Application of CNN-BiGRU-MHSA: A Self-Attention Mechanism Based Detection Method in Electronic Data Forensics

Yang Lei
Department of Investigation, Fujian Police College, Fuzhou 350007, China
E-mail: fzleyan@163.com

Keywords: SAM, electronic data, documents, fragmentation detection, CNN, MHSA

Received: November 7, 2024

In response to the shortcomings of the current file fragmentation detection technology in terms of accuracy and efficiency, this study proposes a file fragmentation detection method based on the multi-head self-attention mechanism. First, the traditional self-attention mechanism is optimized by introducing the concept of multi-dimensional features and multi-head feature detection. Second, a file fragmentation detection model is constructed by combining the optimized feature extraction method, bidirectional gated recurrent units, and convolutional neural networks. This model achieved classification accuracy as high as 99% and 98.9% on the GovDocs1 dataset and the Enron email dataset, respectively, with a mean square error as low as 0.08. In practical applications, the model achieved a high classification accuracy of 99.2%, a low classification time of 0.09 seconds, and a low false detection rate of 0.02%, demonstrating excellent detection performance. The detection algorithms and models designed in this study outperformed existing methods in both performance and practical application effectiveness. This integrated approach not only circumvents the constraints of conventional self-attention mechanisms in one-dimensional feature extraction but also augments the model's capacity to discern and extract multi-dimensional features in an efficacious manner. The results indicate that the model effectively improves the work efficiency in actual electronic data forensics, providing a new reference method for the field.

Povzetek: Predlagana je metoda za zaznavanje fragmentacije datotek, ki temelji na mehanizmu večglave samo-pozornosti (MHSA) za izboljšanje elektronske forenzike. Model združuje konvolucijske nevronske mreže (CNN), dvosmerne enote z zaprtimi ponavljajočimi se vrati (BiGRU) in optimiziran mehanizem samo-pozornosti.

1 Introduction

In today's rapid development of information technology, electronic data forensics has become an important means to combat cybercrime and maintain network security. As a fundamental element of electronic data forensics, the accuracy and efficiency of file fragmentation detection directly influence the efficacy of the resulting forensic analysis [1]. As data storage and transmission technologies have advanced, the phenomenon of file fragmentation has become more prevalent, increasing the difficulty of data reorganization and compromising the accuracy and integrity of forensic results [2-3]. Therefore, finding an accurate and fast method for identifying and reorganizing file fragments has become a pressing issue in the field of electronic data forensics. Traditional file fragmentation detection methods mainly rely on matching features at the head and tail of files, but it is often difficult to achieve the best results when dealing with highly random and complex file fragments [4]. In recent years, with the rapid development of artificial intelligence and deep learning (DL) technology, file fragment detection techniques based on machine learning and DL have gradually emerged. Among them, self-attention mechanism (SAM) is widely used due to its advantages in capturing long-range dependencies and global features [5]. However, most of the existing SAMs focus on unidimensional feature extraction, which limits the full utilization of multi-dimensional information. To overcome this limitation, this study proposes a file fragmentation detection method based on the multi-head self-attention (MHSA) mechanism, which significantly improves the accuracy and efficiency of file fragmentation detection by integrating multi-dimensional features and multi-head feature detection techniques.

The novelty of the research lies in the integration of convolutional neural network (CNN), bidirectional gated recurrent unit (BiGRU) and MHSA, which enables multi-dimensional feature extraction from file fragments. This integrated approach not only overcomes the limitations of traditional SAM in one-dimensional feature extraction, but also improves the recognition and extraction efficiency of the model for multi-dimensional features. This multi-modal feature extraction capability allows the model to more accurately identify and reorganize file fragments, improving the accuracy and integrity of electronic data forensics. The high classification accuracy and low false detection rate (FDR) of the model demonstrate that it can effectively improve the work efficiency in practical electronic data forensics, especially when dealing with large amounts of data.

2 Related works

SAM, as a DL technique, better captures the relationships and dependencies between features by dynamically assigning attention weights to different parts when processing large amounts of data. To forecast drug-target interactions, Cheng et al. developed a DL model based on graph attention networks and the MHSA mechanism. The model was able to extract drug and protein features and evaluate important amino acid sequences in proteins through the attention mechanism. Ultimately, the drug-target interactions were predicted through the full connectivity layer. The outcomes indicated that the precision, recall, and F1 value of the model outperformed the existing methods on multiple datasets [6]. Shen et al. proposed a new SAM-based long short-term convolutional neural network (LS-CNN) model, which was designed to be utilized for network traffic (NT) grayscale image detection and Android malware classification tasks. The self-attentive LS-CNN model was created to take into account both the spatial and temporal aspects of NT after all of the NT had been transformed into grayscale images in chronological sequence. Self-attention weights were also introduced to concentrate on various input aspects. The outcomes indicated that this approach was able to reliably and thoroughly classify malware by category in addition to successfully detecting malware [7]. Tao et al. suggested an SAM-based code recommendation method for automatically finding useful code snippets in a programming context thus assisting the program in its programming task. The method first built a small candidate set from the codebase, and then utilized self-attention networks to capture the deep semantics of the code in an abstract syntax tree, and finally recommended the relevant code to the developer. Experimental results indicated that the model outperformed existing methods in terms of recall, accuracy, and cumulative gain of normalized discounting [8].

File fragmentation detection is an important part of data recovery and digital forensics. Its goal is to identify and reorganize the scattered file fragments on the storage media. File fragmentation detection in essence belongs to the research content in the field of feature recognition, and a number of scholars have studied this field. Coquenet et al. proposed a new end-to-end segmentation free architecture known as Document Attention Network and used this network for handwritten document recognition. The outcomes indicated that the model achieved a character error rate of 3.43% and 3.70% on the page level and double page level of the Document Image Evaluation Database 2016 version dataset, respectively, with a low overall error rate [9]. Li et al. proposed an effective multi-hot coding and classification module for text recognition tasks in multilingual or large character set scenarios. In addition, the study also designed a lightweight converter to combine with this classification module. Finally, it developed a lightweight scene text recognition framework. Experimental results indicated that the built lightweight scene text recognition framework performed well in multiple environments [10]. A multi-domain character distance perception module was proposed by Zheng et al. with the goal of recognizing visual and semantic features in fused scene text recognition. The module combined character spacing, orientation change and semantic affinity through positional embedding and cross-attention mechanisms. Experimental results demonstrated that the module performed well on several publicly available datasets, especially in dealing with problems such as text distortion and chaotic layout of characters [11]. A forensic intelligence system based on signed documents was developed by Widiyasono et al. The system was used to automatically match file extensions and signatures, to recognize file types, and to address the problem of document authenticity maintenance in information technology. The outcomes indicated that the system could detect and recover files with modified extensions while improving the efficiency of forensic investigations [12].

In summary, a number of experts have built file fragmentation detection models using various DL techniques and achieved certain research results. In addition, a number of experts have also proposed various mechanisms to optimize file fragmentation detection techniques. Although existing research has provided a variety of effective file fragmentation detection techniques, further research and optimization are still needed in practical applications, especially when dealing with highly randomized and complex file fragments. This research aims to optimize SAM and combine it with neural networks to improve this shortcoming so as to meet the special detection needs in the field of electronic data forensics. A comparison of existing methods is shown in Table 1.

