International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-14

https://doi.org/10.70917/ijcisim-2026-1105
Article

Risk Traceability of False Content Generation Enabling by Machine Learning Technology in the Intelligent Media Era

Peilin Qi [1,*] and Xiuyu Zhang [2]

[1] Beijing National Day School, Beijing, 100011, China
[2] Capital Normal University High School, Beijing, 100011, China
* Correspondence author: Patrickqi2026@163.com

Abstract: This paper employs a Text-CNN network to extract textual and image features from multimodal fake media content, maps different media features into a joint space, and achieves comprehensive feature extraction of multimodal media content through concatenation and embedding. We propose MTTV, a media false content detection method based on a multimodal Transformer and designed for multidimensional feature information. It embeds the extracted multimodal information into the detection network and uses a multi-layer Transformer encoder to classify content as true or false. Results indicate that the most important multimodal features, in descending order, are follower count, like count, user reputation score, registration duration, and post count, with importance values of 0.312, 0.305, 0.300, 0.271, and 0.250, respectively. The model proposed in this paper achieves an accuracy rate of 0.922 for media fake content detection, the highest among all compared models. Based on the features identified for fake content detection, mitigating the risk of media fake content generation requires the joint participation of multiple stakeholders, including media professionals, commercial platforms, and the public.

Keywords: media content; fake content detection; multimodal Transformer; MTTV; Text-CNN network

1. Introduction

With the development and widespread adoption of artificial intelligence (AI) and the advent of the smart media era, intelligence has become a new form of information dissemination, with high-density, real-time, and multimedia-integrated information flooding the internet [1–3]. Unlike traditional media, in the smart media era, users can access a vast amount of news and information from the comfort of their homes with just a click of the mouse. However, the proliferation of smart media and the application of AI technologies—particularly machine learning—have brought about a serious challenge in the form of fake content generation, causing adverse effects on society [4-5].

Machine learning is one of the key technologies driving AI development. In the era of smart media, malicious actors exploit the characteristics of generative models within machine learning to fabricate false information [6-7]. Through training on large datasets, generative models can learn patterns and features of text, images, and other content, and use these models to generate content that appears authentic but is actually false [8-10]. For example, some malware uses machine learning to generate realistic phishing website pages, tricking users into believing they are legitimate sites and prompting them to enter sensitive information. These generated pages are extremely similar to real websites in terms of appearance and layout, making them difficult to distinguish with the naked eye [11]. In the realm of text generation, language models are used to create fake news reports, academic papers, and other content. Such fabricated texts may span various fields—including politics, economics, and society—spreading false narratives and events that disrupt the normal flow of information [12–14]. Therefore, exploring the use of machine learning technologies to enable risk tracing for the generation of false content is of great significance. The objective is to establish a traceable, verifiable, and

auditable evidence-based record across the entire chain of information generation, dissemination, and consumption, ensuring that key elements such as the source of information, the generation process, the dissemination path, and modification records form a credible "chain of evidence" [15–18].

To achieve risk tracing for the generation of false media content, this paper establishes a false content detection mechanism. This mechanism combines text and image features into a multimodal feature extractor, using a Text-CNN network to extract text features and a VGG19 network to extract image features, and employs data fusion methods to leverage the advantages of different modalities to fuse the various features. Based on these two-level visual features (text and image), we propose an MTTV detection method based on a multimodal Transformer, comprising two sub-processes: multimodal media content embedding, and encoding and classification. The multimodal embedding is achieved by enhancing semantic information through multimodal processing, while encoding and classification are performed via a multi-layer Transformer encoder sequence.

2. A Model for Detecting Fake Content in Multimodal Media

2.1. Multimodal Feature Extraction

The multimodal feature extractor consists of a text feature extractor and an image feature extractor: The text feature extractor comprises a Text-CNN network and a fully connected layer. Pre-trained text word vectors are input into the Text-CNN network to extract text vector representations, and a fully connected layer is then used to reduce the dimension of the text vectors; The image feature extractor consists of a VGG19 network and a fully connected layer. Preprocessed images are fed into the VGG19 network to produce image vector representations, which are then passed through a fully connected layer to adjust their dimension so that it matches the dimension of the text vectors.

2.1.1. Text Feature Extractor

During text pre-training, the word2vec model is used to represent each word in the text as a word embedding vector. The structure of the Text-CNN network is shown in Figure 1. For the I th word x_i in each complete sentence, the corresponding k-dimensional word embedding vector is represented as $T_i \in R^k$; therefore, a sentence containing n words is represented as:

$$T_{1n} = T_1 \oplus T_2 \oplus \cdots \oplus T_n \tag{1}$$

In this context, ($\oplus$ is the concatenation operator. For a matrix $n \times k$, a convolution operation is performed on $T_{i:i+h-1}$ using a convolution kernel of size h, and the feature t_i is output. The specific convolution operation is as follows:

$$t_i = \sigma\left(W \cdot T_{i:i+h-1} + b\right) \tag{2}$$

Here, $\sigma(\cdot)$ represents the ReLU activation function, W represents the weight matrix of $h \times k$, and b represents the bias parameters. Apply the convolution kernel to a sentence, moving down one step at a time from $(i = 1, \cdots, n-h+1)$; perform convolution on $T_{1:h}$ to obtain t_1, perform convolution on $T_{2:h+1}$ to obtain t_2, and then concatenate them to obtain the feature vector:

$$t = \left[t_1, t_2, \cdots, t_{n-h+1}\right] \tag{3}$$

For each feature vector, a max-pooling operation is applied to extract the most significant information, yielding a feature extracted by a specific convolutional kernel. This process is repeated until features extracted by all convolutional kernels are obtained. For each window size, there are n_h different convolutional kernels; therefore, for c different window sizes, there are $c \cdot n_h$ convolutional kernels. The text features after the max-pooling operation are denoted as $R_{T_c} \in R^{c \cdot n_h}$. Following max-pooling, the features from different window sizes are concatenated directly, and a fully connected layer is used to ensure that the final text feature representation $R_T \in R^p$ has the same dimension p as the visual feature representation:

$$R_T = \sigma\left(W_{tf} \cdot R_{T_c}\right) \tag{4}$$

Here, W_{tf} denotes the weight matrix of the fully connected layer.

