Online Criminal Behavior Recognition Based on CNNH and MCNN-LSTM

Jingwei Hu
Department of Legal Practice, Shandong Judicial Police Vocational College, Jinan 250200, China
E-mail: 17866981007@163.com

Keywords: anonymous networks, traffic segmentation, convolutional neural networks, online crime, long short-term memory networks

Received: November 10, 2024

*In light of the proliferation of cybercrimes, the effective identification and mitigation of such online criminal activities has emerged as a significant challenge within the domain of network security. Therefore, this study introduces dilated convolution technology, self-attention mechanism, convolutional neural network and long short-term memory network, and proposes an overlapping traffic recognition model based on improved convolutional neural network and an online crime recognition model with long short-term memory network. In the traffic segmentation model test, the recall rate, F1 value, and error rate of the model under normal traffic conditions were 91.43%, 93.46%, and 92.43%, respectively. The error rate was 4.15%. The accuracy of the online crime recognition model for malware propagation and illegal transactions was 96.54% and 92.87% respectively. In the concept drift test, when the training time and test time interval was 60 days, the accuracy of the model was 48.67% higher than that of the long short-term memory network. Compared with the mainstream framework and traditional methods, its accuracy in high traffic scenarios was 94.78%, the error rate was 3.89%, and the P-*value *was < 0.05. In the final simulation test, the model could effectively identify illegal software transactions. The results show that the proposed model has high accuracy and strong generalization ability in identifying overlapping traffic and website fingerprint crimes, and effectively improves the detection ability of criminal activities in anonymous networks.*

Povzetek: Predstavljen je model za prepoznavanje spletnega kriminala, ki temelji na konvolucijskih in LSTM nevronskih mrežah in z uporabo tehnologije razredčene konvolucije in mehanizma samo-pozornosti dosega visoko točnost pri segmentaciji prometa in prepoznavanju spletnih kaznivih dejanj. Učinkovito izboljšuje zaznavanje kriminalnih aktivnosti v anonimnih omrežjih.

1 Introduction

With the rapid development of Internet technology, the increasing complexity and openness of cyberspace have brought unprecedented opportunities and challenges to society [1]. The emergence and popularization of anonymous networks provide an important guarantee for users' privacy protection in the network. However, it also makes some wrongdoers utilize anonymous networks to engage in various criminalactivities, among which the anonymous communication system represented by the onion router (Tor) is particularly typical [2]. Tor network realizes the high anonymity of user identity and communication content through multi-layer encryption and node forwarding techniques, which is widely used for legitimate purposes such as protecting user privacy and preventing network surveillance. However, the anonymity of Tor network is also used by some criminals to circumvent legal supervision and become a hotbed for cybercriminal activities, such as illegal trading, malware distribution, hacking and other behaviors [3]. In this context, applying overlapping traffic segmentation and website fingerprinting (WF) technology to collect potential criminal evidence and detect abnormal behavior in an early stage in order to identify and combat online criminal behavior in anonymous networks has become a key issue that needs to be addressed urgently. At the same time, the industry's research on anonymous network traffic analysis and criminal behavior identification is also deepening and developing. Wang Y et al. proposed a deep learning-based intrusion detection system SMSO-CNN to address the security risks and privacy issues caused by the transmission of large amounts of data in wireless networks. The system combined the spider monkey swarm optimization algorithm and CNN to improve the ability to identify network attacks. The results showed that the system was superior to LSTM and other methods in terms of accuracy [4]. Gu X et al. proposed an online defense strategy based on non-targeted adversarial patches to address the limitations of existing WF attack defense methods in practical applications. Experiments indicated that the model achieved 95.50% defense accuracy and 12.57% time overhead in real-time traffic [5]. To address the problem of high dimensionality of cybercrime data, Rawat R et al. proposed a feature selection method based on multi-objective evolutionary algorithm (MOEA) and combined it with NSGA-II to reduce data dimensionality and identify the most relevant features. The experimental results indicated that this method effectively improves the efficiency of data processing [6]. Xian K proposed an

improved WF fingerprint recognition algorithm to solve the problem of identifying encrypted traffic in virtual private networks. Moreover, it combined it with an optimized capsule neural network model CapsNet to classify encrypted traffic. The research results showed that this method was superior to the random forest algorithm in terms of recognition accuracy and convergence speed, with a recognition rate of 99.98% [7]. Milad N et al. proposed a blind adversarial perturbation algorithm to address the problem that traffic analysis technology based on deep neural networks (DNN) was vulnerable to adversarial perturbation attacks. By remapping functions to create adversarial perturbations independent of network connections, the algorithm was applied to real-time anonymous network traffic analysis to defeat WF identification and traffic association classifiers. The experimental results indicated that this method was applicable to a variety of traffic classifier types. The robustness test of existing countermeasures performed poorly [8].

Because of their superior 2D data processing capabilities, convolutional neural networks (CNNs) are frequently utilized in image categorization and target recognition applications. Yesodha K et al. suggested a novel intrusion detection system incorporating CNN, fuzzy temporal rules, and an artificial bee colony optimization algorithm for the security vulnerability problem in wireless sensor network communication with the goal of improving the classifier's performance. Based on experimental assessments, the model performs better in terms of increased accuracy and decreased false alarm rate than popular classification algorithms like long short-term memory (LSTM) [9]. A CNN intrusion detection technique based on data imbalance was presented by Gan B et al. to address the hazards to network security brought on by recurrent network intrusions. The findings revealed that, with an implementation time of 1.42 seconds, the method attained an average accuracy of 98.73% in binary and multi-classification identification [10]. An intelligent prediction technique for security performance was suggested by Xu L et al. to address security concerns in mobile IoT healthcare networks. To increase the CNN model's adaptability to nonlinear medical large data, the study combined a four-branch beginning block with a four-layer convolution. The results indicated that the intelligent algorithm improved the security performance prediction accuracy by 20% and had better prediction performance [11]. Yan F. et al. addressed the issue of inadequate training samples and sample class imbalance in intrusion detection systems by proposing an intrusion detection system based on migration learning and integrated learning. The two fundamental learning models that were selected were Xception and Inception. A tree-structured estimator was used to tune the hyperparameters [12]. Finally, the study summarizes the research areas, indicator test results, and limitations of the above literature review. The results are shown in Table 1 below.

Table 1: Literature summary table

Study	Methodology	Performance Metric	Shortcomings
Wang Y et al. [4]	Intrusion detection system based on SMSO-CNN	Higher accuracy than LSTM and nearest neighbor algorithms	Not designed for anonymous network traffic, struggles with overlapping traffic
Gu X et al. [5]	Fingerprint defense strategy of online website based on Grad-CAM	95.50% defense accuracy, 12.57% time overhead	Focuses on defense tasks, does not address abnormal behavior recognition in anonymous networks
Rawat R et al. [6]	Feature selection method based on MOEA combined with NSGA-II for dimensionality reduction	Effectively improves data processing efficiency	Focused on feature selection, lacks real-time traffic analysis
Xian K et al. [7]	Optimized fingerprint recognition for encrypted traffic based on CapsNet	SSL VPN traffic recognition rate of 99.98%, recall rate of 99.98%	Effective for encrypted traffic classification but lacks ability to handle complex anonymous traffic patterns
Milad N et al. [8]	Blind adversarial perturbation algorithm to defeat DNN-based traffic analysis methods	Demonstrated high effectiveness across multiple traffic classifiers	Robustness testing performs poorly
Yesodha K et al. [9]	Intrusion detection system based on FT-ABC-CNN	Low false alarm rate, higher classification accuracy than long short-term memory networks	Limited to generic network features, cannot handle overlapping traffic patterns
Gan B et al. [10]	Intrusion detection method based on CNN-IDMDI	Average binary and multi-class accuracy of 98.73%	Lacks temporal feature extraction, struggles with dynamic and complex behaviors
Xu L et al. [11]	Improved CNN for IoT-enabled security performance prediction	Improves prediction accuracy by 20%	Focused on IoT, does not consider dynamic features of anonymous networks

Yan F et al. [12]	Intrusion detection system based on TL-CNN-IDS	Significantly improves accuracy	Limited datasets, does not address overlapping traffic or anonymous network issues

Combined with Table 1, most studies have some shortcomings while improving the ability of traffic classification and behavior recognition. First, the majority of extant methods prioritize comprehensive network traffic monitoring, yet they are deficient in their capacity to discern intricate and clandestine criminal activities. This is particularly problematic in anonymous network environments, where traditional rule-based matching methods are challenging to implement effectively to detect anomalous behaviors indicative of specific criminal activities. Second, many traffic analysis methods often have a high false alarm rate in practical applications, which makes it difficult for law enforcement agencies to respond quickly when faced with massive alarm information. In addition, these methods have low computational efficiency and are difficult to meet the requirements of real-time monitoring of large-scale network traffic. In view of this, this study introduces the hollow convolution technology in CNN and proposes a Tor overlapping traffic segmentation model based on the hollow convolution convolutional neural network (CNNH). At the same time, combining the attention mechanism, CNN, and LSTM, an online criminal behavior recognition model based on multi-core convolutional neural networks and long short-term memory networks (MCNN-LSTM) is proposed. The model analyzes network traffic characteristics, accurately identifies the websites visited by users, and effectively identifies anomalous network behaviors related to criminal activities, becoming a powerful auxiliary tool for online crime investigation.