Table 1: Comparison of different methods

Research	Key methods	Data sets	Limitations
Cheng et al. [6]	Graph attention networks and MHSA	Drug-target interaction data	Lack of specific applications for file fragmentation detection
Shen et al. [7]	SAM-based LS-CNN	Network traffic data	Inability to fully utilise multi-dimensional information
Tao et al. [8]	SAM-based code recommendation	Programming context data	Limited ability to recognise complex file fragments
Coquenet et al. [9]	Document attention networks	Handwritten document data	Not applicable to file fragmentation detection in electronic data forensics
Li et al. [10]	Multi-hot encoding and classification module	Multilingual text data	Limited ability to reconstruct file fragments

Zheng et al. [11]	Multi-domain character distance awareness module	Scenario text data	Insufficiently comprehensive extraction of file fragmentation features
Widiyasono et al. [12]	Signature document-based forensic intelligence system	Signed document data	Limited ability to detect and recover file fragments
This study	CNN-BiGRU-MHSA	GovDocs1, Enron mail dataset	-

3 Methods and materials

For fully extracting the file fragmentation information in electronic data, the study firstly features the file fragments in electronic data forensics. Secondly, a new fragmentation detection model is built by combining MHSA mechanism with two neural network structures.

3.1 Approaches to document fragmentation in electronic data forensics

3.1.1 Conventional methods of file fragmentation processing and detection

A subset of technology known as "electronic data forensics" makes use of laws, regulations, and computer information technology to gather, examine, and preserve electronic data that may be used as proof in court. This technology is now widely used in cybercrime investigations, corporate data leakage incidents, intellectual property protection and other fields. In the process of electronic data forensics. It involves a variety of technical means, including data recovery, file analysis, NT monitoring, and log auditing and so on. Among them, file fragmentation detection used in the file analysis process has a key role [13-14]. During the process of data storage and transmission, files may be split into multiple discrete fragments and stored in different locations due to various reasons, such as system crashes, disk damage, file deletion, lack of storage space and network transmission interruptions. These fragmented files pose a great challenge to forensic work. Therefore, these scattered fragments must be reassembled to recover the content of the original file. The file fragmentation detection process in electronic data forensics is shown in Figure 1.

In Figure 1, firstly, the file data needs to be read. Secondly, determine whether the file data is complete or not. If the file data is not complete, further check whether it can be optimized for processing. If it cannot be optimized, the file recovery operation is carried out directly. If it can be optimized, then enter the file fragmentation type detection step to identify and classify file fragments. After the detection is complete, return to the process and continue with the file recovery step. During the file recovery process, attempts are made to rejoin the detected file fragments and recover them as complete file data. Finally, the recovered complete file data is input into other processes of digital forensics for further analysis and processing, thereby completing the entire forensic process. In the file fragmentation type detection task, the file header is a distinguishing mark used to identify different file types. Currently there are three common ways to view file types in computers, which are the viewing of the file's attributes, displaying the file format's suffix, and displaying the file's details. However, these viewing methods fail when the file type is tampered with, resulting in the loss of information such as file signatures or file headers. Therefore, understanding the format and storage structure of file types enables better identification of different types of files. Common file types and extensions in electronic data forensics are shown in Figure 2.