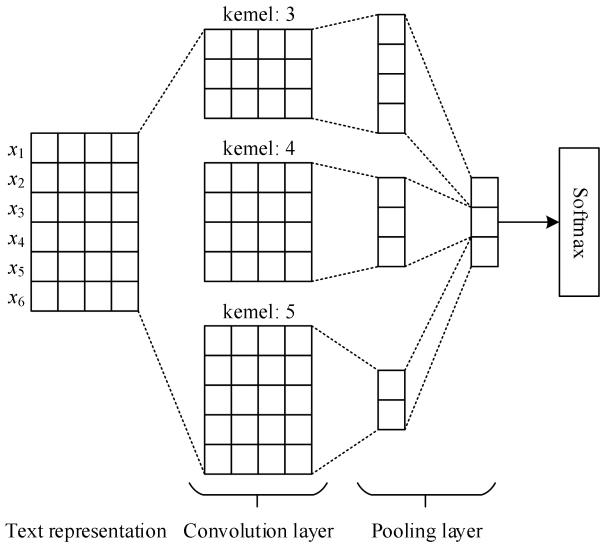


Figure 1. Text-CNN Network Structure

2.1.2. Image Feature Extractor

This paper uses the VGG19 network to extract image features; its input consists of image data, represented as V. The VGG19 network takes image matrices of size $128\times128\times3$ as input, and its specific structure is shown in Figure 2. A fully connected layer was added after the VGG19 network to adjust the dimension of the final image feature representation to p. During joint training with the text feature extractor, the parameters of the pre-trained VGG19 neural network were kept fixed to avoid overfitting.

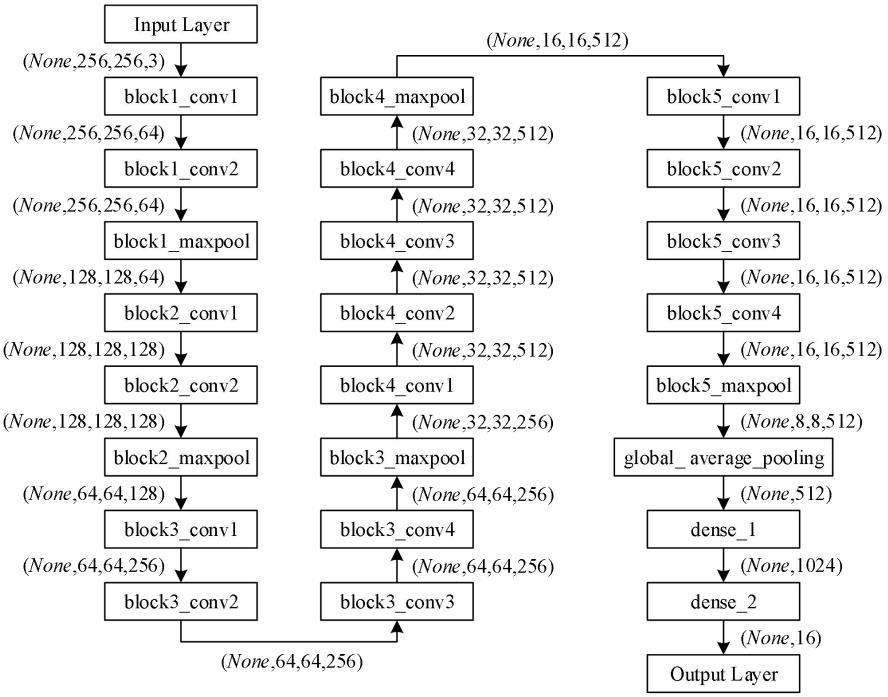


Figure 2. The specific structure of the VGG19 network

Let the visual feature representation of p be denoted as $R_V \in R^p$. The operation in the final layer of the visual feature extractor can be expressed as:

$$R_V = \sigma\left(W_{vf} \cdot R_{V_{vgg}}\right) \tag{5}$$

In this context, $R_{V_{vgg}}$ refers to the image feature representation obtained from the preprocessed VGG19, while W_{vf} refers to the weight matrix of the fully connected layer in the image feature extractor.

The multimodal feature representation is composed of the text feature representation R_T and the image feature representation R_V, specifically defined as:

$$R_F = R_T \oplus R_V \in R^{2p} \tag{6}$$

In this context, R_F is the output of the multimodal feature extractor. The multimodal feature extractor is represented as:

$$G_f\left(M;\theta_f\right) \tag{7}$$

In this context, M typically refers to a set of social media posts containing text and images, which serve as the input to a multimodal feature extractor, while θ_f denotes the parameters to be learned.

2.2. Multimodal Feature Fusion

Multimedia news on social networks is highly diverse, consisting primarily of text but also incorporating a wide range of modalities such as images, videos, and audio. It is difficult for a single modality to provide the complete information contained in multimedia news. Multimodal data fusion effectively integrates information from multiple modalities, leveraging the strengths of each to achieve comprehensive information synthesis. By fusing different modal features in multimodal feature fusion experiments, we can detect rumors from three perspectives simultaneously: text feature analysis, image feature analysis, and speech feature analysis. This approach fully utilizes the information interaction between different modal features, resulting in more comprehensive features and improved effectiveness in rumor detection.