The main contributions of the study are as follows: First, the MCNN-LSTM model based on the combination of multi-core convolution and LSTM network is proposed. By using multi-module collaborative optimization, the modeling capabilities of spatial features and time series features are integrated to improve the theoretical framework and method design of network traffic anomaly detection. Second, the self-attention mechanism (SAM) is introduced into the model architecture, which can dynamically focus on key features and improve the model's adaptability to dynamic environments. Finally, a multi-scale feature extraction method is proposed to capture multi-scale spatial features based on the multi-core convolution module.

2 Methods and materials

2.1 Online crime and its challenge

Online criminals often use the anonymity, privacy protection and global characteristics of the Internet to carry out various illegal activities, including illegal gambling, online transactions, money laundering, malware propagation, etc. Studies have shown that the economic losses caused by cybercrime worldwide each year have reached hundreds of billions of dollars, which has brought a huge burden to the global economy [13]. The diversity and complexity of cybercrime make traditional legal supervision and law enforcement methods face huge challenges in dealing with these behaviors.

Among online crimes, online gambling is a relatively common type. Criminals attract users to participate in online gambling activities by setting up and operating illegal gambling websites. These websites usually rely on anonymous networks, such as the Tor network or cryptocurrency payments, which greatly improves their concealment and evades legal supervision. This makes it difficult for law enforcement agencies to track and collect evidence, making it difficult to effectively combat these criminal activities. Online prostitution is also an illegal activity carried out using the Internet. Criminals usually promote and trade through dark web platforms to avoid tracking. In addition, illegal transactions are also an important aspect of online criminal activities. Criminal's trade prohibited items such as drugs, weapons, and counterfeit goods in anonymous markets such as the dark web. Such markets often rely on complex encryption technology and anonymous payment methods to conduct transactions, making it extremely difficult for law enforcement agencies to investigate. Another important form of online crime is the spread of malware. Malware includes ransomware, phishing software, etc., which can be spread through various network channels and pose a serious threat to individuals, enterprises and even government agencies. The spread of malware can not only steal personal privacy information, but also lead to the loss of core corporate data, and in serious cases, even endanger national security. Every year, the number of data leaks caused by malware is huge, and the economic losses caused are difficult to estimate [14]. In addition, with the popularization of IoT technology, cyber attacks on smart devices are also on the rise, further expanding the scope of online criminal activities.

Faced with these challenges, traditional legal and law enforcement methods are unable to cope with the high concealment and transnational nature of online crimes. Researchers and law enforcement agencies have begun to rely on advanced technical means, especially recognition algorithms based on network traffic analysis and deep learning. Through these technologies, researchers can extract useful features from massive amounts of network data to identify and track criminal behavior. In recent years, more and more research has been devoted to improving traffic analysis methods to improve the ability to detect complex cybercrime, especially crimes in anonymous networks. In the future, with the further development of technology, more intelligent detection systems for online criminal behavior will be widely used to better cope with the growing network threats.

Online crime identification is the process of locating and assessing possible illegal activity, such as online gambling, malware distribution, and illegal transactions, by analyzing network traffic, user behavior patterns, and data characteristics. Unlike traditional network traffic analysis, online crime identification focuses more on the complex characteristics of criminal behavior hidden in anonymous networks, often involving protocol abuse, encrypted data streams, and anomalous behavior patterns. Anomalous network behavior usually manifests itself in the form of anomalous network traffic patterns, including but not limited to the following. On the Tor network, high-frequency, short-duration access patterns may reflect scanning attacks. Abnormal packet intervals or excessively large packet sizes may indicate covert channel communications. Sudden changes in traffic characteristics may indicate malware activity. In this context, this study will explore a network traffic analysis method based on deep learning and explore its application potential in identifying anomalous network behavior in the early stage of online crimes.

2.2 CNN-based model construction for tor overlapping traffic segmentation

With the development of anonymous communication technology, Tor network is widely used for both legal and criminalactivities due to its strong anonymity and privacy protection [15]. Tor achieves anonymity in communications by dividing user communications into multiple data packets, transmitting them through multiple relay nodes, and encrypting and decrypting the data packets. The high privacy of anonymous networks makes them an important tool for legitimate users to protect their privacy, but they also provide shelter for various criminalactivities, such as online gambling, online prostitution, and illegal transactions. These crimes not only cause great social harm, but also bring great challenges to law enforcement agencies in identification and tracking. At the same time, this anonymity also makes traffic analysis and identification more difficult, especially in the case of overlapping traffic. Overlapping traffic segmentation refers to the technique of decoupling and segmenting the traffic when the communication data of multiple users are transmitted simultaneously over the same communication link in an anonymous network environment. In contrast to the broader approach of network traffic analysis, the concept of overlapping traffic segmentation entails the identification of the traffic aliasing relationship between disparate users and the extraction of characteristic information from the traffic of particular users. This facilitates the detection of potential abnormal behavior. The flow of traditional overlapping traffic segmentation is shown in the following Figure 1.

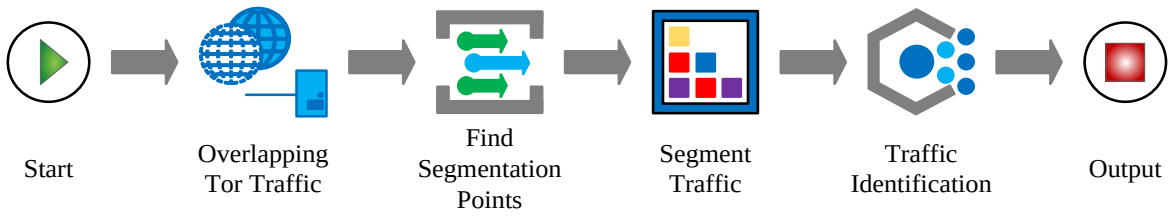


Figure 1: The basic process of overlapping traffic segmentation

As shown in Figure 1, first identify the key segmentation points in the traffic, and use these segmentation points to segment the traffic and extract feature points related to specific behavior patterns. Then, the segmented traffic segments are recognized and classified, and finally further processing is performed based on the recognition results. CNN has a strong feature extraction capability and is suitable for handling overlapping traffic in network traffic. Online criminal activities are often accompanied by complex network traffic patterns that may overlap with normal traffic, increasing the difficulty of identification. By using convolution kernels to extract local features from input traffic, CNN can effectively separate and identify abnormal behavior patterns in overlapping traffic, thereby helping to detect potential criminal activities, such as suspicious transaction requests or abnormal data packet transmissions. Therefore, the study will construct overlapping traffic segmentation model based on CNN. CNN applies convolutional kernel to extract local features by sliding window approach, the calculation is shown in Equation (1) [16].

$$Y_{i,j} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X_{behavior,(i+m),(j+n)} W_{m,n} + b \quad (1)$$

In Equation (1), $Y_{i,j}$ represents the value of the output feature map at position (i, j). $X_{behavior,(i+m),(j+n)}$ is the element in the input feature map. $W_{m,n}$ represents the weight matrix element of the convolution kernel. The training model can automatically adjust the weight to better capture specific behavior features. b is the bias term. M and N denote the height and width of the convolution kernel. Equation (2) illustrates why the rectified linear unit (ReLU), which is easy to understand, quick to compute, and capable of handling deep networks, is chosen as the activation function.

$$f(x) = max(0, x) \quad (2)$$

In Equation (2), x denotes the value input to the activation function after the convolution operation. In the next pooling stage, the expression is shownin Equation (3) [17].

$$P = \max_{window}(X_{behavior\_feature}) \tag{3}$$

In Equation (3), P is the maximum value of the pooling window. $X_{behavior\_feature}$ is the element of the input feature map, which includes behavior features extracted from network traffic such as transmission frequency and directional features. Through downsampling, the pooling procedure shrinks the FM's size, lowering computational cost and enhancing the model's resilience. Finally, the fully connected layer (FCL) expression is shown in Equation (4) [18].

$$\begin{cases} z = Wz' + b \\ z' = \left[z'_{trade}, z'_{malware}, z'_{anomaly} \right] \end{cases} \tag{4}$$

In Equation (4), z' is the input high-dimensional feature vector, and W is the weight matrix of the fully connected layer. z' contains a combination of multiple behavioral features, and z'_{trade}, $z'_{malware}$, and $z'_{anomaly}$ represent features related to illegal transactions, malware propagation, and other abnormal behaviors, respectively. Among the most often utilized loss functions in classification problems is the cross-entropy loss function. Equation (5) illustrates its expression by calculating the difference between the probability distribution (PD) of the real labels and the PD predicted by the model.

$$L = -\sum_{i=1}^{N} w_{behavior\_feature}(y_i \log(\hat{y}_i)) \tag{5}$$

In Equation (5), L is the loss value. $w_{behavior\_feature}$ represents the weight factor related to the behavior characteristics. N represents the number of samples, y_i is the true label, and i represents the actual category corresponding to the sample, that is, whether it is an illegal activity. $\hat{y}_i$ is the probability distribution predicted by the model. By adding the weight factor of the behavior characteristics, the model can more effectively focus on the characteristics related to the criminal behavior, thereby improving the recognition effect of the model in specific criminal behavior scenarios.