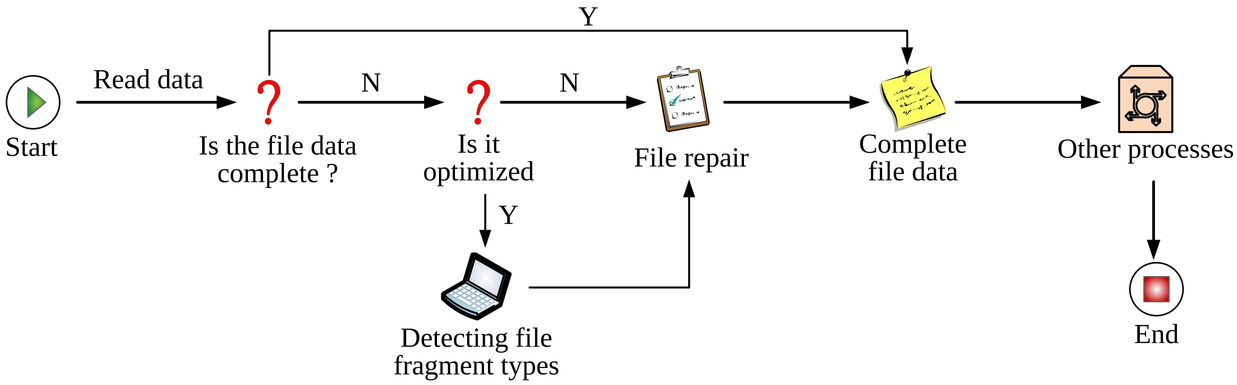


Figure 1: Flowchart of file fragmentation detection in electronic data forensics

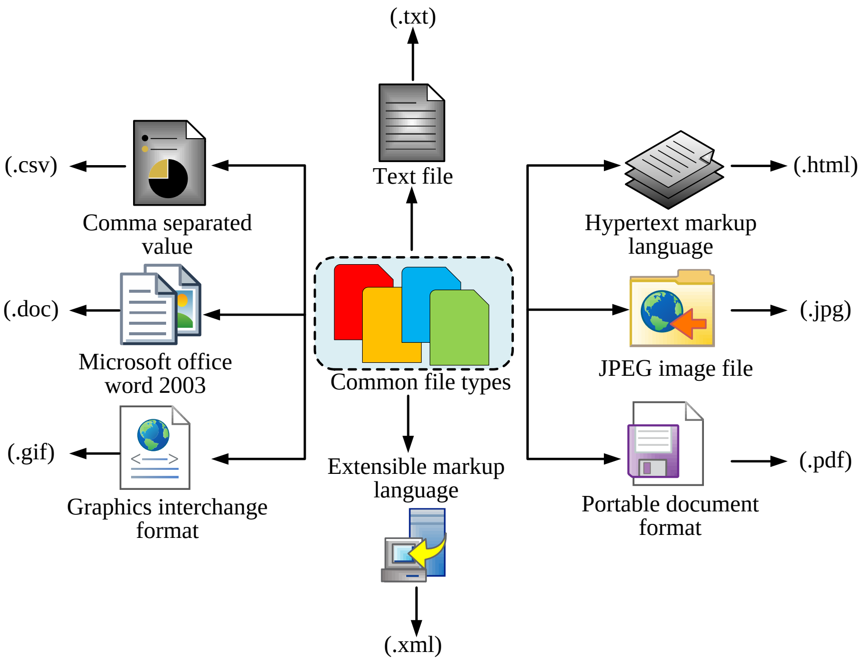


Figure 2: Common file types and extensions in electronic data forensics

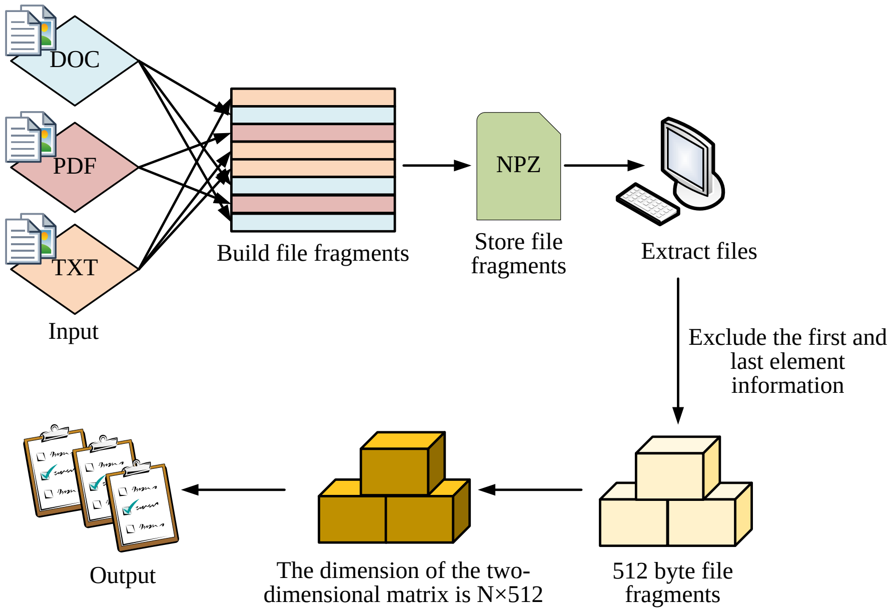


Figure 3: Construction and conversion process of file fragmentation

In Figure 2, a common file type in electronic data forensics is often exemplified by the comma-separated value file type, which can be notated with the .csv extension. Therefore, when performing file fragmentation detection, accurate fragmentation classification can be performed by recognizing these file types and file extensions. The similar files after classification are pieced together to ensure that the files can be recovered accurately [15-16]. In addition to recognizing file extensions to accomplish the fragmentation detection task, it is also possible to improve the detection accuracy by transforming the manifestation of fragmentation features. Currently, traditional fragmentation feature extraction methods rely on manual extraction, which is limited and inefficient. Considering that neural networks are excellent in automatically learning complex features, the study will use neural networks for fragmentation detection.

3.1.2 Document fragmentation data preprocessing

According to the file fragment detection method analysed in the previous section, the study first preprocessed the data. To fully extract the information of file fragments in electronic data, the file fragments are first characterized in electronic data forensics. In the data preprocessing stage, the steps shown in Figure 3 are taken to ensure the diversity and representativeness of the data, as well as the effectiveness of model training prior to detection.

In Figure 3, firstly, it is necessary to split multiple types of files into multiple small fragments to construct the experimental dataset to ensure the diversity and representativeness of the data. Second, these file fragments are preprocessed. The first and last meta-information of the files are removed, the core content is retained, and the consistency and purity of the data are ensured. The first and last meta-information typically contain non-content information such as the file's path and size, which are not important for identifying and restructuring the file's content and can introduce noise that degrades the model's performance. Instead, the core content of the file fragment is retained, allowing the model to focus on the actual content of the file for feature extraction and learning. Then, the preprocessed file fragments are fixedly partitioned according to 512-byte size to form small units suitable for neural network processing. Finally, these fixed-size file fragments are converted into a two-dimensional matrix. The dimension of the matrix is the number of fragments N×512 to facilitate subsequent feature extraction and model training. Through the above steps, the file fragments can be efficiently converted into a data format suitable for neural network processing, thereby facilitating feature extraction. The Z-score standardization method was also used to standardize the data. This method transforms the data by subtracting the mean and dividing by the standard deviation so that the data set has a mean of 0 and a standard deviation of 1. The specific standardization formula is shown in Equation (1).

$$\kappa = \frac{(x-\mu)}{\vartheta} \quad (1)$$

In Equation (1), x denotes raw data. μ denotes mean. ϑ denotes standard deviation. κ denotes standardised data. The dataset is divided into training and test sets in 8:2 ratio, where the training set mean attribute is $\mu_{train}=0.0$ and variance is $\vartheta_{train}^{2}=1.0$. The test set mean attribute is $\mu_{test}=0.01$ and variance is $\vartheta_{test}^{2}=1.05$. The dataset is taken from the GovDocs1 dataset and the Enron email dataset, which contain file and email data in a variety of formats. After preprocessing, the document fragment dataset is generated for training and testing. A total of 15695 normalized document fragment data were collected and divided into training and test sets in an 8:2 ratio. The GovDocs1 dataset contains document data in a variety of formats that are very common in electronic data forensics. The Enron email dataset contains a large amount of email data, which is also an important component of e-discovery. Both datasets are closely related to the e-discovery research focus. Furthermore, the GovDocs1 dataset and the Enron mail dataset comprise a plethora of heterogeneous document and email data, thereby enabling a comprehensive assessment of the model's efficacy in processing diverse and intricate forms of electronic data.

3.2 CNN-BiGRU-MHSA based file fragmentation detection model construction

After completing the construction and transformation of the file fragmentation dataset, the research will further combine MHSA mechanism, CNN, and BiGRU to build the file fragmentation detection model. BiGRU is a variant of recurrent neural networks specifically designed to process sequence data, especially time series or text data. It captures the information before and after the sequence through a bidirectional flow of information, which is crucial for understanding the context of file fragments. Transformer models typically require more computational resources. Especially, when using the self-attentive mechanism, its computational complexity increases with the length of the sequence. In contrast, file fragmentation detection involves capturing both local features (e.g., headers and tails) and global features (e.g., the overall structure of the file) of a file, and BiGRU is able to efficiently handle this type of data. Therefore, the study notated the final built detection model as CNN-BiGRU-MHSA, whose structure is shown in Figure 4.

In Figure 4, firstly, the preprocessed file fragments can be inputted into the model through the input layer, and the input file fragments have the size of N×512. Then, the embedding layer converts the input data into embedding vectors. Next, the CNN layer is introduced to extract local features and perform pooling. This is immediately followed by capturing the forward and backward temporal features of the file fragments using the BiGRU layer. Subsequently, the MHSA layer captures the hidden features through multiple attention heads as a way to enhance the feature extraction. The global average pooling layer (GAPL) pools the MHSA layer's output in order to minimize the feature dimensions. Meanwhile, to prevent overfitting, a dropout layer is added after the GAPL to randomly ignore some neuron outputs. Finally, the model contains two classification layers. Classification layer 1 outputs a dimension of N×64 and classification layer 2 outputs a dimension of N×20, which together constitute the final classification result. In the CNN-BiGRU-MHSA model, in an attempt to avoid the feature value gap being too large and thus affecting the extraction of features, it is necessary to perform a normalization operation on the input data [17]. The normalized formula is obtained by making the cleaned data feature r_{ij} as shown in Equation (2).

$$r_{ij}' = \frac{r_{ij} - r_{\min}}{r_{\max} - r_{\min}} \quad (2)$$

In Equation (2), r_{ij}' denotes the normalized $r_{\min}$, $r_{\min}$ and are the minimum and maximum values of the sample data. The set of normalized data is denoted as $s=\left[r_1, r_2, \cdots, r_n\right]$ and n is the individuals. According to Equation (2), the obtained s is used as an input to the CNN-BiGRU-MHSA model, which is first subjected to a convolution operation via CNN. The study chooses a 3×3 convolutional kernel for local feature extraction and used 64 filters to capture the features of the input data from different perspectives. Taking V_i in V_s as an example, the new features of V_i after convolution operation are obtained as shown in Equation (3) [18-20].

$$h_i = f\left(W_i * V_i + b\right) \quad (3)$$

In Equation (3), h_i denotes the new feature of V_i after convolutional processing. $f(\cdot)$ is the ReLU activation function (AF). b denotes the bias. W_i denotes the coefficients of the i th convolution kernel in V_i. A convolution kernel d is utilized to convolve all the features in V_s to obtain all the features at this point as shown in Equation (4).

$$H^d = \left[h_1^d, h_2^d, \cdots, h_i^d\right] \quad (4)$$

In Equation (4), H^d denotes the new features obtained after the convolution kernel d performs convolution operation on all the features in V_s. The H^d after convolution of all the convolution kernels of different sizes in the CNN is superimposed to obtain the output sequence (OS) of the final CNN, as shown in Equation (5).

$$H_s = \left[H^1, H^2, \cdots, H^d\right] \quad (5)$$

In Equation (5), H_s is the OS of the CNN. In BiGRU layer, the formula for reset gate is shown in Equation (6).