Multimodal feature fusion involves combining features from multiple different modalities and mapping them into a joint space. The different types of data fusion techniques are as follows:

(1) Concatenation

In this method, text features, speech features, and visual features are simply concatenated into a single feature vector. Thus, the extracted features t, v and a are simply concatenated to form a fused representation:

$$z_C = [t;v;a] \tag{8}$$

The method of directly concatenating and fusing multiple different modalities is referred to as MMF-C. The text feature vector t can be expressed as:

$$t = \left[t_1, t_2, \cdots, t_f\right] \tag{9}$$

The visual feature vector v can be expressed as:

$$v = \left[v_1, v_2, \cdots, v_f\right] \tag{10}$$

The speech feature vector a can be expressed as:

$$a = \left[a_1, a_2, \cdots, a_f\right] \tag{11}$$

The specific calculation process for the composite fusion vector z_C is as follows:

$$\begin{aligned} z_C &= [t;v;a] \\ &= \left[t_1, t_2, \cdots, t_f, v_1, v_2, \cdots, v_f, a_1, a_2, \cdots, a_f\right] \\ &= \left[z_1, z_2, \cdots, z_f, z_{f+1}, z_{f+2}, \cdots, z_{2f}, z_{2f+1}, z_{2f+2}, \cdots, z_{3f}\right] \end{aligned} \tag{12}$$

(2) Hadamard product

In this method, the Hadamard product is used to fuse text features, speech features, and visual

features:

$$z_H = \left[t_f \odot v_f \odot a_f \right] \tag{13}$$

In this context, $\odot$ refers to the Hadamard product operation between matrices, and the method of fusing multiple modalities based on the Hadamard product is called MMF-H.

The specific calculation process for the feature vector z_H, which is fused using the Hadamard product, is as follows:

$$\begin{aligned} z_H &= \left[t_f \odot v_f \odot a_f \right] \\ &= \left[t_1, t_2, \cdots, t_f \right] \odot \left[v_1, v_2, \cdots, v_f \right] \odot \left[a_1, a_2, \cdots, a_f \right] \\ &= \left[t_1 v_1, t_2 v_2, \cdots, t_f v_f \right] \odot \left[a_1, a_2, \cdots, a_f \right] \\ &= \left[t_1 v_1 a_1, t_2 v_2 a_2, \cdots, t_f v_f a_f \right] \end{aligned} \tag{14}$$

Unlike the concatenation method, which simply stacks features from different modalities without effectively leveraging the complementary nature of the information across modalities, the Hadamard product method fuses features from different modalities at corresponding positions, resulting in cross-modal fusion features with richer information.

(3) Kronecker Product

In this method, the Kronecker product is used to fuse text, speech, and visual features:

$$z_K = \left[t_f \otimes v_f \otimes a_f \right] \tag{15}$$

Here, $\otimes$ denotes the Kronecker product between matrices. The method of fusing multiple different modalities based on the Kronecker product is referred to as MMF-K.

The specific calculation process for the eigenvector z_K, which is obtained through Kronecker product fusion, is as follows:

$$\begin{aligned} z_K &= \left[t_f \otimes v_f \otimes a_f \right] \\ &= \left[t_1, t_2, \cdots, t_f \right] \otimes \left[v_1, v_2, \cdots, v_f \right] \otimes \left[a_1, a_2, \cdots, a_f \right] \\ &= \left[t_1 v_1, \cdots, t_1 v_f, \cdots, t_2 v_1, \cdots, t_2 v_f, \cdots, t_f v_1, \cdots, t_f v_f \right] \otimes \left[a_1, a_2, \cdots, a_f \right] \\ &= \left[t_1 v_1 a_1, \cdots, t_1 v_1 a_f, \cdots, t_f v_f a_1, \cdots, t_f v_f a_f \right] \\ &= \left[z_1, \cdots, z_f, \cdots, z_{f \times f \times f - f + 1}, \cdots, z_{f \times f \times f} \right] \end{aligned} \tag{16}$$

Unlike Hadamard product-based fusion, which combines only the corresponding features across modalities—leaving individual elements from different modalities unfused—multimodal fusion using the Kronecker product yields a fused feature that incorporates the results of fusing every element across all modalities. This is equivalent to performing data augmentation on the features during the fusion process, ensuring that all features from different modalities are fully integrated.

2.3. False Content Detection Based on a Multimodal Transformer

2.3.1. Overview of Methods

This section proposes MTTV, a method for detecting fake media based on a multimodal Transformer that utilizes two-level visual features. MTTV fully leverages visual information from image data and ensures sufficient interaction between multimodal information to improve the accuracy of fake media detection. MTTV primarily consists of two sub-processes:

(1) Embedding multimodal media content

This process extracts two-level visual information from media images and models it together with media text as a unified sequence representation. First, a ResNet network is used to extract global visual features from the images; Faster RCNN is used to identify entities appearing in the images and represent them as embedding vectors; and learnable word embeddings are used to represent the media text.

(2) Encoding and Classification

This process encodes the multimodal embedding sequence to ensure that multimodal information

from the media content interacts fully, and performs the final classification task. It employs a multi-layer Transformer encoder to encode the multimodal embedding sequence, thereby obtaining a media content representation vector containing high-quality semantic information. Finally, a fully connected layer is used to convert the representation vector into a predicted media authenticity label.

2.3.2. Embedding multimodal content

The input to this process consists of text and associated images from a news article, and the output is a sequence of embedding vectors containing multimodal information, which serves as the input to the subsequent Transformer encoder. The process first constructs text embeddings, global visual embeddings, and entity-level visual embeddings separately, and then fuses them into multimodal embeddings.