Due to the highly encrypted and complex time series characteristics of Tor traffic, the study introduces the hollow convolution technique, by introducing cavities in the convolution kernel. That is, extending the receptive field without adding more parameters by adding gaps between the convolution kernel's parts. Hollow convolution (also known as expanded convolution) is a technique that expands the receptive field by inserting holes in the convolution kernel, capturing a wider range of features without increasing the number of parameters. This method helps the model handle long-range dependencies while maintaining computational efficiency, as shown in Figure 2.

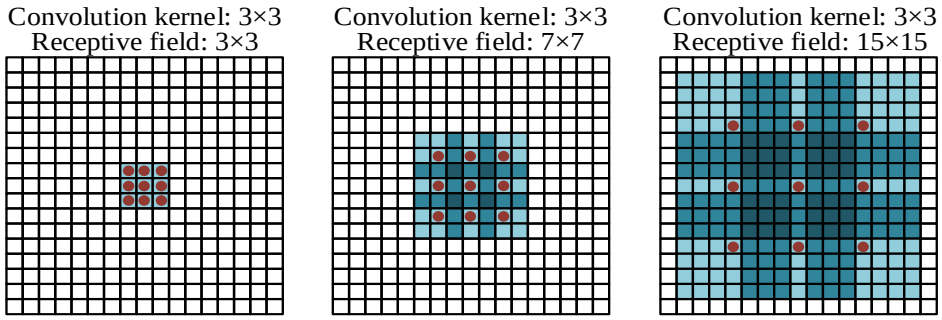


Figure 2: Multi-scale feature extraction using dilated convolution

Figure 2(a), (b), and (c) represent the convolutional kernel arrangement with convolutional expansion rate of 1, 2, and 4, respectively. In Figure 2(a), when the convolutional expansion rate is 1, the convolutional kernel size is 3×3, which is the same as the conventional convolutional kernel, and the sensory field only covers the local area. In Figure 2(b) with a convolution expansion rate of 2, the convolution kernel sense field expands to 7×7, but the actual parameters remain 3×3. In Figure 2(c) with a convolution expansion rate of 4, the sense field further expands to 15×15, and the number of parameters remains the same. Null convolution can effectively extract multi-scale information and remote-dependent features without increasing the computational complexity, and is suitable for processing complex features in Tor traffic.

In the rest of the model, batch normalization is first introduced after each convolutional layer to accelerate convergence and improve generalization. Second, a larger range of contextual information is captured by expanding the sensory field by the application of null convolution. Moreover, to prevent overfitting, a Dropout layer is introduced to enhance model robustness. Furthermore, to better handle the complex aspects of Tor traffic, a deep network structure is built by stacking numerous convolutional, pooling, and FCLs. Therefore, the structure of the overlapping traffic segmentation model of CNNH is shown in Figure 3 below.

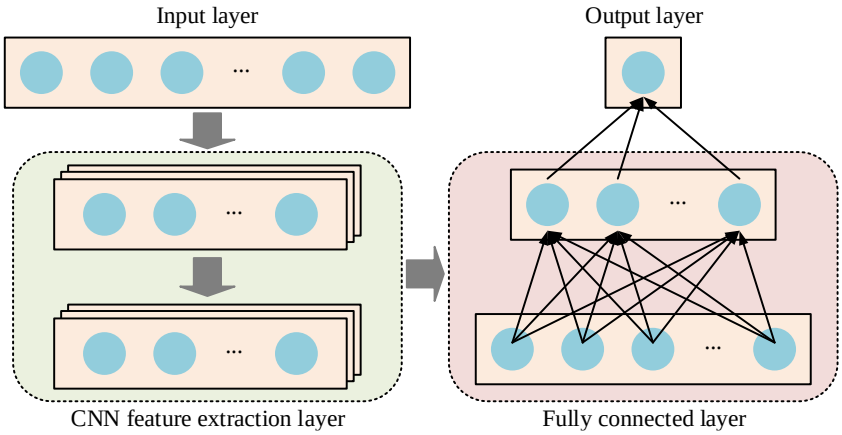


Figure 3: Overlapping traffic segmentation model based on CNNH

As shown in Figure 3, the process of the CNNH overlapping traffic segmentation model consists of four parts. First, the input layer receives the original Tor traffic data and passes it to the CNN layer. The CNN layer extracts representative features from the input traffic through a series of convolution operations and pooling operations, including network behavior features such as packet size, transmission time interval, and transmission frequency. These extracted features are then passed to the FCL, where the features are further comprehensively analyzed to generate a high-dimensional feature vector. Finally, the output layer completes the prediction and classification of the traffic segmentation results based on the output of the FCL, helping the model distinguish between legitimate traffic and potential criminal behavior. To facilitate understanding of the specific implementation of the CNNH model, the pseudo code is given below, as shown in Figure 4.

```
# Pseudocode for CNNH Model
# Input: Network traffic data (X), labels (Y)
# Output: Predicted labels (Y_hat)

# Step 1: Data Preprocessing
X_preprocessed = preprocess_data(X)  # Normalize and extract features

# Step 2: Dilated Convolution (Hollow Convolution) Module
def DilatedCNN_Module(X):
    Conv1 = Conv2D(filters=32, kernel_size=(3, 3), dilation_rate=1, activation='relu')(X)
    Conv2 = Conv2D(filters=64, kernel_size=(3, 3), dilation_rate=2, activation='relu')(Conv1)
    Conv3 = Conv2D(filters=128, kernel_size=(3, 3), dilation_rate=4, activation='relu')(Conv2)
    PooledFeatures = MaxPooling2D(pool_size=(2, 2))(Conv3)
    return PooledFeatures

X_dilated = DilatedCNN_Module(X_preprocessed)

# Step 3: Fully Connected Layers for Classification
def ClassificationHead(X):
    Dense1 = Dense(units=64, activation='relu')(X)
    Output = Dense(units=num_classes, activation='softmax')(Dense1)
    return Output

Y_hat = ClassificationHead(X_dilated)

# Step 4: Model Training
model = compile_model(optimizer='adam', loss='categorical_crossentropy')
model.fit(X_preprocessed, Y, epochs=50, batch_size=32)
```

Figure 4: Overlapping traffic segmentation model based on CNNH

The pseudo code in Figure 4 shows the workflow of the CNNH model in complex network traffic feature extraction. The model effectively expands the receptive field through the hole convolution module. Therefore, it can reduce information loss while maintaining the integrity of spatial features.

2.3 Research on online criminal behavior recognition model based on LSTM and CNN

CNN for traffic segmentation, although excellent in spatial feature extraction, still suffers from recognition limitations when confronted with time-series features in Tor traffic. In contrast, LSTM, as a recurrent neural network that excels in processing sequence data, is suitable for the field of network traffic analysis due to its powerful modeling capability of time series features [19]. In online criminal behaviors, such as cyber attacks or illegal transactions, specific time patterns are often shown, such as persistent illegal access attempts or regular small-amount fund transfers. By analyzing the time series features in network traffic, LSTM can identify the regularity of these criminal behaviors and provide support for crime prevention by predicting future behavior trends. Therefore, the study will try to combine CNN and LSTM and introduce the SAM to extract and classify important features.

In the overall process design, the input data is first processed through a data encoding module to convert the raw data into a form suitable for model input. Then, it is passed through the SAM module in order to enhance the attention to the key features. Then, CNN and LSTM modules perform feature extraction and time series analysis on the data processed by the attention mechanism, to capture behavioral patterns that recur over long periods of time. Finally, the model outputs the recognition results to realize the recognition of WF. In the data encoding module, the training data is shown in Equation (6).

$$\begin{cases} T = \{(X_1, G_1), (X_2, G_2), ..., (X_n, G_n)\} \\ X = (1, -1, 1, -1, ..., 1) \end{cases} \quad (6)$$

In Equation (6), T denotes the training data set. X_n and G_n denote the n th traffic instance and website class label, respectively. One-Hot encoding, a popular encoding technique in neural network multi-classification tasks, is crucial for guaranteeing the classification model's accuracy, preventing label misrepresentation, and increasing computational efficiency. Therefore, One-Hot state bits are used for encoding. Further, in the SAM module, the correlation matrix of the input sequence is first calculated as shown in Equation (7) [20].

$$\begin{cases} V = X \cdot W_v \\ K = X \cdot W_k \\ Q = X \cdot W_q \end{cases} \quad (7)$$

In Equation (7), V , K , Q denote the value, key, and query matrices, respectively. W_v , W_k , and W_q all denote the initial weight matrices, which correspond to the value, key, and query weight matrices, respectively. These matrices project the input sequences into different vector spaces for subsequent computation of the attention scores. The attention score is shown in Equation (8).

$$attention(a, V) = \sum_{j=1}^{j=n} a_i \cdot v_j i \quad (8)$$

In Equation (8), a_i denotes the i th attention weight. v_j is the element at the j th position in the value vector V . Thus, Figure 5 shows a schematized version of the SAM structure.

