	92.948	92.932	70,720

Table 10 Accuracy and parameters of different methods on AWID

Method	Accuracy (Tst)			Params
	Mean (%)	Max (%)	Min (%)	
FSDAM + GRU	93.472	93.635	93.355	49,042
one-hot + DAM + GRU	93.268	93.460	93.078	56,719



(a) UNSW-NB15 (b) AWID

Fig. 8 Comparison results with one-hot method on two data sets

in Table 11. The experimental results based on the AWID data set are presented in Table 12. Figure 9 is the comparison results on the two data sets.

From Fig. 9, we can find that FSDAM using LSTM is worse than others in terms of median classification accuracy (black dashed line on the box). In the two tables, the LSTM structure is also lower than other structures in terms of mean and maximum values. The LSTM structure uses more training parameters. Therefore, the LSTM structure is more likely to overfit, resulting in decreased accuracy. Compared with the LSTM structure, GRU and RNN are relatively simple, and they have higher time efficiency. Based on the experimental results, we recommended RNN and GRU.

5.4.4 Comparison of different methods

In this section, we compared our method with existing methods in detail. Tables 13, 14, 15 and 16 provide comparisons between our methods and existing methods. Figure 10 is the comparison results on the UNSW-NB15 data set. Figure 11 is the comparison results on the AWID data set.

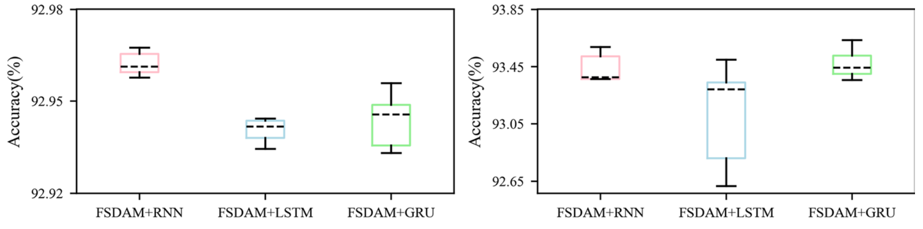
From Fig. 10, we can see that the method using FSDAM achieved the highest value of median classification accuracy (black dashed line on the box). In Fig. 11, the accuracy distribution of the method using FSDAM is at a higher position. The two figures show that the method using FSDAM can more accurately judge the network security situation. Looking at the two figures, we can also see that global attention cannot significantly

Table 11 Accuracy of different methods on UNSW-NB15

Method	**Accuracy (Tst)**		
	Mean (%)	**Max (%)**	**Min (%)**
FSDAM + RNN	92.962	92.967	92.958
FSDAM + LSTM	92.940	92.944	92.935
FSDAM + GRU	92.944	92.956	93.933

Table 12 Accuracy of different methods on AWID

Method	**Accuracy (Tst)**		
	Mean (%)	**Max (%)**	**Min (%)**
FSDAM + RNN	93.442	93.587	93.362
FSDAM + LSTM	93.110	93.496	92.616
FSDAM + GRU	93.472	93.635	93.355



(a) UNSW-NB15

(b) AWID

Fig. 9 Comparison results of three methods on two data sets

Table 13 Accuracy of different methods on UNSW-NB15

Method	Accuracy (Tst)		
	Mean (%)	Max (%)	Min (%)
one-hot + RNN	92.953	92.958	92.948
one-hot+global attention+RNN	92.945	92.950	92.939
FSDAM + RNN	92.962	92.967	92.958

Table 14 Accuracy of different methods on UNSW-NB15

Method	Accuracy (Tst)		
	Mean (%)	Max (%)	Min (%)
one-hot + GRU	92.940	92.949	92.936
one-hot+global attention+GRU	92.941	92.946	92.935
FSDAM + GRU	92.944	92.956	92.933

Table 15 Accuracy of different methods on AWID

Algorithm	Accuracy (Tst)		
	Mean (%)	Max (%)	Min (%)
one-hot + RNN	93.160	93.532	92.950
one-hot+global attention+RNN	92.460	93.147	92.163
FSDAM + RNN	93.442	93.587	93.362

Table 16 Accuracy of different methods on AWID

Method	Accuracy (Tst)		
	Mean (%)	Max (%)	Min (%)
one-hot + GRU	93.152	93.322	92.971
one-hot+global attention+GRU	93.119	93.332	92.713
FSDAM + GRU	93.472	93.635	93.355

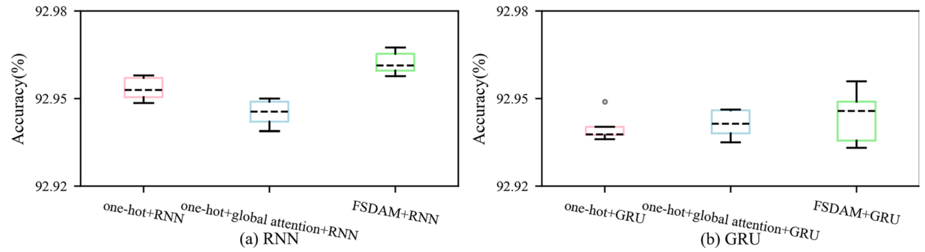


Fig. 10 Comparison results with other existing methods on UNSW-NB15

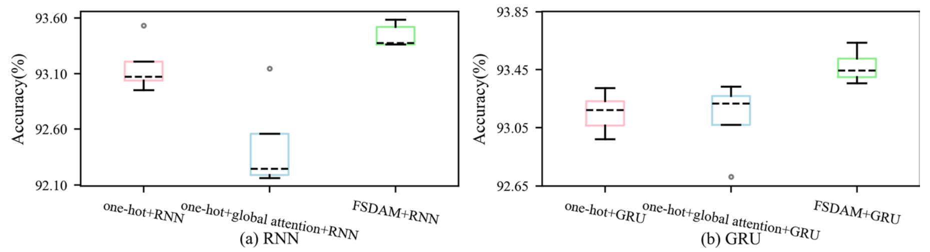


Fig. 11 Comparison results with other existing methods on AWID

improve the accuracy of one-hot+GRU, or even reduce the accuracy of one-hot+RNN. It may indicate that global attention does not improve feature representation very well.