(1) Text Embeddings

First, according to the dictionary, the original text is represented as a sequence in one-hot encoding format, i.e., $T=\left[t_1,t_2,\ldots,t_n\right]$, where n is the maximum length of the text sequence. Then, using a learnable embedding matrix $W^{\tau}\in\mathbb{R}^{H\times|V|}$, the one-hot encoding is mapped to a text embedding vector. This process can be expressed by the following formula:

$$R_T \Leftrightarrow \left[w_1,w_2,\ldots,w_n\right]=\left[W^T t_1,W^T t_2,\ldots,W^T t_n\right] \tag{17}$$

Here, H denotes the length of each text embedding vector, $|V|$ is the length of the vocabulary, and $w_i\in \mathrm{R}^H$ represents the embedding vector corresponding to word t_i; the sequence of all text embeddings in a sentence is denoted as R_T. In this work, both the vocabulary V and the text embedding matrix W^T are initialized using a pre-trained BERT model.

(2) Global Visual Embeddings

Global visual embeddings are visual features extracted from the entire news image. First, ResNet is used to extract global features from the image; then, pooling layers and linear mappings are applied to transform the image features into sequence-based features similar to the text embedding sequence R_T. This process can be expressed by the following formula:

$$\begin{aligned} F_V &= \mathrm{Re}\,snet(V) \\ \bar{F}_V &= AvgPool\left(F_V,g\right) \\ R_V &= \bar{F}_V W^V \\ R_V &\Leftrightarrow \left[v_1,v_2,\ldots,v_g\right] \end{aligned} \tag{18}$$

In the above formula, $F_V\in \mathrm{R}^{7\times7\times2048}$ represents the output of the final convolutional layer of ResNet. To transform the image features into a sequential format, an average pooling operation is used to downsample the features and merge the first two dimensions of F_V into a single dimension, yielding $\bar{F}_V\in\mathbb{R}^{g\times2048}$, where $g\in\{1,2,3,\ldots,49\}$ is a hyperparameter used to control the length of the global visual features. Finally, a linear projection is used to map F_V to $R_V\in\mathbb{R}^{g\times H},W^V\in R^{2048\times H}$, where $R_V\in\mathbb{R}^{g\times H},W^V\in R^{2048\times H}$ is a learnable linear projection matrix. Finally, the global visual embedding R_V is expressed as a sequence of embeddings, namely $\left[v_1,v_2,\ldots,v_g\right],v_i\in\mathbb{R}^H$.

(3) Entity-level visual embedding

Entity-level visual embedding refers to the embedding of visual entities appearing in news images. This paper uses Faster RCNN to detect entities in images. News images are fed into the pre-trained Faster RCNN to obtain detection results, from which the top r entities with the highest confidence scores, denoted as $\left[E_1,E_2,\ldots,E_r\right]$, are selected, this is used to construct entity-level visual embeddings, where r is an adjustable hyperparameter. Next, we continue to use ResNet to obtain the embedding representations of these visual entities, and then use a linear projection to adjust the dimensions of the embedding representations. The above process can be expressed by the following formula:

$$\begin{aligned}
&\left[E_1, E_2, \ldots, E_r\right] = \text{FasterRCNN}(V) \\
&\left[\overline{E}_1, \overline{E}_2, \ldots, \overline{E}_r\right] = \text{ResNet}\left(\left[E_1, E_2, \ldots, E_r\right]\right) \\
&R_E \Leftrightarrow \left[e_1, e_2, \ldots, e_r\right] = \left[\overline{E}_1 W^E, \overline{E}_2 W^E, \ldots, \overline{E}_r W^{\overline{E}}\right]
\end{aligned} \tag{19}$$

Here, E_i refers to the object image cropped from the original image based on the bounding boxes obtained from the object detection results; $\overline{E}_i \in \mathrm{R}^{2048}$ is the output of the final pooling layer of the ResNet network; $W^E \in R^{2048\times H}$ is a learnable parameter matrix; and $e_i \in \mathbb{R}^H$ is an embedding representation of an object. For simplicity, we denote the object embedding sequence $\left[e_1, e_2, \ldots, e_r\right]$ as the object-level visual feature R_E.

(4) Multimodal Embeddings

Multimodal embeddings are obtained by combining the three embeddings described above and incorporating position and type embeddings to enhance semantic information. Similar to the position and paragraph embeddings in the BERT model, the position embeddings in MTTV describe the positional information within each modality, while the type embeddings are used to distinguish the modality type of each embedding. The process of constructing multimodal embeddings is shown in the following formula:

$$\begin{aligned}
\overline{R}_T &= \left[w_1 + w^{type}, w_2 + w^{type}, \ldots, w_n + w^{type}\right] + W^{pos} \\
\overline{R}_V &= \left[v_1 + v^{type}, v_2 + v^{type}, \ldots, v_g + v^{type}\right] + V^{pos} \\
\overline{R}_E &= \left[e_1 + e^{type}, e_2 + e^{type}, \ldots, e_r + e^{type}\right] + E^{pos}
\end{aligned} \tag{20}$$

$$D = \overline{R}_V \oplus \overline{R}_E \oplus \overline{R}_T \tag{21}$$

In Equation (21), $w^{type}, v^{type}, e^{type} \in \mathbb{R}^H$ represents the type embedding vectors for text, global images, and entity images, respectively; $V^{pos} \in \mathbb{R}^{n\times H}, W^{pos} \in \mathbb{R}^{g\times H}, E^{pos} \in \mathbb{R}^{r\times H}$ represents the in-modal position embedding matrices for the three embeddings, which allow the position information inherent in the original images and text sequences to be incorporated into the multimodal embeddings. Both the position embeddings and the type embeddings are learnable parameters.