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right) \quad (6)$$

In Equation (6), t denotes the moment. r_t is the output value (OV) of the reset gate under t moment. W_r denotes the weight matrix of the reset gate. x_t denotes the input under the t moment. h_{t-1} denotes the hidden state (HS) at the $t-1$ moment. σ denotes the sigmoid AF. The formula for the candidate HS is shown in Equation (7).

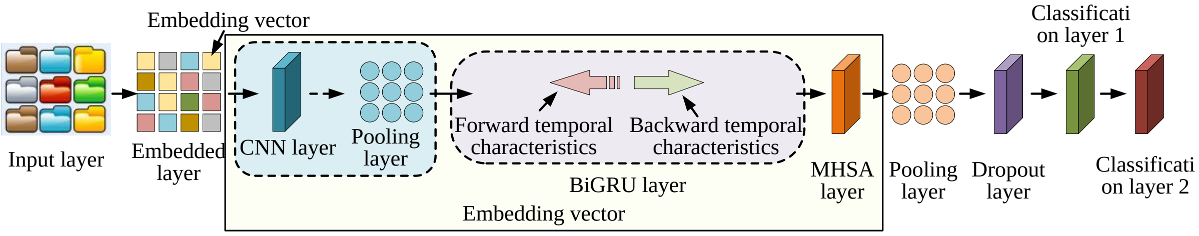


Figure 4: Structure diagram of CNN-BiGRU-MHSA model

$$h_t = \tanh\left(W_r \cdot \left[r_t * h_{t-1}, x_t\right]\right) \quad (7)$$

In Equation (7), h_t denotes the HS at the moment of t. The formula for updating the gate is shown in Equation (8).

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \quad (8)$$

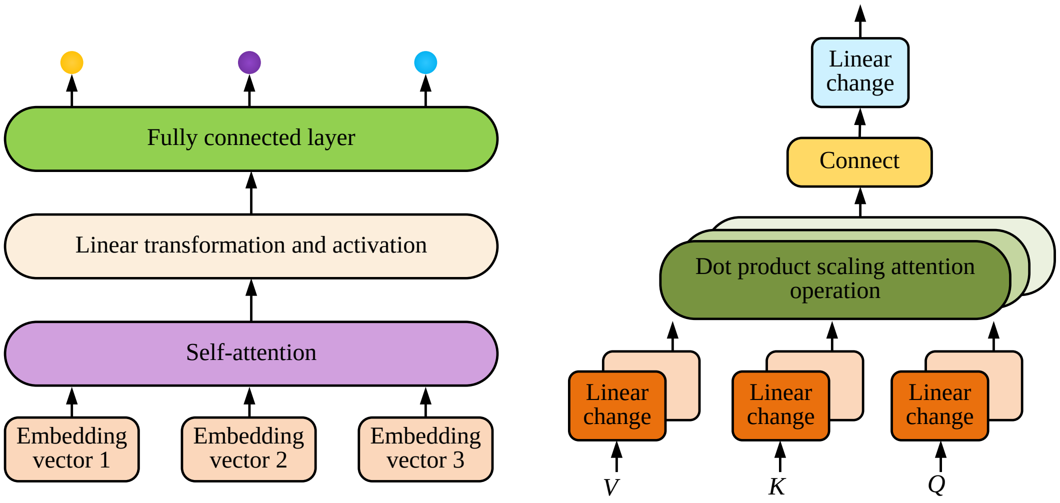
In Equation (8), z_t is the OV of the update gate (UG) at the t moment. W_z denotes the weight matrix of the UG. The formula for the HS under S moment is shown in Equation (9).

$$h_t = \left(1 - z_t\right) * h_{t-1} + z_t * h_t \quad (9)$$

In Equation (9), h_t is the HS at the t moment. In the MHSA layer, the adopted MHSM consists of a combination of SAM and multi-head attention mechanism (MHAM). This structure can enhance the model's ability to understand the input data by processing the data in parallel with multiple attention heads and learning features in different subspaces. The structure of SAM and MHAM is shown in Figure 5.

In Figure 5(a), SAM converts the input sequences into embedding vectors and then generates the weight matrix of query Q, key K, and value V. Next, the dot product of Q and K is computed and normalized by applying the Softmax function, and the resulting weights are used to weight and sum the V matrix and generate the self-attentive output. As shown in Figure 5(b), MHAM first linearly transforms the input sequence into multiple Q, K, and V matrices, respectively. Each attention head computes the attention output independently. The final multi-head attention output is then created by splicing and linearly transforming each head's outputs. The formula for generating multiple Q, K, and V matrices is shown in Equation (10).

$$\begin{cases} Q_m = W_{qm} X \\ K_m = W_{km} X \quad m = 1, 2, \cdots n \\ V_m = W_{vm} X \end{cases} \quad (10)$$



(a) Self-attention mechanism (b) Multi-head attention mechanism

Figure 5: SAM and MHAM structure diagram

In Equation (10), X denotes the embedding vector of the input sequence. Q_m , K_m , and V_m denote the query, key and value matrices of the m th header, respectively. W_{qm} , W_{km} . and W_{vm} denote the weight matrices of Q_m , K_m . and V_m , respectively. The attention OV of each head is shown in Equation (11).

$$head_m = Attention\left(Q_m, K_m, V_m\right) = soft\max\left(\frac{Q_m K_m^T}{\sqrt{d_k}}\right)V_m \qquad (11)$$

In Equation (11), $head_m$ denotes the attention OV of the m th head. K_m^T denotes the transpose matrix of K_m . d_k denotes the dimension of the key vector, which is used to scale the dot product result. At the MHSA layer, the output process of splicing all the heads is shown in Equation (12).

$$MultiHead\left(Q,K,V\right) = Concat\left(head_1, head_2, \cdots head_n\right)W_O \qquad (12)$$

In Equation (12), *Concat* denotes the splicing operation. W_O denotes the final output weight matrix. The operational flow of GAPL is shown in Figure 6.

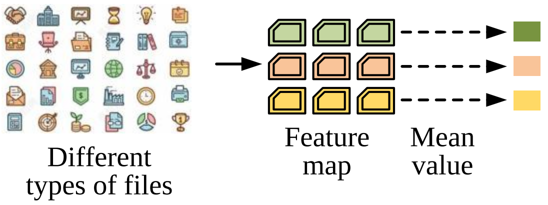


Figure 6: Schematic diagram of GAPL

In Figure 6, GAPL generates a smaller feature representation by averaging the pooling of all elements in each channel. This preserves important spatial information and reduces feature dimensions, thereby reducing computational complexity. The GAPL is characterized by the absence of learning parameters, which renders it an effective means of reducing the complexity of the model and the risk of overfitting. This is achieved while retaining the global information of the feature map, thereby enabling the model to capture a more expansive range of features. The flow of the Dropout operation is shown in Figure 7.