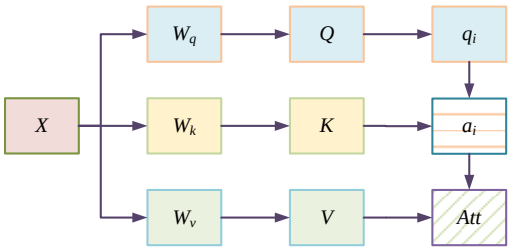


Figure 5: Self attention mechanism layer structure

In Figure 5, the input sequence X is first converted to value matrix V , key matrix K , and query matrix Q through three weight matrices W_v , W_k , and W_q , respectively. Then, the Q and K calculate the correlation through dot-product operation, and the result is inputted into the Softmax function (SF) to generate the attention weights a_i after scaling. These attention weights are used to weight the corresponding elements in the V elements, and finally the weighted value matrix is passed through a summation operation to obtain the output. By using this approach, the model can dynamically concentrate on important features according to how important each segment of the input sequence is. This successfully boosts the model's performance when processing complex data and improves its capacity to capture vital information. Finally, in the CNN and LSTM module, the resulting feature sequence is spliced into a two-dimensional feature matrix. Then the one-dimensional maximum pooling layer (PL) is connected for data dimensionality reduction processing, and the expression is shown in Equation (9).

$$Y_{i,h=5}^{l} = \max(Z_{j-1}^{l}, Z_{j}^{l}, Z_{j+1}^{l}, Z_{j+2}^{l}) \quad (9)$$

In Equation (9), $Y_{i,h=5}^{l}$ then denotes the result of the pooling operation via the convolution kernel of 5. Z_{j}^{l} , Z_{j+2}^{l} , and Z_{j+2}^{l} all denote the neighboring feature values in the previous layer of l . Subsequently, the extracted spatial features are fused as shown in Equation (10).

$$F_j^l = concat(Y_{i,h=3}^{l}, Y_{i,h=4}^{l}, Y_{i,h=5}^{l}) \quad (10)$$

In Equation (10), F_j^l denotes the fused features after convolution and pooling. $Y_{i,h=3}^{l}$, $Y_{i,h=4}^{l}$, and $Y_{i,h=5}^{l}$ denote

the i th output of the pooling of the l th layer with convolution kernel size 3, 4, and 5, respectively. Equation (11) illustrates how the data is put into the LSTM to extract the temporal features once the fusion is finished.

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b) \quad (11)$$

In Equation (11), h_t and h_{t-1} represent the hidden states of the current time step and the previous time step, respectively, that is, the contextual information of the behavioral features at the current moment. For identifying criminal behavior, the information of the previous time step, such as the occurrence of certain abnormal behaviors at the previous moment, can help predict whether the behavior at the current moment is abnormal. x_t represents the input features of the current time step, and W_h represents the weight matrix of the hidden state, which can learn how to transfer the criminal behavior features of the previous moment to the current moment. W_x is the weight matrix of the input features, which is used to weight the input features of the current time step. These weights can learn the importance of different behavioral features in predicting criminal behavior. b is the bias term, and σ is the activation function. By introducing nonlinearity, the model can capture complex behavioral patterns. Finally, the model further fuses spatial features, temporal features and behavioral features to form a unified feature representation. Specifically, spatial features are extracted through the convolution layer, temporal features are captured through the LSTM layer, and behavioral features are extracted based on high-risk behavior patterns in traffic. The fused feature representation is shown in Equation (12).

$$z = \alpha z_{spatial} + \beta z_{temporal} + \gamma z_{behavioral} \quad (12)$$

In Equation (12), $z_{spatial}$ represents the spatial features extracted by the convolution layer, which can help identify local anomalies in network traffic. $z_{temporal}$ represents the temporal features extracted by the LSTM layer, which captures recurring patterns in the time dimension, especially high-frequency packet transmission behaviors. $z_{behavioral}$ represents the high-level features obtained by the behavioral feature extraction mechanism, which reflects specific behavioral patterns such as malware propagation and illegal transactions. α , β , and γ are all weighting factors. The weights are adjusted according to the importance of different features to ensure the sensitivity of the model to specific behavioral patterns. Therefore, the improved CNN-LSTM structure is shown in Figure 6 below.

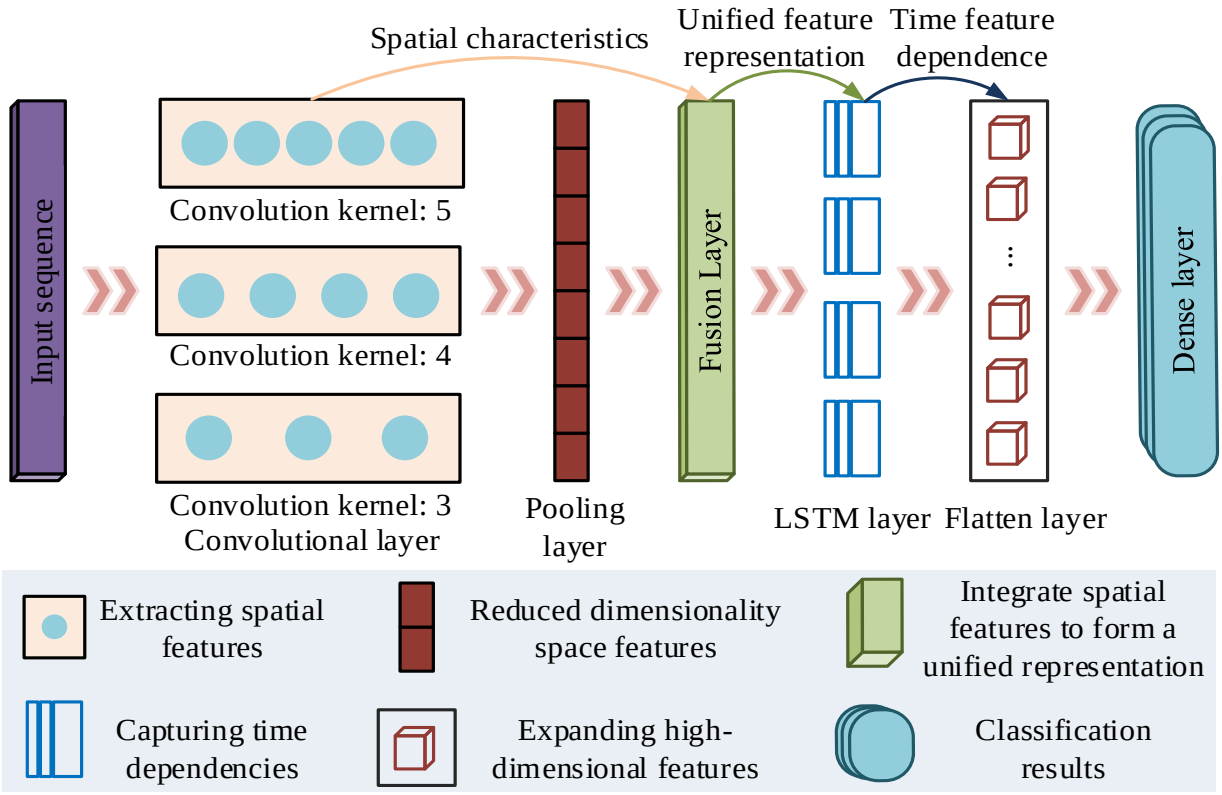


Figure 6: The structure and temporal feature fusion of the MCNN-LSTM Model

In Figure 6, first, the input sequence passes through multiple convolutional layers, each with a different convolutional kernel size to capture different scale features in the input data. Subsequently, a PL is used to downsample the convolved FMs, so decreasing their size and, consequently, the computational complexity. Then, the multi-scale features are integrated through a fusion layer to form a unified feature representation, helping the model capture more comprehensive traffic information. Immediately afterward, these features are passed to the LSTM layer. The LSTM layer specializes in processing time-series data and is able to capture long-range dependencies in the data. Subsequently, the high-dimensional features output from the LSTM layer are expanded into one-dimensional vectors through the Flatten layer. Finally, the output of the classification or regression task is carried out through the FCL, thereby identifying potential criminal behavior in network traffic. Therefore, according to the above calculations, the online criminal behavior recognition process based on MCNN-LSTM is shown in Figure 7.