2.3.3. Coding and Classification

This section performs fake news detection by encoding and classifying multimodal embeddings using D. First, a Transformer encoder with N layers is used to perform deep encoding of the multimodal embeddings to obtain high-quality representations of news content, where N is an adjustable hyperparameter.

The core principle of the Transformer encoder is to compute multi-head self-attention on sequential inputs; essentially, it models sequential data by calculating the correlation between any two elements in the sequence. The attention mechanism in the Transformer encoder can be described by the following equation:

$$Attention(Q, K, V) = Soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{22}$$

Here, Q, K, V stands for *Query*, *Key*, and *Value*, respectively, and d_k represents the length of K. In the MTTV model, we compute self-attention for the multimodal embedding D. After applying different linear projections to D, it is treated as Q, K, V, and self-attention scores are calculated.

The multi-head self-attention mechanism takes a single sequence as input and computes multiple sets of independent self-attention scores simultaneously. These results are then concatenated and aggregated using a weight matrix. This process can be expressed by the following formula:

$$\begin{aligned}
&\text{MultiHead}(D) = \text{Concat}\left(head_1, \ldots, head_n\right) W^O \\
&\text{where head}_i = \text{Attention}\left(DW_i^Q, DW_i^K, DW_i^V\right)
\end{aligned} \tag{23}$$

Here, $W_i^{Q/K/V} \in R^{H\times d_h}$ denotes the projection matrix specific to each attention head, which projects

the input sequence into different local spaces for computation to ensure the diversity of attention heads. n represents the number of attention heads. The matrix $W^O \in R^{hd_h \times H}$ aggregates the outputs of multiple concatenated self-attention groups. The encoding module of MTTV uses N cascaded Transformer encoders to process the multimodal embedding D. This encoding process can be described by the following formula:

$$\begin{aligned} D_0 &= D \\ \hat{D}_i &= LN\left(D_{i-1} + MultiHead\left(D_{i-1}\right)\right) \\ D_i &= LN\left(\hat{D}_i + FFN\left(\hat{D}_i\right)\right), i \in \{1,2,\ldots,N\} \end{aligned} \tag{24}$$

In the above formula, the multimodal embedding D obtained from the embedding module is first used as the initial sequence D_0. The Transformer encoder in each layer first computes the multi-head self-attention of the input sequence D_{i-1}, performs a residual connection with D_{i-1}, and then applies layer normalization (LN) to obtain the intermediate state $\hat{D}_i$. Next, the $\overline{D}_i$ is processed using a feedforward network (FFN) with a hidden layer, followed by another residual connection to the $\overline{D}_i$. Finally, a LN operation is performed to obtain the D_i, which is the final output of the encoder at this layer.

Next, to reduce the dimensionality of the multimodal features, the multimodal sequences are pooled. Unlike common averaging or max-pooling methods, MTTV employs a specialized pooling scheme, as shown in the following equation:

$$R_M = \mathrm{Tanh}\left(D_N^0 W_P\right) \tag{25}$$

Here, D_N^0 represents the first element of the sequence D_N, $W_P \in R^{H \times H}$ is the trainable parameter matrix, Tanh is the activation function, and $R_M \in \mathrm{R}^H$ is the final vector representation of the multimodal news content.

Finally, MTTV uses a linear classifier with a fully connected layer to map R_M to the label domain. The classification process is shown in the following equation:

$$\hat{y} = Soft\max\left(R_M W_c\right) \tag{26}$$

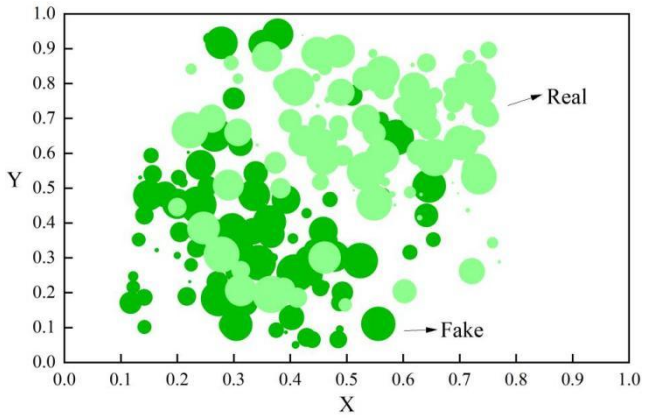
Here, $W_c \in \mathbb{R}^{H \times c}$ is a trainable matrix, c is the number of label categories, and the Softmax function normalizes the output of the linear projection to represent the probabilities that a news sample belongs to each category, i.e., $\hat{y}$; the category with the highest probability is the model's predicted label.

3. Testing for the Detection of Fake Content in the Media and Risk Mitigation

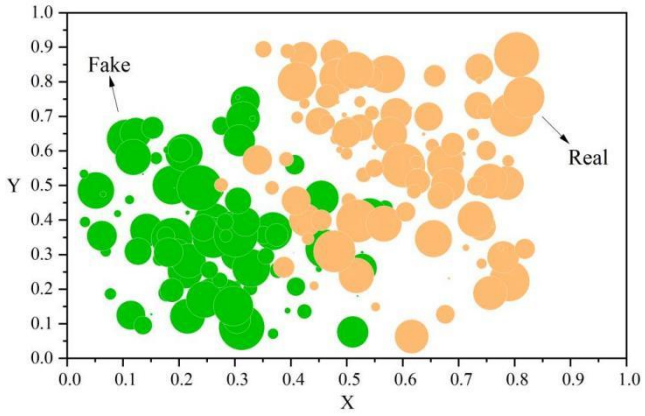
3.1. Feature Extraction and Visualization of Misinformation in the Media