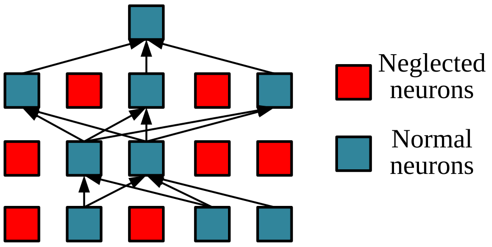


Figure 7: Dropout layer schematic diagram

In Figure 7, the dropout layer randomly ignores the outputs of a subset of neurons with a certain probability during the training process. That is, by setting the outputs of these neurons to zero, the model is better able to generalize and learn using different sub-networks for each training iteration. In the process of building the CNN-BiGRU-MHSA model, the study carefully parameterized the BiGRU and MHSA layers to optimize the feature extraction and classification performance of the model. The number of hidden layer units in the BiGRU layer is 256. The number of attention header bits in the MHSA layer is 8, and the dropout rate is set to 0.2.

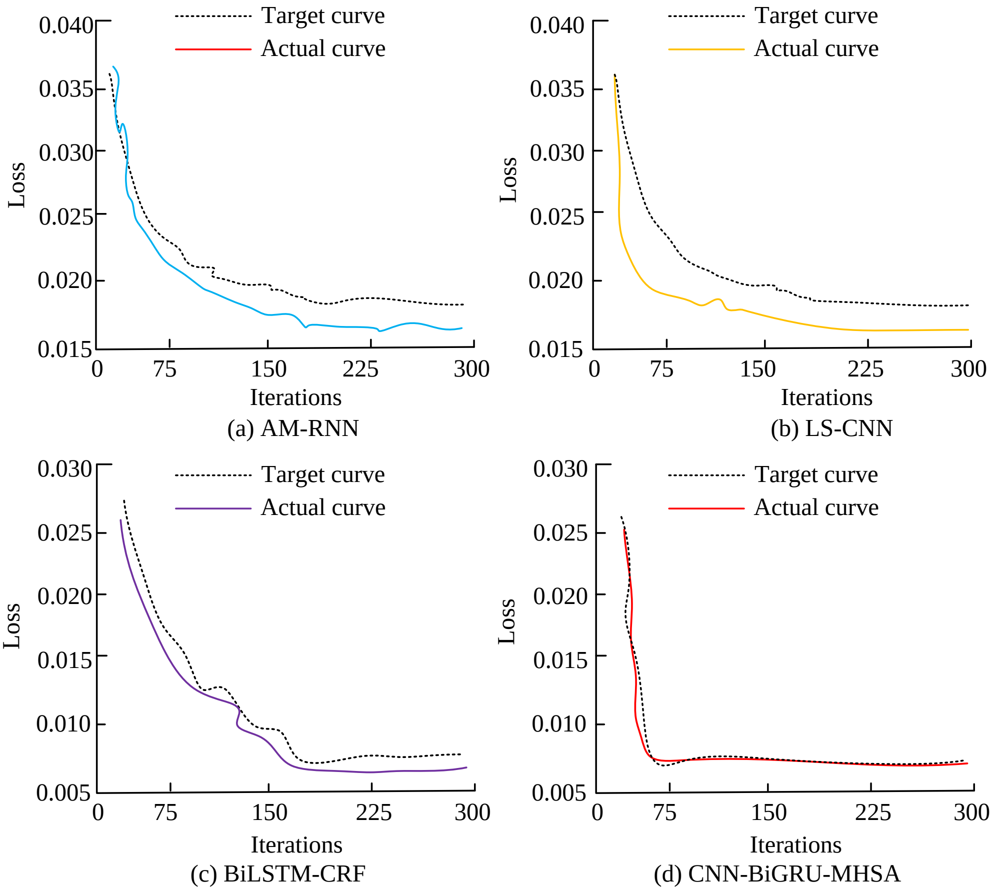
4 Results

To demonstrate that the constructed CNN-BiGRU-MHSA model has better performance in the file fragmentation detection task, the study tests the baseline performance (BP) of the model and the effect of practical application respectively. LS-CNN, bidirectional long short-term memory with conditional random fields (BiLSTM-CRF), and attention mechanism-recurrent neural network (AM-RNN) are selected as three comparison models. RNN as three comparison models. The performance of the four models is tested under the detection metrics such as loss function, accuracy, recall, and F1 value.

4.1 Baseline performance analysis of CNN-BiGRU-MHSA models

The experimental environment configuration used for the study includes an Intel Core i7-10700K processor with 32GB of RAM, an NVIDIA RTX 3080 graphics card, an Ubuntu 20.04 operating system, and the TensorFlow 2.5.0 DL framework. Figure 8 illustrates the comparison of the objective loss function and actual loss function trends for each of the four models.

The target loss function curves and actual loss function curves of AM-RNN, LS-CNN, BiLSTM-CRF, and CNN-BiGRU-MHSA are given in Figure 8, respectively. In Figure 8, the actual loss function curve of CNN-BiGRU-MHSA matches well with the target loss function curve throughout the iteration process. However, there are different degrees of ups and downs between the actual loss function curves and the target loss function curves of all three models, AM-RNN, LS-CNN, and BiLSTM-CRF, which do not match well. This illustrates that CNN-BiGRU-MHSA has better stability during the iterative process. The classification precision rate of the four models in different datasets is tested, as shown in Figure 9.