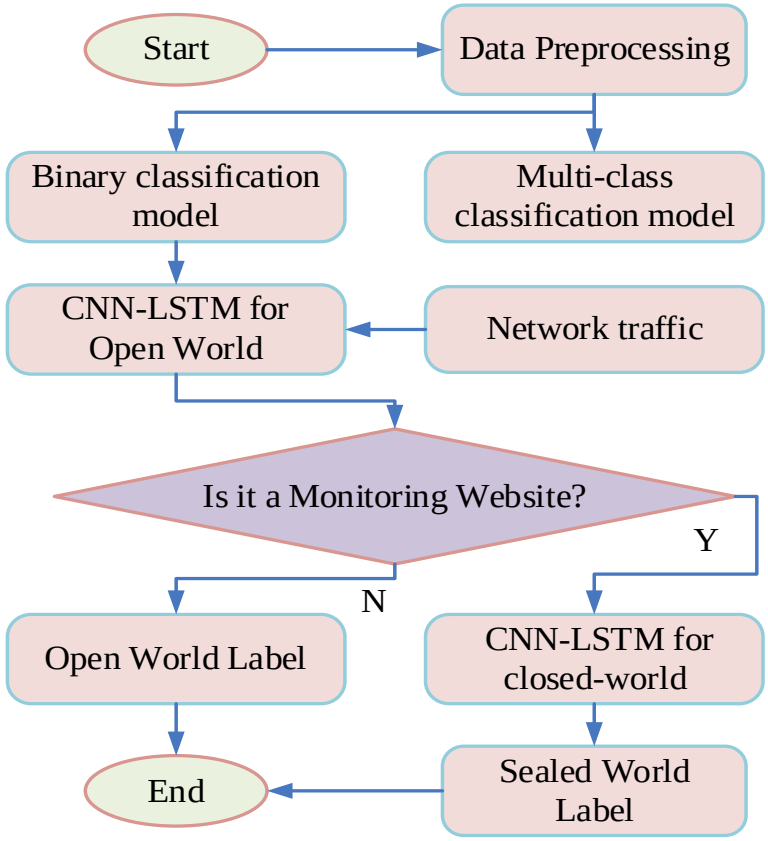


Figure 7: Online criminal behavior identification process

As shown in Figure 7, first, during the training phase, the preservation model is trained using the binary and multi-classification datasets created from the network traffic data, respectively. In the recognition phase, the input network traffic is first processed by the open-world MCNN-LSTM to determine whether it is labeled in the accusation domain. If the traffic belongs to the accusation domain, it enters the open-world label processing and is recognized using the closed-world MCNN-LSTM. If it does not belong to the accusation domain, it enters the closed-world labeling processing. Through the staged processing, the model is able to process the open-world and closed-world labels separately, thus improving the accuracy and efficiency of the recognition. To intuitively demonstrate the implementation process of the MCNN-LSTM model, its pseudo code is given below, as shown in Figure 8.

```
# Pseudocode for MCNN-LSTM Model
# Pseudocode for MCNN-LSTM Model
# Input: Network traffic data (X), labels (Y)
# Output: Predicted labels (Y_hat)

# Step 1: Data Preprocessing
X_preprocessed = preprocess_data(X)  # Normalize and extract features

# Step 2: Multi-Scale Convolution (MCNN) Module
def MCNN_Module(X):
    Conv1 = Conv2D(filters=32, kernel_size=(3, 3), activation='relu')(X)
    Conv2 = Conv2D(filters=64, kernel_size=(5, 5), activation='relu')(Conv1)
    Conv3 = Conv2D(filters=128, kernel_size=(7, 7), activation='relu')(Conv2)
    CombinedFeatures = concatenate([Conv1, Conv2, Conv3])
    PooledFeatures = MaxPooling2D(pool_size=(2, 2))(CombinedFeatures)
    return PooledFeatures

X_spatial = MCNN_Module(X_preprocessed)

# Step 3: Temporal Feature Extraction with LSTM
def LSTM_Module(X):
    LSTM_output = LSTM(units=128, return_sequences=True)(X)
    return LSTM_output

X_temporal = LSTM_Module(X_spatial)

# Step 4: Self-Attention Mechanism (SAM)
def SelfAttention(X):
    Q = dot(X, Wq)  # Query matrix
    K = dot(X, Wk)  # Key matrix
    V = dot(X, Wv)  # Value matrix
    AttentionScores = Softmax(dot(Q, K.T) / sqrt(d_k))  # Scaled Dot-Product Attention
    Output = dot(AttentionScores, V)  # Weighted sum of values
    return Output

X_attention = SelfAttention(X_temporal)

# Step 5: Fully Connected Layers for Classification
def ClassificationHead(X):
    Dense1 = Dense(units=128, activation='relu')(X)
    Output = Dense(units=num_classes, activation='softmax')(Dense1)
    return Output

Y_hat = ClassificationHead(X_attention)

# Step 6: Model Training
model = compile_model(optimizer='adam', loss='categorical_crossentropy')
model.fit(X_preprocessed, Y, epochs=50, batch_size=32)
```

Figure 8: Schematic diagram of MCNN-LSTM pseudo code

This pseudo code in Figure 8 clearly shows the main modules of the MCNN-LSTM model and their interaction process. First, the multi-core convolution module captures the multi-scale features of the input data and combines the pooling layer to reduce the computational complexity. Subsequently, the LSTM module is employed to model the time series features, with the self-attention mechanism further emphasizing the key features to enhance the classification performance. Finally, the network traffic classification is completed by the fully connected layer.

3 Results

3.1 Performance testing of overlapping traffic segmentation model for CNNH

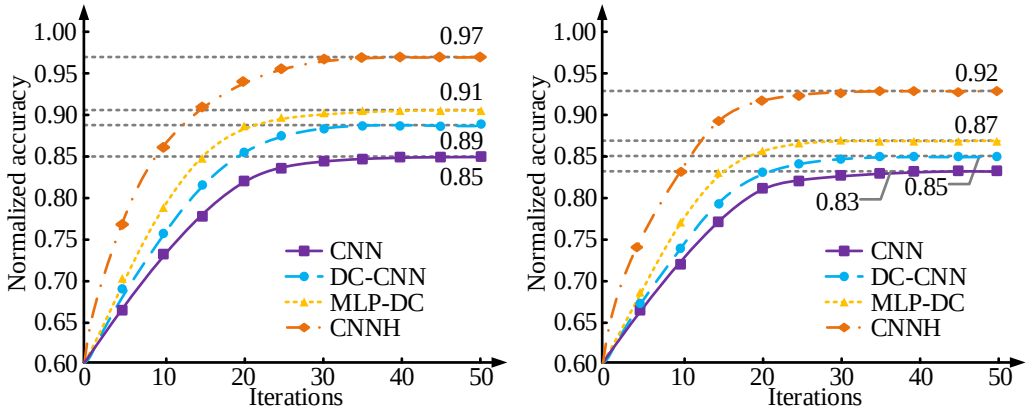
The study began by setting up a suitable experimental environment to meet the computational requirements of the experiment. The experiment uses Windows 10 operating system with a 12-core Xeon Platinum 8163 processor and a graphics card NVIDIA Tesla P100-16GB. The model development language is Python 3.7. The study selects the CW200 dataset as the experimental object, which contains a variety of normal and abnormal traffic with high noise and complex traffic patterns, meeting the needs of overlapping traffic segmentation and abnormal behavior identification in anonymous networks. The diversity of protocol distribution and user behavior is taken into account during data collection in order to mimic traffic patterns in real-world scenarios as closely as possible. The dataset collects traffic data from 200 different websites accessed through the Tor network in a closed world. Each site has 2,500 traffic accesses, which are divided into training and test sets in a 6:4 ratio. A stratified sampling method is used to ensure that the proportions of the training and test sets are consistent in terms of protocol type, traffic feature distribution, and attack type, thus avoiding the bias of the model performance evaluation due to uneven

data distribution. In addition, to reduce the risk of overfitting, the dropout regularization technique is introduced into the experiment, and the diversity of the training data is improved by data enhancement. Among them, normal traffic accounts for 60%, abnormal traffic accounts for 40%, and is further divided into four categories: Trojans, Worms, Viruses, and Adware. The proportion of category samples is balanced and covers a variety of network protocols and anonymous network scenarios. Data cleaning, feature extraction, and standardization are performed during data preprocessing, and traffic behavior patterns are labeled as normal or abnormal. First, the settings of each parameter in CNNH are shown in Table 2 below.

Table 2: Model parameter settings

Parameter	Value
Input dimension	5000
Network architecture layers	12
Batch size	256
Epochs	50
Gradient optimization function	Adam
Learning rate	0.001
Dropout	0.4

Table 2 shows the settings for input dimension, network architecture layers, training details, optimizer, learning rate and Dropout rate, respectively. The study uses CNN, dilated CNN (DC-CNN), and multi-layer perceptron with dilated convolution (MLP-DC) as comparison models. When criminal activities are carried out in anonymous networks, criminal behavior is often hidden in normal traffic. A high segmentation accuracy means that the model can more accurately distinguish normal network behavior from potential criminal behavior, and can more accurately capture traffic patterns related to criminal activities such as illegal transactions and malware propagation, thereby reducing false positives and improving the effectiveness of crime identification. Therefore, the traffic segmentation accuracy is used as an indicator, and the test results are shown in Figure 9.



(a) Training set (b) Test set

Figure 9: Accuracy trends on training and test sets for different models

Figure 9(a) and (b) show the accuracy of CNN, DC-CNN, MLP-DC, and CNNH with the iterations on the training set and test set, respectively. In the case of malware propagation, the model identified multiple suspicious data packets through high-precision traffic segmentation. The transmission frequency and time characteristics of these data packets are highly consistent with known malware propagation behaviors, thereby enabling law enforcement to swiftly identify the source of the behavior.

In Figure 9(a), when the number of iterations is 50, the accuracy of CNN, DC-CNN, MLP-DC and CNNH models on the training set is 0.85, 0.89, 0.91, 0.97, respectively. In Figure 9(b), the accuracy of the four models on the test set are 0.83, 0.85, 0.87, 0.92, respectively. DC-CNN and MLP-DC introduce the advantage of null convolution to extract deep features more comprehensively. To verify whether the difference in accuracy between different models on the training and test sets is statistically significant, a paired *t*-test is performed on the normalized accuracy and the 95% confidence interval is calculated, as shown in Table 3.

Table 3: Statistical significance analysis

Dataset	Model Comparison	Normalized accuracy difference (%)	95% confidence interval (%)	*P*-value	Statistical significance
Training set	CNNH vs. CNN	12	[10.2, 13.8]	< 0.01	Significant
	CNNH vs. DC-CNN	8	[6.4, 9.6]	< 0.05	Significant
	CNNH vs. MLP-DC	6	[4.7, 7.3]	< 0.05	Significant
Testing set	CNNH vs. CNN	9	[7.5, 10.5]	< 0.01	Significant
	CNNH vs. DC-CNN	7	[5.6, 8.4]	< 0.05	Significant
	CNNH vs. MLP-DC	5	[3.8, 6.2]	< 0.05	Significant

The statistical analysis results in the table show that the accuracy improvement range of CNNH on the training set is from +6.0% to +12.0%, and the improvement range on the test set is from +5.0% to +9.0%. The *P*-values for all comparisons are less than 0.05, indicating statistical significance. In addition, the 95% confidence interval indicates that the range of differences is relatively stable. Subsequently, the segmentation effect of each model under different traffic flows is shown in Table 4 below.