In each table, we can also find that compared with other methods, the method using FSDAM has higher mean and maximum values, which indicates that FSDAM is a relatively stable method.

Looking at all the results, FSDAM is more suitable for network security situation prediction. Therefore, we recommended FSDAM+RNN and FSDAM+GRU to predict the security situation.

6 Conclusion

In this paper, we proposed a network security situation prediction method based on feature separation and dual attention mechanism. The experimental results show that the method is helpful to improve the accuracy of situation prediction, and the feature separation (FS) method used in the paper is helpful to alleviate the overfitting problem and reduce cost of model training. In the next step of research, the Transformer's [21] ability in time series prediction should be paid attention to, and its application in situation prediction needs further research.

Abbreviations

RNN:: Recurrent neural network; LSTM:: Long short-term memory; GRU:: Gated recurrent unit; FS:: Feature separation; DAM:: Dual attention mechanism; FSDAM:: Feature separation and dual attention mechanism; ATC:: Air traffic control; SVM:: Support vector machine; NAWL:: Nadam with look-ahead; AWID:: Aegean WiFi intrusion dataset; IDSes:: Current-generation intrusion-detection systems..

Acknowledgements

The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of our study. We would also like to thank Moustafa et al. and Kolias et al. Our experiments use the data sets they created.

Authors' contributions

All the authors participated in the writing and experiments of this paper, and ZL was mainly responsible for the implementation and design of the algorithms in this paper. All authors read and approved the final manuscript.

Funding

This research was funded by National Natural Science Foundation of China (No. 61672206), Hebei Science Supported Planning Projects (No. 20310701D), and Central Government Guides Local Science and Technology Development Fund Projects (No. 216Z0701G).

Availability of data and materials

The UNSW-NB15 data set and AWID data set we used can be obtained by using Google search. They are public.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details
[1]School of Computer and Cyber Security, Hebei Normal University, Shijiazhuang 050024, China. [2]Hebei Key Laboratory of Network and Information Security, Hebei Normal University, Shijiazhuang 050024, China. [3]School of Cyber Engineering, Xidian University, Xi'an 710071, China. [4]School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China.

Received: 29 December 2020 Accepted: 9 September 2021
Published online: 21 September 2021

References

1. M.R. Endsley, Design and evaluation for situation awareness enhancement. in *Proceedings of the Human Factors Society Annual Meeting*, **32**, 97–101 (1988)
2. T. Bass, D. Gruber, A glimpse into the future of id. Mag. USENIX SAGE **24**(3), 40–49 (1999)
3. R. Xi, X. Yun, S.-Y. Jin, Y.-Z. Zhang, Research survey of network security situation awareness. J. Comput. Appl. **32**(1), 1–4 (2012)
4. G. Wang, Comparative study on different neural networks for network security situation prediction. Secur. Privacy **4**(1), 138 (2021)
5. R. Zhang, M. Liu, Y. Yin, Q. Zhang, Z. Cai, Prediction algorithm for network security situation based on bp neural network optimized by sa-soa. Int. J. Performability Eng. **16**(8), 1171–1182 (2020)
6. J. Hu, D. Ma, C. Liu, Z. Shi, H. Yan, C. Hu, Network security situation prediction based on mr-svm. IEEE Access **7**, 130937–130945 (2019)
7. H. Zhang, Q. Huang, F. Li, J. Zhu, A network security situation prediction model based on wavelet neural network with optimized parameters. Digital Commun. Netw. **2**(3), 139–144 (2016)
8. J. Zhu, C. Sen, Network security situation prediction method based on nawl-ilstm. Comput. Sci. **46**(10), 161–166 (2019)
9. L. Shang, W. Zhao, J. Zhang, Q. Fu, Q. Zhao, Y. Yang, Network security situation prediction based on long short-term memory network. in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–4 (2019). IEEE
10. X. Hu, Prediction of network security situation based on rnn. Modern Comput. (6), 14–16 (2017)
11. S. Li, D. Zhao, Q. Li, A framework for predicting network security situation based on the improved lstm. EAI Endorsed Trans. Collab. Comput. **4**(13), 165278 (2020)
12. J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
13. K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation. in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734 (2014)
14. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate. in Bengio, Y., LeCun, Y. (eds.) *3rd International conference on learning representations (ICLR)*, pp. 1–15 (2015)
15. T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation. in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421 (2015)
16. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
17. N. Moustafa, J. Slay, Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). in *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6 (2015). IEEE
18. C. Kolias, G. Kambourakis, A. Stavrou, S. Gritzalis, Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Commun. Surveys Tutor. **18**(1), 184–208 (2015)
19. H. Chen, J. Chen, Recurrent neural networks based wireless network intrusion detection and classification model construction and optimization. J. Electron. Inf. Technol. **41**, 1427–1433 (2019)
20. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization. in *3rd International Conference on Learning Representations (ICLR)*, pp. 1–15 (2015)
21. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need. in *Advances in Neural Information Processing Systems*, pp. 5998–6008 (2017)

Publisher's Note