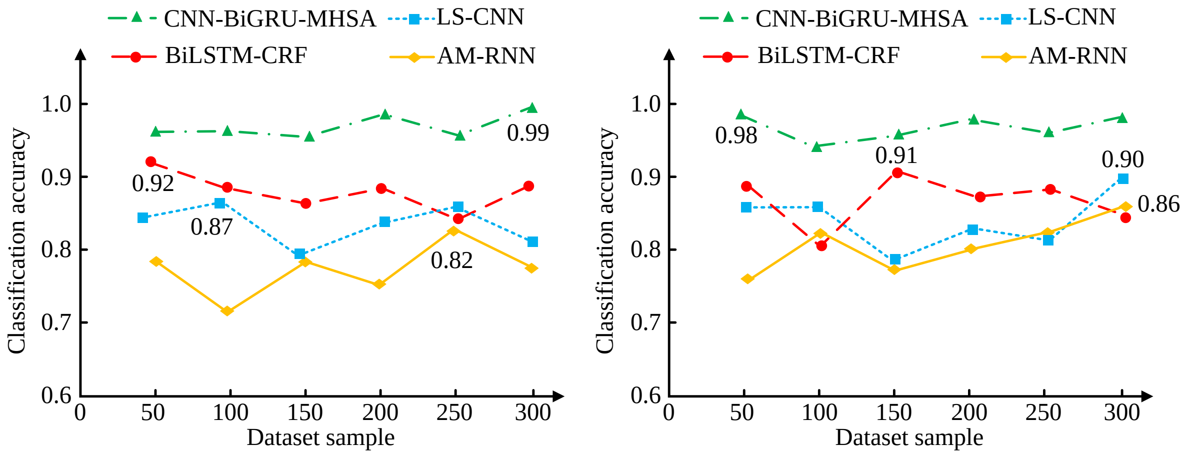
(a) AM-RNN

(b) LS-CNN

(c) BiLSTM-CRF

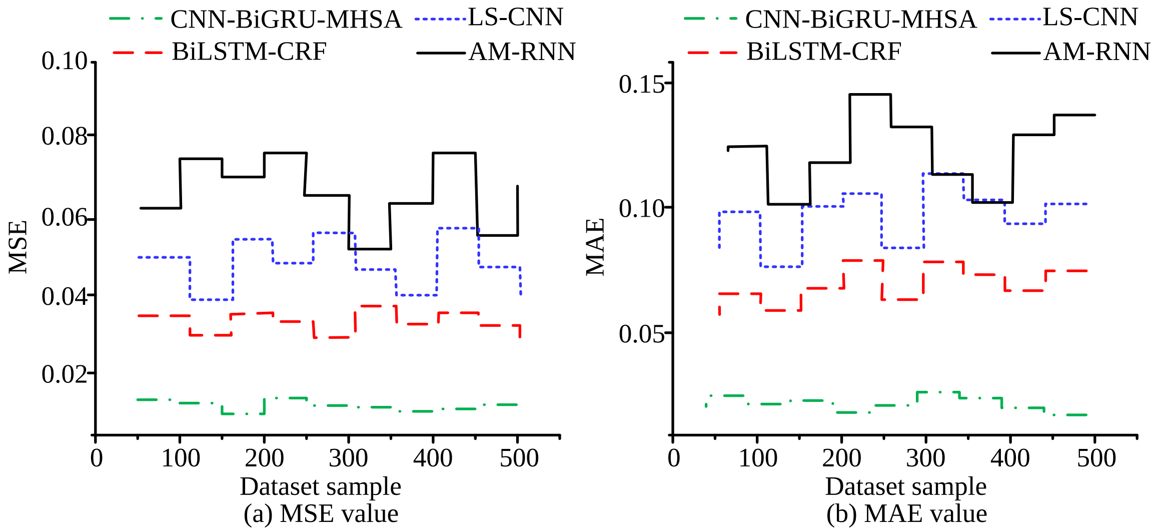
(d) CNN-BiGRU-MHSA

Figure 8: Loss curves for different models



(a) GovDocs1 dataset (b) Enron email dataset

Figure 9: Classification precision rate of different models in two data sets



(a) MSE value (b) MAE value

Figure 10: MSE and MAE for different models

Table 2: Statistical significance test results of different models

Data set	ANOVA		t-test		
	F-value	P-value	Model comparison	t-value	P-value
GovDocs1	5.89	0.00	CNN-BiGRU-MHSA vs AM-RNN	3.52	0.00
			CNN-BiGRU-MHSA vs LS-CNN	4.11	0.00
			CNN-BiGRU-MHSA vs BiLSTM-CRF	2.87	0.01
Enron email	6.23	0.00	CNN-BiGRU-MHSA vs AM-RNN	3.98	0.00
			CNN-BiGRU-MHSA vs LS-CNN	3.65	0.00
			CNN-BiGRU-MHSA vs BiLSTM-CRF	3.21	0.00

Table 3: Results of ablation experiment

CNN	BiGRU	MHSA	Average accuracy (%)
√	×	×	76.57
×	√	×	78.24
×	×	√	71.23
√	√	×	92.11
	√	√	91.59
√	×	√	93.77
√	√	√	98.97

Figure 9(a) and Figure 9(b) show the classification precision rate values of the four models in the GovDocs1 and Enron email dataset, respectively. In Figure 9(a), the maximum classification precision rate of AM-RNN, LS-CNN, BiLSTM-CRF, and CNN-BiGRU-MHSA in GovDocs1 dataset are 0.82, 0.87, 0.92, and 0.99, respectively, when the samples rising. Similarly, it can be obtained that in Figure 9(b), AM-RNN, LS-CNN, BiLSTM-CRF and CNN-BiGRU-MHSA have maximum classification precision rate of 0.86, 0.90, 0.91, and 0.98 in the Enron email dataset dataset, respectively. In summary, the classification accuracies of CNN-BiGRU-MHSA in both datasets exceed 0.95, demonstrating the model's generally strong baseline classification performance. The mean square error (MSE) and mean absolute error (MAE) performances of the four models are tested, as shown in Figure 10.

Figures 10(a) and 10(b) show the MSE and MAE of AM-RNN, LS-CNN, BiLSTM-CRF, and CNN-BiGRU-MHSA, respectively, in the test set. The CNN-BiGRU-MHSA performs best in the error test, with the lowest MSE and MAE values of 0.08 and 0.12, respectively, which are much lower than that of the AM-RNN model in the MSE value of 0.53 and MAE value of 0.10. To verify the statistical significance of the performance differences between the models, t-test and analysis of variance (ANOVA) are further conducted. The specific results are shown in Table 2.

In Table 2, the classification performance of CNN-BiGRU-MHSA model on GovDocs1 and Enron mail data sets is statistically significant compared with other models, and the P values are all less than 0.05. This shows that the performance improvement of the CNN-BiGRU-MHSA model is remarkable. At the same time, it further verifies the superiority of CNN-BiGRU-MHSA model in classification performance. Finally, the ablation experiment is continued and the specific results are shown in Table 3.

Table 3 shows the average classification accuracy in two datasets under different combinations. It can be seen that the classification accuracy of CNN+BiGRU+MHSA is significantly better than other combinations. Among them, the classification accuracy of single MHSA is the lowest, while the accuracy of single BiGRU is significantly higher than that of MHSA and CNN. This may be because BiGRU has some ability in sequence modeling, but its performance is not as good as that of CNN when used alone. Overall, each component in the CNN-BiGRU-MHSA model contributes to improving the classification performance, especially when all three components are included, the performance of the model is the best. This shows that the CNN-BiGRU-MHSA model is reasonable and superior.

4.2 Application effect of CNN-BiGRU-MHSA modeling

To corroborate the CNN-BiGRU-MHSA model's superior classification performance in actual file fragmentation detection scenarios, the research employs a crawler tool to gather over 1,000 distinct categories of electronic data from an information website. It is also categorized into eight common file fragmentation types according to the classification rules in Figure 2. The crawler tool used for research is an automatic data collection program that can crawl the required data from the specified website according to the preset rules. The tool has the ability to automatically identify the site structure, extract files of specific formats and support multithreading operations. The selected information website is an online resource platform that provides various types of electronic documents and data files. The website is known for its rich data types and high data update frequency, which is suitable for the data collection needs of research. The rules are based on common file type and extension standards in electronic data forensics. Classification criteria include file extension, file header information, file content characteristics, etc. In the process of implementation, the file extension is initially utilized for preliminary classification, and subsequently, the file header information and content characteristics are integrated for detailed classification. This approach ensures the accuracy and integrity of the classification process. The classification effect of the four models for the eight file fragment types is tested, as shown in Figure 11.

Figure 11 give the classification accuracy and classification time of CNN-BiGRU-MHSA, BiLSTM-CRF, LS-CNN, and AM-RNN for detecting eight file fragment types, respectively. Among them, file types 1-file types 8 are (.txt), (.csv), (.doc), (.gif), (.xml), (.pdf), (.jpg), and (.html), respectively. In Figure 11, the CNN-BiGRU-MHSA model has the best classification accuracy and the shortest classification time. The highest classification accuracies of CNN-BiGRU-MHSA, BiLSTM-CRF, LS-CNN, and AM-RNN are 99.2%, 93.4%, 88.1%, and 86.9%, respectively. The shortest classification times are 0.09s, 0.38s, 0.56s, and 3.62s, respectively. The four models are further tested for the FDR for different document types, as shown in Table 4.