Table 4: Performance evaluation indicators for each algorithm

Index	CNN		DC-CNN		MLP-DC		CNNH	
	Normal	Attack	Normal	Attack	Normal	Attack	Normal	Attack
P/%	83.52	85.67	86.14	88.43	88.79	90.57	91.43	93.45
R/%	84.67	86.82	88.53	89.92	90.35	91.74	93.46	94.32
FPR/%	13.65	12.34	10.74	9.98	8.96	7.43	4.15	3.07
F1/%	84.09	86.24	87.32	89.17	89.56	91.15	92.43	93.88
AUC	0.769	0.788	0.812	0.828	0.839	0.846	0.928	0.935
Time/s	12.34	13.02	15.89	16.58	19.65	20.23	18.41	19.12
Resource consumption/%	68.54	69.85	72.32	73.46	75.69	76.78	70.17	71.54

Table 4 displays the performance comparison of the models for segmentation under Normal and Attack traffic. False positive rate (FPR) is critical in law enforcement contexts, as a high FPR could lead to misidentifying benign traffic as criminal activity, resulting in wasted resources. The CNNH model has significantly higher values for P, R, F1 and AUC. Especially, the *P*-value of CNNH reaches 93.45% and the R value is 94.32% under Attack traffic. Meanwhile, the FPR of CNNH is only 4.15%, indicating that it can effectively reduce the false alarms. However, with the increase of model complexity, the resource consumption rate and calculation time of CNNH increase accordingly, reaching 71.54% and 19.12s, respectively. Although its resource requirements are high, the significant improvements in accuracy and sensitivity make up for this shortcoming. In contrast, the traditional CNN is at a lower level in all performance indicators. However, its resource consumption rate and computation time are low, which makes it suitable for scenarios with limited computational resources. The proposed model has been demonstrated to effectively reduce the FPR, ensuring higher accuracy and reliability in identifying criminal behavior. Furthermore, it has been shown to facilitate the optimization of resource allocation and action decisions.

In the experiment, the recall rate is equivalent to the sensitivity, i.e., the proportion of actual anomalous traffic that is correctly detected. In practical scenarios, this balance of performance is critical. Maintaining sensitivity ensures that abnormal behavior is not ignored due to low detection capabilities. Further analysis of the experimental results shows that false positives occur mainly in normal traffic with high access frequency, such as normal data transmission of certain legitimate protocols being misclassified as abnormal traffic. This may be due to the similarity between the characteristics of high-frequency access patterns and abnormal traffic. False negatives, on the other hand, are mainly concentrated on abnormal traffic with weaker characteristics or close to normal traffic characteristics, such as covert adware traffic. False positives can lead to reduced efficiency in resource allocation, while false negatives can cause some potential threats to be ignored.

3.2 Online crime recognition experiment based on MCNN-LSTM

In the hyperparameter setting of MCNN-LSTM, the learning rate is optimized in the range of 0.0001 to 0.01 by grid search and finally selected as 0.001. The batch size is set to 32. The number of hidden layer nodes is set to 128, which can effectively capture the time series characteristics of traffic data. The time step is set to 20. Adam is used as the optimizer to improve the training efficiency. The number of training rounds is set to 50, and the early stopping strategy is combined to avoid

overfitting. To improve the generalization ability of the model, Dropout is added to the network and the ratio is set to 0.3. The study labeled the traffic data set according to different crime types, mainly including three types of crimes: online fraud, malware propagation, and illegal transactions. multi-layer perceptron convolutional neural network (MLP-CNN), long short-term memory with attention mechanism (LSTM-Att), and LSTM are selected as comparison algorithms. First, the accuracy test results of the four models for different types of online criminal behaviors are shown in Figure 10 below.

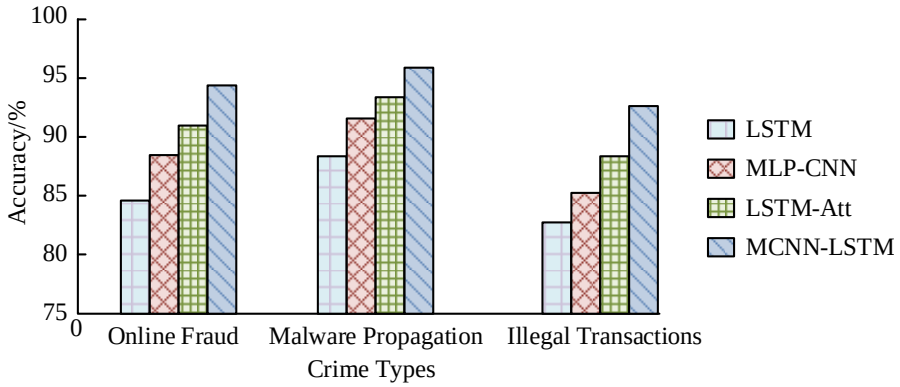


Figure 10: Model performance in crime type identification

In Figure 10, the MCNN-LSTM model showed the best accuracy, especially in the identification of malware propagation and illegal transactions, reaching 96.54% and 92.87% respectively. This is because MCNN-LSTM combines the spatial feature extraction capability of CNN with the temporal feature capture capability of LSTM, and can better handle the complex patterns and temporal dependencies in criminal behavior. Although LSTM-Att improves the focus on important features by introducing the attention mechanism, its spatial feature extraction capability is weak, so it is still inferior to MCNN-LSTM in multi-dimensional feature extraction. LSTM has the worst performance among the three crime types, especially in the identification of malware propagation, which is only 87.43%. Subsequently, to evaluate the performance of each model in crime prediction and prevention, the following indicators are used: prediction accuracy, early warning time (early warning time is defined as the time interval between the first detection of an abnormal traffic pattern by the model and the actual occurrence of the attack behavior), precision, FPR, mean detection time, and area under the receiver operating characteristic curve (AUC). The results are shown in Table 5 below.

Table 5: Performance comparison of models in crime prediction and early warning tasks

Metrics	LSTM	MLP-CNN	LSTM-Att	MCNN-LSTM
Prediction Accuracy /%	80.45	84.67	88.76	92.43
Average Early Warning Time /Minutes	15	18	25	30
Precision /%	79.87	83.54	87.34	91.23
False Positive Rate /%	9.67	8.23	6.45	5.12
Mean Time to Detect /Seconds	42.8	35.6	28.1	24.3
AUC	0.835	0.874	0.915	0.945

In Table 5, MCNN-LSTM shows the best comprehensive performance. Compared with other models, MCNN-LSTM achieved a prediction accuracy of 92.43%, which is significantly higher than LSTM's 80.45% and MLP-CNN's 84.67%. Although LSTM-Att introduces the attention mechanism, its spatial feature extraction capability is insufficient, resulting in the advance warning time and prediction accuracy being inferior to MCNN-LSTM. In addition, MCNN-LSTM can warn of criminal behavior 30 minutes in advance, this capability is mainly due to the model's deep modeling of time series characteristics. In particular, the introduction of the SAM further enhances the model's ability in key feature extraction, enabling it to quickly focus on abnormal behavior features and reduce the interference of irrelevant features. The accuracy of MCNN-LSTM is also better than other models, with the lowest false positive rate of only 5.12%. The model performs well in reducing false positives. In contrast, LSTM has a higher false positive rate of 9.67%, due to the lack of spatial feature modeling, its adaptability to traffic pattern changes is poor and the false alarm rate is significantly high. In terms of average detection time, the MTTD of MCNN-LSTM is 24.3 seconds, which is better than 28.1 seconds of LSTM-Att and 35.6 seconds of MLP-CNN, which further proves the real-time detection capability of the model.

Finally, the concept of concept drift is used to evaluate the robustness and adaptability of the models in the face of changing data distributions. Conceptual drift refers to the phenomenon that the data distribution changes over time. Its practical application is due to the fact that the traffic pattern, feature distribution, and user behavior of a website may change over time. The drift simulation involves the gradual adjustment of the ratio of normal traffic to abnormal traffic, thereby reflecting the dynamic changes in network attack behaviors. The protocol-related features (e.g., packet length, time interval) are subject to random changes, thereby simulating fluctuations in protocol usage and traffic characteristics. Furthermore, the incorporation of novel attack types at various temporal points serves to mirror the progression of attack patterns. These designs are intended to closely mirror the evolving trends in the actual network environment, thereby facilitating the evaluation of the model's efficacy in handling long-term distribution shifts. The results are shown in Figure 11.