This paper visualizes the feature representations learned using a frequency-domain subnetwork (a), a spatial-domain subnetwork (b), and a Text-CNN (c). We conducted experiments on a Twitter dataset, and the visualizations of multimodal feature extraction are shown in Figure 3. As can be seen, the networks with different configurations demonstrate varying performance in distinguishing between real and fake news. The Text-CNN exhibits significantly stronger data separability than the other two networks, while the spatial-domain network outperforms the frequency-domain network in terms of separability. Specifically, in the visualization of the frequency-domain subnetwork, the overlap rate of feature representations is high. This is because images uploaded to social media are compressed, reducing the difference in the frequency domain between fake images—which are often compressed or tampered with from the start—and real images. In the visualization of the spatial-domain subnetwork, the feature representations exhibit some discriminative power, but some features overlap, with the overlap being most pronounced in the central region. In contrast, in the visualizations of the Text-CNN, the feature representations exhibit relatively distinct boundaries. Based on these observations, we can conclude that the spatial domain is more effective than the frequency domain at distinguishing fake news. Furthermore, the spatial and frequency domains are complementary in their analysis of detected images; therefore, integrating spatial and frequency domain information enables the Text-CNN to

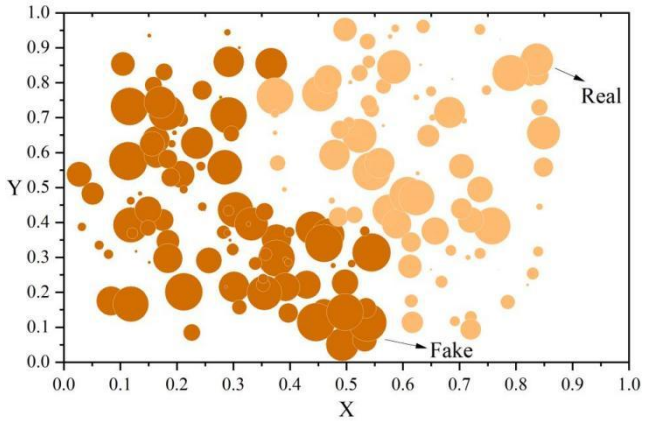
obtain better feature representations and achieve superior performance compared to single-domain networks.



(a) Frequency domain sub-network



(b) Spatial domain sub-network



(c) Text-CNN

Figure 3. Multimodal Feature Extraction Visualization

3.2. Ranking of Feature Extraction Results

To more clearly observe the extent to which each feature influences the detection performance of fake media content, this paper ranks the importance of all features. Since random forests evaluate the importance of a feature by randomly adding noise to a specific feature in the out-of-bag data and

comparing the resulting changes in classification error, this method has proven to be effective in practice. Therefore, this paper employs random forests to characterize the importance of each feature. The importance values of each feature were obtained using the `feature_importances_` method. The ranking of feature importance for fake media content is shown in Figure 4. To highlight more direct differences, the feature importance values were not normalized here. Among user features, the number of followers (0.312), user reputation score (0.300), registration duration (0.271), and number of posts (0.250), as well as the number of likes (0.305) among dissemination features, exhibit high levels of importance. Among media content features, with the exception of the sentiment bias of comments, the importance of the remaining features is generally low. This is because fabricated media content typically mimics the writing style of authentic media information, with extremely similar vocabulary and sentence structures, resulting in weak discriminatory power. In contrast, user characteristics serve as a user's "business card" on social networks and, to some extent, represent the user's essence. Dissemination characteristics, meanwhile, embody the "collective wisdom" of social network users; in particular, the number of likes indicates the level of acceptance the information receives among netizens and, to a certain degree, represents the essence of the information itself.

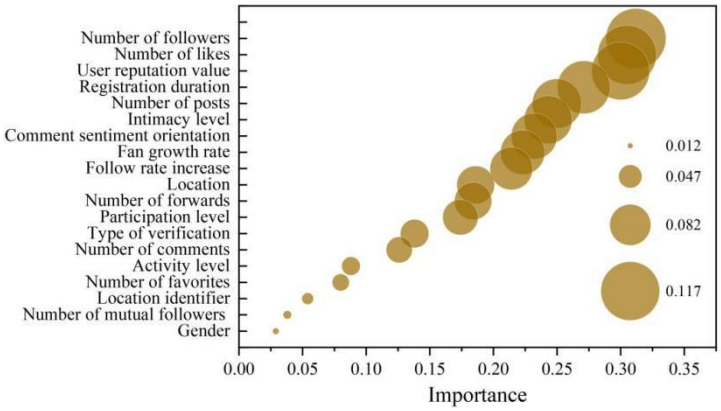


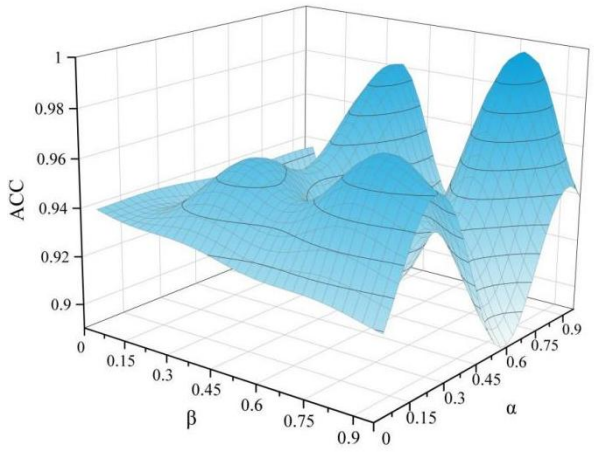
Figure 4. Ranking of the Importance of Characteristics of False Content in Media