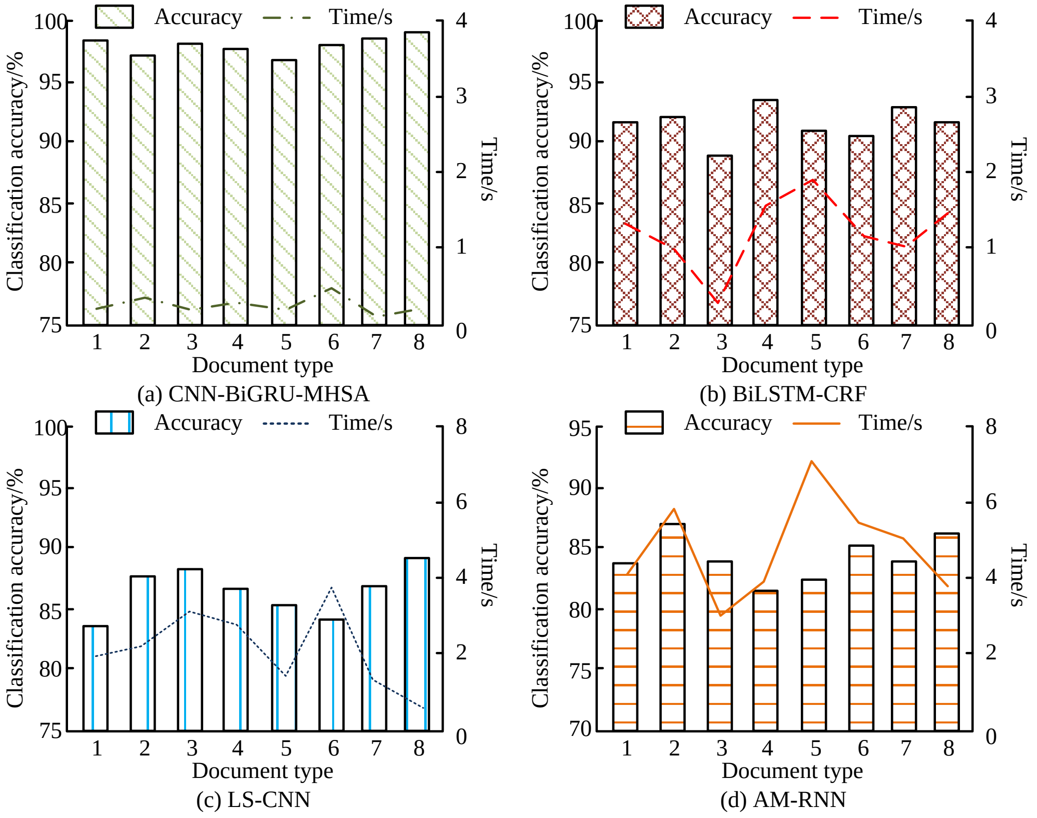
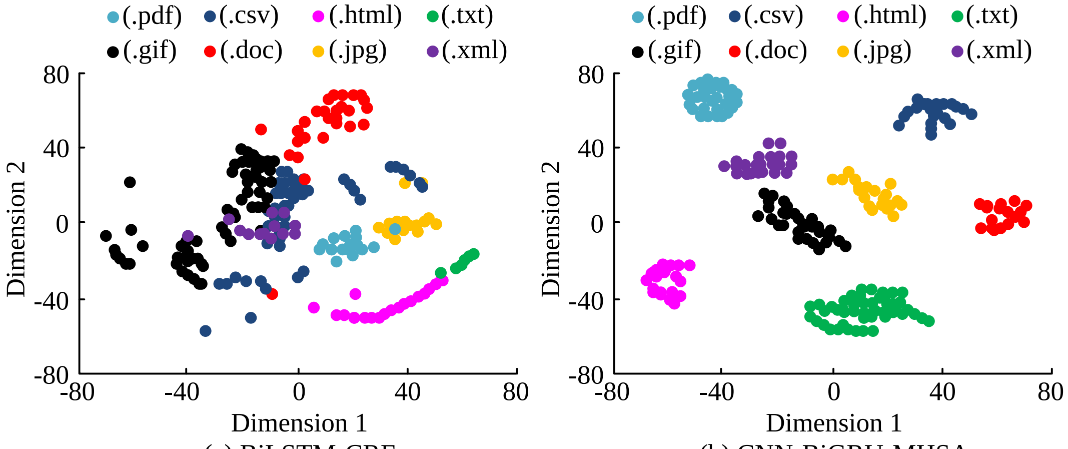


Figure 11: CA and classification time of different models

Table 4: FDR of four models in the process of file classification detection

File type	AM-RNN	LS-CNN	BiLSTM-CRF	CNN-BiGRU-MHSA
1	0.48%	0.28%	0.22%	0.06%
2	0.51%	0.32%	0.17%	0.11%
3	0.42%	0.20%	0.20%	0.08%
4	0.36%	0.25%	0.12%	0.02%
5	0.44%	0.31%	0.15%	0.10%
6	0.46%	0.24%	0.11%	0.05%
7	0.50%	0.29%	0.19%	0.13%
8	0.39%	0.33%	0.23%	0.11%



(a) BiLSTM-CRF

(b) CNN-BiGRU-MHSA

Figure 12: Classification results of file fragmentation of different models in the two models

The FDR is the proportion of all samples that are actually negative categories (i.e., not the category of interest to the study) that the model incorrectly predicts to be positive categories (i.e., the category of interest to the study.) The lower the FDR, the less the tendency of the model to misclassify negative categories as positive categories. In Table 4, the lowest FDRs for the eight file types detected using CNN-BiGRU-MHSA, BiLSTM-CRF, LS-CNN, and AM-RNN are 0.02%, 0.11%, 0.24%, and 0.36%, respectively. The CNN-BiGRU-MHSA model's total FDR is significantly lower than the other three models when compared, staying around 0.15%. The classification effects of the four models are represented by clustering in the dimension space, as shown in Figure 11.

Figure 12(a) and 12(b) show the classification results of BiLSTM-CRF and CNN-BiGRU-MHSA for file fragments in dimensional space, respectively. Eight different types of file fragments are better classified in the CNN-BiGRU-MHSA model than the BiLSTM-CRF model. In Figure 12(b), the classification results of CNN-BiGRU-MHSA are neater and there are no misclassifications and omissions. However, in Figure 12(a), the classification of BiLSTM-CRF is more chaotic and misclassification of document fragments occurs. Overall, the proposed model is easier to identify text files (such as .txt and .csv) because such files usually have a consistent structure and format. For image files (such as .gif and .jpg), the model must rely on the binary pattern and visual characteristics of the files to distinguish them.

5 Discussion

The CNN-BiGRU-MHSA model proposed in the study performed exceptionally well on the task of detecting file fragmentation in electronic data forensics, significantly outperforming models such as AM-RNN, LS-CNN, and BiLSTM-CRF. The model achieved classification accuracy of 0.99 and 0.989 on the GovDocs1 and Enron email datasets, respectively. The model proposed in reference [6], based on graph attention networks and MHSA, demonstrated higher precision, recall, and F1 values on the drug-target interaction dataset. This indicated that the introduction of MHSA for fragmentation detection classification had a certain rationality. The LS-CNN model based on SAM proposed in reference [7] showed good classification reliability on NT data, but had limitations in utilizing multi-dimensional information. In contrast, the CNN-BiGRU-MHSA model proposed in this study combined CNN, BiGRU, and MHSA mechanisms to optimize feature extraction and sequence modeling. This combination not only overcome the limitations of traditional SAM in one-dimensional feature extraction, but also improved the model's detection and extraction efficiency for multi-dimensional features. The use of the MHSA mechanism was a significant innovation of the study. It processed data in parallel with multiple attention heads, improving the model's understanding of the input data, which was a novel approach in the field of electronic data forensics. Accurate detection of file fragmentation was critical for data recovery and evidence integrity. The high accuracy and low error rates of the model proposed in the study indicated that it could effectively improve the accuracy and efficiency of electronic data forensics.