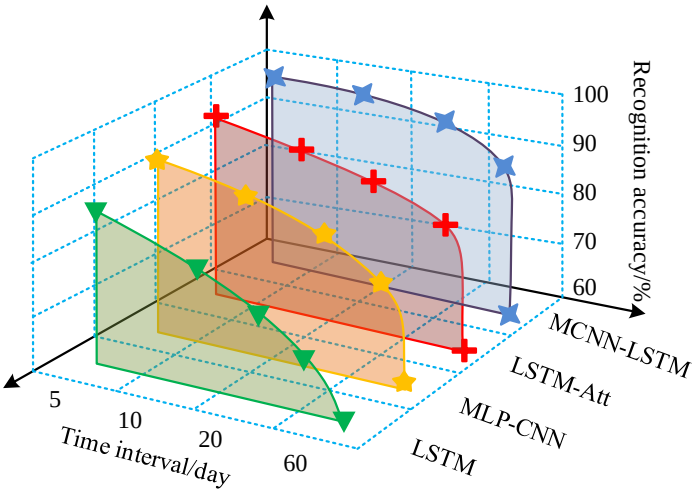


Figure 11: Impact of concept drift on model accuracy over time

Figure 11 shows the model accuracy of the four models under training time and testing time of 5, 10, 20, and 60, respectively. As the interval time increases, the concept drift leads to different degrees of decrease in the accuracy of each model. A lower drop value suggests that the model is more flexible and can continue to perform well in classification even when concepts diverge. When the interval between training and testing events is 60 days, the recognition accuracies of LSTM, MLP-CNN, LSTM-Att, and MCNN-LSTM models are 60.2%, 73.8%, 80.7%, and 89.5%, respectively. The advantage of MCNN-LSTM in dynamically changing environments lies in its optimized model architecture. The multi-kernel convolution module extracts multi-scale spatial features by convolution kernels of different sizes. The SAM dynamically focuses on key features to reduce interference. It works in conjunction with time series modeling to significantly improve adaptability to dynamic changes in the traffic feature distribution. In contrast, LSTM lacks spatial feature extraction capabilities and relies only on time series modeling, resulting in high sensitivity to changes in traffic patterns and rapid loss of accuracy. MLP-CNN is biased toward fixed patterns in feature extraction and has insufficient adaptability to concept drift.

Finally, several representative models including time-series Transformer, spatial-temporal graph convolutional network and transformer framework (ST-GCN+Transformer), bidirectional long short-term memory with attention mechanism (BiLSTM+Attention), random forest and principal component analysis (RF+PCA), and K-nearest neighbor (KNN) are selected for comparison. These five models cover the hybrid framework and transformer method in modern deep learning, as well as the classic algorithms of traditional machine learning and non-deep learning. It can fully reflect the advantages and disadvantages of different technical routes in network traffic analysis. The dataset used is the representative open world network traffic dataset CIC-IDS2017. It records normal traffic and 12 malicious attack behaviors, has 80 traffic features, and has highly complex traffic patterns and open network environment characteristics. The results are shown in Table 6.

Table 6: Performance comparison and scalability testing of models under different traffic loads

Traffic condition	Model Name	Accuracy /%	FPR /%	Average processing time (ms/sample)	Accuracy at 300% data expansion /%	*P*-value
Small traffic (10% data)	MCNN-LSTM	96.78	2.95	18.6	94.12	< 0.05
	Time-series Transformer	95.23	3.21	17.5	92.45	< 0.05
	ST-GCN+Transformer	95.78	3.1	20.9	93.34	< 0.05
	BiLSTM+Attention	92.67	4.89	19.2	89.45	< 0.05
	Random Forest+PCA	88.23	7.34	14.7	86.34	< 0.05
	KNN	84.12	9.78	15.9	82.45	< 0.05
Medium traffic (50% data)	MCNN-LSTM	95.89	3.45	20.8	93.34	< 0.05
	Time-Series Transformer	94.12	3.89	18.3	91.67	< 0.05
	ST-GCN+Transformer	94.78	3.56	21.2	92.78	< 0.05
	BiLSTM+Attention	90.78	5.45	19.6	88.01	< 0.05
	Random forest+PCA	86.34	8.12	15.2	84.78	< 0.05
	KNN	82.45	10.78	16.5	79.67	< 0.05
High traffic (100% data)	MCNN-LSTM	94.78	3.89	22.5	92.12	< 0.05
	Time-series Transformer	93.12	4.12	19.9	90.56	< 0.05
	ST-GCN+Transformer	93.78	4.01	22.8	91.45	< 0.05
	BiLSTM+Attention	89.34	6.12	20.3	87.12	< 0.05
	Random forest+PCA	84.89	9.34	15.8	82.45	< 0.05
	KNN	81.12	11.78	16.7	78.34	< 0.05

In table 6, MCNN-LSTM shows high accuracy in all traffic load scenarios, reaching 96.78% in small traffic scenarios and maintaining 94.78% in high traffic scenarios. Moreover, it demonstrated strong classification capabilities with an FPR of 3.89%. This is mainly due to its multi-module synergy combining CNN and LSTM, which can effectively capture the complex relationship between spatial and temporal features. Time-series Transformer and ST-GCN+Transformer also perform similarly in terms of FPR and accuracy. The global modeling capabilities of these two models allow them to perform well in dynamic network scenarios. The accuracy of the BiLSTM+attention model is subject to a significant decrease in high-traffic scenarios due to the limitations of the feature extraction method. In contrast, the KNN and Random Forest methods demonstrate a higher degree of suitability for small-scale data sets. However, when the data is expanded to 300%, the accuracy undergoes a substantial decline, indicating a lack of adaptability to large-scale, complex scenarios.

3.3 Simulation test

In online criminal behavior on anonymous networks, the illegal software trading market is active, and many websites specialize in the illegal sale of pirated software. The illegal sale of pirated software not only violates intellectual property laws, but also involves illegal transactions and fund transfers through anonymous networks, which is a common and widespread form of online crime. Such websites conduct transactions through encrypted networks and anonymous payment systems, and users can purchase unauthorized commercial software, hacking tools, and cracked software. On one of the websites, called Dark Web Software Mall, about 4,000 users visit and trade every day. The website uses encrypted communication protocols and anonymous payment methods such as Bitcoin.

The experiment uses web crawler technology to capture network traffic data from the website for 10 days, with a total of 400,000 packets, of which 200,000 packets are directly related to illegal software transactions, including user login, browsing illegal software, ordering, and anonymous payment. At the same time, for comparison, the study also obtains traffic data from legal e-commerce platforms in the same period, totaling 150,000 packets, which are related to browsing and purchasing legal software. Traffic capture and model training are performed on a server running the Linux

operating system, with a 16-core CPU, 32GB memory, and 500GB storage space. The experiment uses the Wireshark tool to capture network traffic to ensure the accuracy and integrity of the data. The traffic data includes parts obtained from legal e-commerce websites and illegal software trading websites, with a total of 400,000 packets. To ensure that the model can effectively identify traffic behaviors related to illegal software transactions, the study preprocesses the data, removed noise, and extracted key features, including packet size, time interval, and transmission direction. By analyzing the traffic characteristics, the model is further used to distinguish the network traffic of legal software transactions, and illegal software sales. The results are shown in Table 7 below.

Table 7: Detection results of illegal software transactions compared to legitimate traffic

Metric	Detection results	Legitimate traffic (control group)	Illegal software transaction traffic
Large-scale software downloads (times/day)	Average of 3.2 detections/day	0.5-1.1 times/day	8.7-12.3 times/day
Frequent small anonymous payments (transactions/day)	Average of 1.8 transactions/day	1.3-2.2 times/day	20.6-30.4 times/day
Abnormal data packet transmission (data volume/day)	Average detection of 100 MB/day	75.4 MB/day	502.6 MB/day
Average file size of downloads (MB)	15.3 MB	10.7 MB	50.8 MB
Anonymous payment amount (per transaction)	Average of $512.4	$ 52.3 - $ 98.5	$ 10.7 - $ 49.6

The experimental results show that the model can effectively identify illegal software transactions. First, in the detection of large-scale software downloads, there are an average of 8.7 to 12.3 large file downloads per day in illegal transaction traffic, while there are only 0.5 to 1.1 downloads in legal traffic. Secondly, frequent small anonymous payments have also become an important feature for identification. An average of 20.6 to 30.4 small payments are made per day in illegal transaction traffic, while legal transactions are only about 1.8. In addition, the transmission volume of abnormal data packets in illegal traffic far exceeds the normal range, with an average of 502.6MB of data transmitted per day, while the transmission volume of legal traffic is about 75.4MB.

Through the traffic identification of criminal behavior, technology not only provides analysis results, but more importantly, it helps law enforcement agencies take quick action. The proposed model provides categorized anomalous traffic patterns and their associated characteristics, such as traffic types and time intervals, which can help law enforcement identify potential threats and prioritize suspicious behavior for further investigation.

4 Discussion

The CNNH model showed excellent performance in the overlapping traffic segmentation task, with precision and recall reaching 91.43% and 93.46%, respectively, and a false positive rate of only 4.15%. In contrast, DC-CNN and MLP-DC each had an accuracy lower than 87% due to their limited feature extraction capabilities. The main reason for this performance difference was that CNNH achieved effective extraction of multi-scale features by introducing atrous convolution technology. Second, the accuracy trends of different models on the training and test sets. As the number of iterations increases, the normalized accuracy of CNNH reached 97% and 92% on the training and test sets, respectively. In addition, the statistical significance analysis further proved the reliability of these performance differences, with *P*-values less than 0.05 for all comparisons. The prediction accuracy of the MCNN-LSTM model reached 92.43%, the precision was 91.23%, the false positive rate was 5.12%, and it could achieve a 30-minute early warning capability. In comparison, the accuracy of the traditional LSTM model and the MLP-CNN model under complex traffic patterns was 84.67% and 80.45%, respectively.