3.3. Effectiveness of Media Misinformation Detection

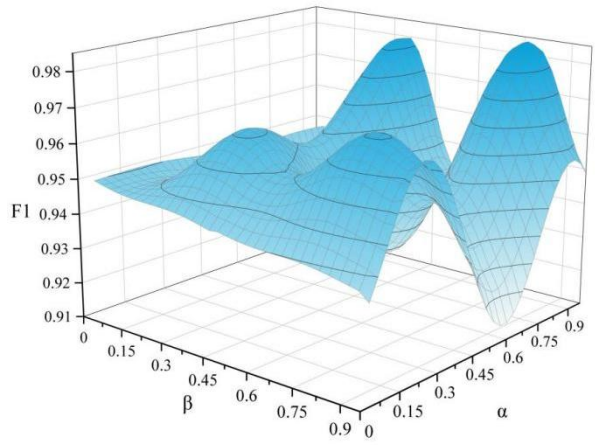
3.3.1. Parameter Analysis

The objective function of the model in this paper uses α and β to assign relative weights to the classification loss (α) and the MMD loss (β). To investigate the impact of the values of α and β on the experimental results, we varied their values from 0 to 0.95 in increments of 0.15. The results of the parameter tuning experiments are shown in Figure 5.

The experiments demonstrate that the model's performance is more sensitive to changes in the β value (MMD loss), indicating the feasibility and effectiveness of calculating semantic consistency between the title and the body. When β is set to a higher value, the accuracy and F1 score are higher, indicating superior classifier performance. In Figure 5(a), the accuracy ranges from 0.890 to 0.995, and changes in the values of α and β have little effect on accuracy. In Figure 5(b), the F1 score ranges from 0.912 to 0.984, a difference of approximately 7 percentage points. The experimental results show that the model performs best when $\alpha = 0.75$ and $\beta = 0.88$.



(a) The influence of loss function parameters on accuracy



(b) The influence of loss function parameters on F1 score

Figure 5. Parameter detection experimental results

3.3.2. Model Comparison

In this experiment, precision, recall, F1 score, and accuracy were used as metrics to evaluate model performance. All models were trained and tested using five-fold cross-validation. Model performance is represented by the mean of each evaluation metric obtained from cross-validation. The detection performance of different models is shown in Table 1. In the table, "+" and "–" denote the metric values corresponding to real news and fake news, respectively. By comparing the performance of each model, it can be observed that, overall, deep learning methods outperform methods combining hand-engineered features with machine learning, indicating that the features extracted by deep learning models yield better classification results. MTTV achieved an accuracy of 0.922. The primary reason Transformer outperformed Word2vec is that its deep network architecture and multi-dimensional convolutional kernels can capture more comprehensive text features. On the one hand, this is due to MTTV's deep network architecture; on the other hand, MTTV integrates both network structural features and node features, whereas DeepWalk and Node2vec only exploit network structural features. Consequently, the Transformer model outperforms XGBoost and Node2vec.

Table 1. Detection performance of different models

Model	Word2vec	Text-CNN	DeepWalk	Node2vec	XGBoost	MTTV
Precision+	0.654	0.701	0.678	0.806	0.897	0.892
Recall+	0.849	0.769	0.690	0.801	0.850	0.916
F1-score+	0.791	0.693	0.837	0.858	0.818	0.874
Precision-	0.687	0.823	0.725	0.846	0.800	0.927
Recall-	0.815	0.780	0.767	0.892	0.869	0.904

F1-score-	0.805	0.701	0.732	0.816	0.823	0.945
Accuracy	0.757	0.794	0.720	0.824	0.880	0.922

Given the significant variation in the duration of news dissemination, the experiment used the size of the dissemination network as a control variable in place of dissemination time. Specifically, for the set of posts comprising each news item, nodes appearing in the top 8%, 16%, 24%, 32%, 40%, 48%, 56%, 64%, 72%, 80%, 88%, and 96% of the propagation network were extracted to construct propagation networks of varying sizes. Furthermore, the data was trained and tested using the multimodal MTTV proposed in this paper (with model parameters set to optimal values). This process employed a five-fold cross-validation method, and the performance of the multimodal MTTV under different network sizes is shown in Table 2. As the scale of the propagation network increases, the accuracy of the proposed multimodal MTTV improves to varying degrees, with distinct inflection points in the trend. When the network scale is below 48%, the model's accuracy is very low; however, once the scale reaches 48%, the accuracy improves significantly and stabilizes at a relatively steady level. The multimodal MTTV demonstrates the ability to detect fake news at an early stage.

Table 2. Performance under different scale communication networks

Scale	Precision+	Recall+	F1-score+	Precision-	Recall-	F1-score-	Accuracy
8%	0.425	0.416	0.457	0.413	0.458	0.442	0.457
16%	0.433	0.417	0.473	0.480	0.505	0.449	0.486
24%	0.452	0.450	0.537	0.492	0.511	0.449	0.544
32%	0.456	0.496	0.556	0.582	0.582	0.464	0.547
40%	0.601	0.582	0.561	0.608	0.611	0.473	0.562
48%	0.602	0.588	0.586	0.617	0.613	0.516	0.572
56%	0.720	0.691	0.682	0.720	0.730	0.743	0.683
64%	0.740	0.739	0.711	0.822	0.829	0.797	0.746
72%	0.751	0.749	0.733	0.828	0.848	0.805	0.754
80%	0.814	0.782	0.783	0.831	0.859	0.835	0.806
88%	0.845	0.816	0.863	0.835	0.875	0.855	0.829
96%	0.869	0.816	0.886	0.842	0.891	0.889	0.862

3.4. Mitigating the Risks of Misinformation in the Media

Data- and algorithm-driven communication models pose significant challenges to the authenticity of information dissemination, while the entry of diverse actors into the media landscape challenges journalistic professionalism. Consequently, in the face of this complex and fragmented information landscape, both media professionals and the public must continuously enhance their media literacy to navigate the rapidly evolving communication environment.