6 Conclusion

A new method for detecting file fragments in electronic data forensics, the CNN-BiGRU-MHSA model, was proposed by this study. This model combined the advantages of CNN, BiGRU, and MHSA, and achieved excellent performance in classification accuracy and efficiency. The integration of these components allowed multi-dimensional features to be extracted more comprehensively, which overcome the limitations of traditional SAM. In practical application, the classification accuracy of this model reached 99.2%, and the FDR was only 0.02%, highlighting its potential to improve the effectiveness of electronic data forensics. The research results provided an accurate and efficient file fragment detection method for this field. This not only fills the gap in the existing technology, but also points out the direction for future research and practical application. Nevertheless, the proposed model still has room for improvement, especially when dealing with extremely complex and random file fragments. Future work will concentrate on further optimizing the model structure with the objective of enhancing its adaptability and robustness in the context of challenging situations.

References

[1] Vidyapati Kumar, Kanak Kalita, Prasenjit Chatterjee, Edmundas Kazimieras Zavadskas, and Shankar Chakraborty. A SWARA-CoCoSo-based approach for spray painting robot selection. Informatica, 33(1): 35-54, 2022. https://doi.org/10.15388/21-INFOR466

[2] Jing Chen, Deying Chen, Hao Jiang, Xiren Miao, and Cunyi Yin. Skeleton-based 3D human pose estimation with low-resolution infrared array sensor using attention-based CNN-BiGRU. International Journal of Machine Learning and Cybernetics, 15(5):2049-2062, 2024. https://doi.org/10.1007/s13042-023-02015-0

[3] Mehdi Gheisari, Hooman Hamidpour, Yang Liu, Peyman Saedi, Arif Raza, Ahmad Jalili, Hamidreza Rokhsati, and Rashid Amin. Data mining techniques for web mining: a survey. Artificial Intelligence and Applications, 1(1):3-10, 2023. https://doi.org/10.47852/bonviewAIA2202290

[4] Hanane Elfaik, and El Habib Nfaoui. Leveraging feature-level fusion representations and attentional bidirectional RNN-CNN deep models for arabic affect analysis on twitter. Journal of King Saud University-Computer and Information Sciences, 35(1):462-482, 2023. https://doi.org/10.1016/j.jksuci.2022.12.015

[5] Roop Ranjan, and A.K. Daniel. Cobico: a model using multi-stage convnet with attention-based bi-LSTM for efficient sentiment classification. International Journal of Knowledge-based and Intelligent Engineering Systems, 27(1):1-24, 2023. https://doi.org/10.3233/KES-230901

[6] Zhongjian Cheng, Cheng Yan, Fangxiang Wu, and Jianxin Wang. Drug-target interaction prediction

using multi-head self-attention and graph attention network. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19(4):2208-2218, 2021. https://doi.org/10.1109/TCBB.2021.3077905

[7] Limin Shen, Jiayin Feng, Zhen Chen, Zhongkui Sun, Dongkui Liang, Hui Li, and Yuying Wang. Self-attention based convolutional-LSTM for android malware detection using network traffics grayscale image. Applied Intelligence, 53(1):683-705, 2023. https://doi.org/10.1007/s10489-022-03523-2

[8] Chuanqi Tao, Kai Lin, Zhiqiu Huang, and Xiaobing Sun. Cram: code recommendation with programming context based on self-attention mechanism. IEEE Transactions on Reliability, 72(1):302-316, 2022. https://doi.org/10.1109/TR.2022.3171309

[9] Denis Coquenet, Clément Chatelain, and Thierry Paquet. Dan: a segmentation-free document attention network for handwritten document recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(7):8227-8243, 2023. https://doi.org/10.48550/arXiv.2203.12273

[10] Bingcong Li, Xin Tang, Xianbiao Qi, Yihao Chen, Chunguang Li, and Rong Xiao. EMU: Effective multi-hot encoding net for lightweight scene text recognition with a large character set. IEEE Transactions on Circuits and Systems for Video Technology, 32(8):5374-5385, 2022. https://doi.org/10.1109/TCSVT.2022.3146240

[11] Tianlun Zheng, Zhineng Chen, Shancheng Fang, Hongtao Xie, and Yugang Jiang. Cdistnet: perceiving multi-domain character distance for robust text recognition. International Journal of Computer Vision, 132(2):300-318, 2024. https://doi.org/10.48550/arXiv.2111.11011

[12] Nur Widiyasono, Randi Rizal, Siti Yuliyanti, Siti Rahayu Selamat, and Mugi Praseptiawan. A forensic intelligence system for identification of data originality based on signature files. Journal of Advanced Research in Applied Sciences and Engineering Technology, 48(1):193-204, 2024. https://doi.org/10.37934/araset.48.1.193204

[13] Siqi Tang, Guiwu Wei, and Xudong Chen. Location selection of express distribution centre with probabilistic linguistic MABAC method based on the cumulative prospect theory. Informatica 33(1): 131-150, 2022. https://doi.org/10.15388/21-INFOR467

[14] Azra Parveen, Zishan Husain Khan, and Syednaseem Ahmad. Classification and evaluation of digital forensic tools. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(6):3096-3106, 2020. https://doi.org/10.12928/telkomnika.v18i6.15295

[15] Abdullah Ayub Khan, Aftab Ahmed Shaikh, Asif Ali Laghari, Mazhar Ali Dootio, M. Malook Rind, and Shafique Ahmed Awan. Digital forensics and cyber forensics investigation: security challenges, limitations, open issues, and future direction. International Journal of Electronic Security and Digital Forensics, 14(2):124-150, 2022. https://doi.org/10.1504/IJESDF.2022.121174

[16] Abdullah Ayub Khan, and Syed Asif Ali. Network forensics investigation: Behaviour analysis of distinct operating systems to detect and identify the host in IPv6 network. International Journal of Electronic Security and Digital Forensics, 13(6):600-611, 2021. https://doi.org/10.1504/ijesdf.2021.118542

[17] Varshapriya Jyotinagar, and Bandu B. Meshram Digital forensic analysis of attack detection and identification in private cloud environments for databases. Journal of Integrated Science and Technology, 12(4):798-798, 2024. https://doi.org/10.62110/sciencein.jist.2024.v12.798

[18] Himanshu Himanshu, Shobha Bhatt, and Lokesh Negi. Digital forensics techniques and trends: a review. The International Arab Journal of Information Technology, 20(4):644-654, 2023. https://doi.org/10.34028/iajit/20/4/11

[19] Lejun Zhang, Yuan Li, Ran Guo, Guopeng Wang, Jing Qiu, Shen Su, Yuan Liu, Guangxia Xu, Huiling Chen, and Zhihong Tian. A novel smart contract reentrancy vulnerability detection model based on BiGAS. Journal of Signal Processing Systems, 96(3):215-237, 2024. https://doi.org/10.1007/s11265-023-01859-7

[20] Yan Liang, and Feng Pan. Study of automatic piano transcription algorithms based on the polyphonic properties of piano audio. IEIE Transactions on Smart Processing & Computing, 12(5):412-418, 2023. https://doi.org/10.5573/IEIESPC.2023.12.5.412