Finally, compared to traditional methods, the proposed model showed significant advantages in scalability and dynamic adaptability. Traditional methods had acceptable performance on small data sets, but their accuracy was less than 80% in high-load traffic scenarios, making it difficult to effectively capture dynamic characteristics in complex network environments. In contrast, by combining deep learning technology, MCNN-LSTM not only performed stably in highly complex scenarios, but also provided early warning capabilities for criminal behavior, showing a wide range of practical application potential.

5 Conclusion

Through real-time monitoring of network traffic, the system can detect potential risks before criminal activities occur. In view of this, this study proposed a CNNH-based

Tor overlapping traffic segmentation model and an MCNN-LSTM website fingerprint recognition model. The performance test results indicated that the average segmentation accuracy of CNNH was 95.05% when the number of iterations was 50. Under Attack traffic, the P, R, F1, and AUC values of CNNH were 93.45%, 94.32%, 93.88%, and 0.935, respectively. The FPR was only 3.07%, which was better than the comparison model. Its computational time consumption was 19.12s, and the resource consumption rate was 71.54%. In the MCNN-LSTM performance test, its recognition accuracy for malware propagation and illegal transactions reached 96.54% and 92.87% respectively. In the prediction experiment results, the prediction accuracy of MCNN-LSTM was 92.43%, and it could issue an early warning 30 minutes in advance, with a false positive rate of only 5.12% and a detection time of only 24.3s. In terms of computational time consumption, the MCNN-LSTM model consumes 102ms per round of training. In the concept drift test, the recognition accuracy of the MCNN-LSTM model was 89.5% when the training and testing events were separated by 60 days. This shows that the proposed model in the study had excellent recognition accuracy and robustness.

6 Limitations and future research

The proposed MCNN-LSTM model performs well in anonymous network traffic analysis, but it still has some limitations. As the model complexity increases, CNNH and MCNN-LSTM have high computational resource and time consumption requirements, and may be difficult to deploy in real-time in hardware resource-constrained environments. The study simulated concept drift by adjusting feature distribution, protocol variations, and attack types, but drift in real-world scenarios may be more complex, such as sudden changes in user behavior or nonlinear changes in traffic patterns. In addition, advanced attackers may confuse traffic patterns by disguising malicious traffic or using complex encryption techniques to increase the difficulty of detection. For highly dynamic features or low-frequency anomalous behavior, the model may run the risk of failing to detect them. Although the false positive rate has been reduced, it may still cause false alarms that affect monitoring efficiency. In addition, the model may cause privacy issues when applied to anonymous network monitoring, such as excessive monitoring or false alarms that result in innocent users being tagged. The scope of monitoring must be strictly limited and privacy regulations must be followed.

Future research will focus on optimizing the performance and practical value of the model. First, through the lightweight design of the model and the distributed computing architecture, the computational and memory consumption can be reduced, and the scalability of large-scale real-time monitoring can be improved. Second, by combining long-term real Tor traffic data, the adaptability of the model in complex concept drift scenarios will be verified, and the robustness of the model against obfuscation strategies will be improved through adversarial training and multimodal data fusion. In addition, integrating data from multiple sources, such as user behavior logs and vulnerability information, can improve the ability to detect low-frequency anomalous behavior. In terms of privacy protection, future work will introduce data encryption and anonymization processing technologies, and combine the context post-processing mechanism to optimize false positive control to ensure the credibility and legality of the model application. Future research will also explore the applicability of the model in other potential application areas. For example, in enterprise network security, MCNN-LSTM can be used to detect abnormal traffic and potential attack behavior in the enterprise internal network, helping to improve security protection capabilities. At the same time, future research must focus on the ethical and privacy implications of model deployment, strictly adhere to relevant laws and regulations, and ensure the social responsibility and legality of the technology.

References

[1] F. Zhou, B. Zhou, S. Zhao, and G. Pan, “DeepOffense: a recurrent network-based approach for crime prediction,” CCF Transactions on Pervasive Computing and Interaction, vol. 4, no. 3, pp. 240-251, 2022. https://doi.org/10.1007/s42486-022-00100-x

[2] R. H. Shi, and X. Q. Fang, “Anonymous classical message transmission through various quantum networks,” IEEE Transactions on Network Science and Engineering, vol. 11, no. 3, pp. 2901-2913, 2024. https://doi.org/10.1109/TNSE.2024.3354327

[3] Y. J. Chen, Y. Su, M. Y. Zhang, H. Y. Chai, Y. K. Wei, and S. Yu, “Fedtor: an anonymous framework of federated learning in internet of things,” IEEE Internet of Things Journal, vol. 9, no. 19, pp. 18620-18631, 2022. https://doi.org/10.1109/JIOT.2022.3162826

[4] Y. Wang, “Deep learning models in computer data mining for intrusion detection,” Informatica, vol. 47, no. 4, 2023. https://doi.org/10.31449/inf.v47i4.4942

[5] X. D. Gu, B. C. Song, W. Lan, and M. Yang. “An online website fingerprinting defense based on the non-targeted adversarial patch,” Tsinghua Science and Technology, vol. 28, no. 6, pp. 1148-1159, 2023. https://doi.org/10.26599/TST.2023.9010062

[6] R. Rawat, and A. Rajavat, “Illicit events evaluation using NSGA-2 algorithms based on energy consumption,” Informatica, vol. 48, no. 18, 2024. https://doi.org/10.31449/inf.v48i18.6234

[7] K. Xian, “An optimized recognition algorithm for SSL VPN protocol encrypted traffic,” Informatica, vol. 45, no. 6, 2021. https://doi.org/10.31449/inf.v45i6.3730

[8] M. Nasr, A. Bahramali, and A. Houmansadr, "Defeating DNN-based traffic analysis systems in real-time with blind adversarial perturbations," In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), 2705-2722, 2021.

[9] K. Yesodha, M. Krishnamurthy, M. Selvi, and A. Kannan, “Intrusion detection system extended CNN and artificial bee colony optimization in wireless sensor networks,” Peer-to-peer Networking and Applications, vol. 17, no. 3, pp. 1237-1262, 2024. https://doi.org/10.1007/s12083-024-01650-w

[10] B. Q. Gan, Y. Q. Chen, Q. P. Dong, J. L. Guo, and R. X. Wang, “A convolutional neural network intrusion detection method based on data imbalance,” The Journal of Supercomputing, vol. 78, no. 18, pp. 19401-19434, 2022. https://doi.org/10.1007/s11227-022-04633-x

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

[12] F. R. Yan, G. H. Zhang, D. W. Zhang, X. H. Sun, B. T. Hou, and N. W. Yu, “TL-CNN-IDS: transfer learning-based intrusion detection system using convolutional neural network,” The Journal of Supercomputing, vol. 79, no. 15, pp. 17562-17584, 2023. https://doi.org/10.1007/s11227-023-05347-4

[13] G. Di Méo, “Historical co-offending networks: A social network analysis approach,” The British Journal of Criminology, vol. 63, no. 6, pp. 1591-611, 2023. https://doi.org/10.1093/bjc/azad005

[14] M. Merouane, “Convenient detection method for anonymous networks" I2P vs Tor",” Journal of Information Science and Engineering, vol. 39, no. 6, pp. 1371-1382, 2023. https://doi.org/10.6688/JISE.202311_39(6).0008

[15] M. Y. Jiang, B. J. Cui, J. S. Fu, T. Wang, and Z. Q. Wang, “KimeraPAD: a novel low-overhead real-time defense against website fingerprinting attacks based on deep reinforcement learning,” IEEE Transactions on Network and Service Management. vol. 21, no. 3, pp. 2944-2961, 2024. https://doi.org/10.1109/TNSM.2024.3360082

[16] M. Guo, Y. R. Sun, Y. L. Zhu, M. Q. Han, G. Dou, and S. P. Wen, “Pruning and quantization algorithm with applications in memristor-based convolutional neural network,” Cognitive Neurodynamics, vol. 18, no. 1, pp. 233-245, 2024. https://doi.org/10.1007/s11571-022-09927-7

[17] T. Li, Y. B. Yin, Z. Yi, Z. Guo, Z. L. Guo, and S. L. Chen, “Evaluation of a convolutional neural network to identify scaphoid fractures on radiographs,” Journal of Hand Surgery, vol. 48, no. 5, pp. 445-450, 2023. https://doi.org/10.1177/17531934221127092

[18] S. F. Lyu, and J. Q. Liu, “Convolutional recurrent neural networks for text classification,” Journal of Database Management, vol. 32, no. 4, pp. 65-82, 2021. https://doi.org/10.4018/JDM.2021100105

[19] A. Mahmoodzadeh, M. Mohammadi, S. G. Salim, H. F. H. Ali, H. H. Ibrahim, S. N. Abdulhamid, H. R. Nejati, and S. Rashidi, “” Machine learning techniques to predict rock strength parameters,” Rock Mechanics and Rock Engineering, vol. 55, no. 3, pp. 1721-1741, 2022. https://doi.org/10.1007/s00603-021-02747-x

[20] W. Chen, Y. Lu, H. Ma, Q. Chen, X. Wu, and P. Wu, “Self-attention mechanism in person re-identification models,” Multimedia Tools and Applications, vol. 81, no. 4, pp. 4649-4667, 2022. https://doi.org/10.1007/s11042-020-10494-4