In addition to possessing traditional information gathering and processing skills, media professionals must also flexibly utilize artificial intelligence technologies and new media tools to enhance their capabilities. Given the collapse of public discourse order and the increasingly profit-driven nature of platforms, improving the professional competence of media practitioners has become an urgent priority. On an individual level, the decentralization of communication power has led to an exponential expansion of communication actors. The public must not only improve their media literacy but also extend their engagement to information production and social collaboration, consciously maintaining the order of the public discourse space so that it develops in a positive direction. Although numerous obstacles remain to fully realizing the noble vision of rational human interaction and creating a shared, orderly discursive space, this should at least become a collective ideal pursued by the public.

Past research indicates that AI-driven solutions and decision-making are, at their core, complex social system problems defined by a mix of ambiguous information, stakeholders and decision-makers with conflicting values, and significant consequences. Although OpenAI asserts that the AIGC applications it has developed contain no ideological bias and will not influence the ideologies of any nation, the gap between the design and operation of algorithms and our understanding of their ethical implications may have serious consequences for individuals, groups, and society as a whole.

4. Conclusion

We propose a multimodal feature extraction method for media misinformation that combines text and image features, and develop a misinformation detection model based on a multimodal Transformer. The following conclusions were drawn from the test analysis:

(1) Among the modal features for detecting media misinformation, user features such as follower count (0.312), user reputation score (0.300), registration duration (0.271), and number of posts (0.250), as well as dissemination features such as the number of likes (0.305), exhibit high feature importance for detection.

(2) Parameter analysis indicates that the detection performance is optimal when $\alpha = 0.75$ and $\beta = 0.88$. In terms of comprehensive metrics such as precision, recall, F1 score, and accuracy, the MTTV model proposed in this paper outperforms all other models, with an accuracy of 0.922. This model can identify fake news and mitigate the risks associated with false content.

References

1. Liu, Y., & Li, J. (2024). "Expansion" and" Obstacles": The New Wave of Intelligent Media Empowering the Development of Film and Television Arts with Digital Technology. Journal of Social Science Humanities and Literature, 7(6), 57-66.
2. Huang, L., Gao, B., & Gao, M. (2023). Smart media era: The third transformation in the age of internet communication. In Value realization in the phygital reality market: Consumption and service under conflation of the physical, digital, and virtual worlds (pp. 77-97). Singapore: Springer Nature Singapore.
3. Zhang, Y., Liu, Y., & Guo, Z. (2025). Optimising news dissemination pathways in the media convergence era: an interactive digital media technology approach. International Journal of Information and Communication Technology, 26(29), 110-126.
4. Shoaib, M. R., Wang, Z., Ahvanooey, M. T., & Zhao, J. (2023, November). Deepfakes, misinformation, and disinformation in the era of frontier AI, generative AI, and large AI models. In 2023 international conference on computer and applications (ICCA) (pp. 1-7). IEEE.
5. Bontridder, N., & Poullet, Y. (2021). The role of artificial intelligence in disinformation. Data & Policy, 3, e32.
6. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.
7. Liang, F., Hatcher, W. G., Liao, W., Gao, W., & Yu, W. (2019). Machine learning for security and the internet of things: the good, the bad, and the ugly. Ieee Access, 7, 158126-158147.
8. Bhardwaj, P., Yadav, K., Alsharif, H., & Aboalela, R. A. (2021, September). GAN-based unsupervised learning approach to generate and detect fake news. In International Conference on Cyber Security, Privacy and Networking (pp. 384-396). Cham: Springer International Publishing.
9. Zobaed, S., Rabby, F., Hossain, I., Hossain, E., Hasan, S., Karim, A., & Md. Hasib, K. (2022). Deepfakes: Detecting forged and synthetic media content using machine learning. In Artificial intelligence in cyber security: impact and implications: security challenges, technical and ethical issues, forensic investigative challenges (pp. 177-201). Cham: Springer International Publishing.
10. AbdElminaam, D. S., Sherif, N., Ayman, Z., Mohamed, M., & Hazem, M. (2023). DeepFakeDG: A deep learning approach for deep fake detection and generation. Journal of Computing and Communication, 2(2), 31-37.
11. Patel, Y., Tanwar, S., Gupta, R., Bhattacharya, P., Davidson, I. E., Nyameko, R., ... & Vimal, V. (2023). Deepfake generation and detection: Case study and challenges. IEEE Access, 11, 143296-143323.
12. Iqbal, T., & Qureshi, S. (2022). The survey: Text generation models in deep learning. Journal of King Saud University-Computer and Information Sciences, 34(6), 2515-2528.
13. Lin, Y., Ruan, T., Liu, J., & Wang, H. (2023). A survey on neural data-to-text generation. IEEE Transactions on Knowledge and Data Engineering, 36(4), 1431-1449.
14. Garneau, N. (2020). The case of fake news and automatic content generation in the era of big data and machine learning. MÉDIAS SOCIAUX, 139.

15. Wang, X., Xie, H., Ji, S., Liu, L., & Huang, D. (2023). Blockchain-based fake news traceability and verification mechanism. Heliyon, 9(7).
16. Monroy, F. J., Mohatar, O. D., & Ortigosa, Á. (2020). Tracking news stories using blockchain to guarantee their traceability and information analysis. IJIMAI, 6(3), 39-46.
17. Taillon, P. (2020). 13 From veracity to traceability. Misinformation in Referenda, 25.
18. Huckle, S., & White, M. (2017). Fake news: A technological approach to proving the origins of content, using blockchains. Big data, 5(4), 356-371.