ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Research on Deep Learning-Based Legal Text Classification Methods

Kunchi Wang[1], *Enchang Liu*[2] *and Yuchen Meng*[3,*]

[1] Law School, Xi’an JiaotongUniversity, Xi’an710049, Shanxi, China
[2] Law School, East China University of Political Science and Law, Shanghai 710049, China
[3] Corresponding author: Yuchen Meng, Law School, Xi’an JiaotongUniversity, Xi’an 710049, Shanxi, China

SUMMARY: *Legal documents such as rulings, contracts, and statutes are typically voluminous and complex, making manual processing inefficient. This paper employs deep learning algorithms for legal document text classification, proposing two deep learning models: TextCNN and GRU. To achieve superior classification performance, a combined model TextCNN-GRU was constructed using a linearly weighted model fusion approach. The highest classification accuracy was attained when both models were fused with equal weights of 0.5. Comparisons with other single models and the high-performing CNN-HAN model demonstrate that TextCNN-GRU significantly outperforms in accuracy, macro-precision, macro-recall, and macro* F_t *scores. Generalization experiments further validate the model's performance. To explore further optimization directions, 500 misclassified samples were analyzed, revealing issues such as ambiguous category boundaries, non-standard textual expressions, and mixed narratives involving multiple events and roles. Future improvements could include refining the dataset, incorporating additional domain knowledge, and implementing secondary classification schemes to enhance classification performance.*

KEYWORDS: *TextCNN; GRU; Linear Weighting; Legal Text Classification*

1 Introduction

In the judicial sphere, as public legal awareness continues to grow and the rate of new cases increases, coupled with the ongoing updates and refinements to laws to adapt to society's ever-evolving realities, vast amounts of new data emerge daily [1-4]. This data originates from various civil and criminal case files and judgments, as well as supplementary expansions to laws and regulations and judicial interpretations [5]. Concurrently, China's judicial informatization initiatives have advanced steadily. After screening and cleaning, this data is increasingly made publicly available. The China Judgments Online platform, operated by the Supreme People's Court, hosts over 100 million judicial documents and continues to grow, establishing itself as the largest legal database in the country [6-9].

On the other hand, as data volumes expand, the workload for judicial professionals grows increasingly heavy. Judges and lawyers must not only review vast historical case precedents for reference but also conduct in-depth analysis and research on new laws and regulations, as well as supplementary expansions to existing legal provisions [10-13]. In recent years, artificial intelligence technologies, represented by deep learning and natural language processing, have achieved continuous breakthroughs. Their research outcomes have driven development across

---

*yuchenm@stu.xjtu.edu.cn

https://doi.org/10.65102/is2026974

manufacturing, healthcare, education, and other sectors, enhancing productivity and alleviating labor burdens [14-17]. However, AI-related research in the judicial domain remains largely in its infancy. Text classification is a critical task in legal text processing applications, where diverse legal text processing tasks can be transformed into different types of text classification problems [18-20]. For instance, determining whether a defendant in a case qualifies for a confession circumstance constitutes a simple binary classification problem; analyzing case types involves multi-class classification; and identifying which legal provisions a defendant has violated falls under multi-label classification. To fully leverage the value of legal text data and alleviate the heavy data processing burden on legal professionals, AI researchers have undertaken a series of studies on deep learning-based legal text classification techniques in recent years [21-25].

To understand the application of deep learning in legal text classification, numerous scholars conducted research on this topic between 2018 and 2020. Reference [26] reports research using deep learning in legal document review, comparing results with SVM algorithms and verifying that convolutional neural networks (CNNs) demonstrate superior performance and suitability for legal text classification. Reference [27] introduces the effectiveness of CNNs in text classification and proposes a semi-supervised CNN framework to enhance its application in legal text classification, significantly improving classification accuracy. Reference [28] proposes the “Supreme Court Classifier” (SCC), a system applying CNNs, recurrent neural networks (RNNs), and other methods to classify legal court opinion documents. Training and evaluation on court databases demonstrate the system's high classification accuracy. Reference [29] develops a deep learning-based automatic classification method for legal documents, combining CNNs with bidirectional long short-term memory (LSTM) RNNs, achieving superior classification accuracy over existing approaches. Reference [30] proposes and compares three distinct legal document classification approaches based on convolutional neural networks, recurrent neural networks, and two word embedding schemes. It verifies that the recurrent neural network model with Word2Vec embeddings achieves the highest classification accuracy. Reference [31] introduces a multi-label text classification model based on multi-label text convolutional neural networks, validating its effectiveness in legal document classification and offering an improved approach for legal services.

However, academic attention to deep learning-based legal text classification methods has not been confined to any single period. The highly efficient and accurate classification results it delivers continued to attract significant interest after 2020. Reference [32] proposed a neural network-based model with dynamically variable input lengths and applied it to legal text classification. Test results revealed this model outperformed baseline methods. Reference [33] details the successful application of deep learning in text classification and compares different deep learning techniques in predictive coding, demonstrating that convolutional neural networks outperform other models in legal document classification. Reference [34] explores text classification in both natural language processing and legal domains, collecting extensive datasets from public sources and testing traditional classifiers alongside more complex deep learning and transformer-based models. Reference [35] compares the effectiveness of various text classification techniques, conducting legal text classification research using a large annotated case document collection. It proposes a machine learning algorithm employing domain concepts as features and a random forest as the classifier. Experimental results demonstrate that the proposed method outperforms deep learning systems built with deep neural networks. Reference [36] introduces the POSTURE50K legal multi-label classification dataset and a deep learning architecture employing domain-specific pre-training and label attention mechanisms for multi-label document classification, validating the approach's effectiveness in legal document classification. Reference [37] examines deep learning applications in legal document classification, comparing model performance across various metrics. It reveals that the PRTIRCDNL method outperforms document classification approaches, demonstrating superior recall and accuracy. Reference [38] analyzes deep learning models against manually feature-engineered conditional random field (CRF) approaches, revealing that the optimal model for legal judgment classification excels in legal document summarization and judgment accuracy. Reference [39] proposes a deep learning-based automatic legal consultation classification model and the KP+BiLSTM+Attention model. Test results reveal the KP+BiLSTM+Attention model excels in legal consultation text classification.

This paper explores classification methods for legal texts based on deep learning. First, the open-source Jieba word segmentation tool is employed to establish a specialized dictionary for text processing tasks, completing text preprocessing. Subsequently, Word2vec was employed for textual digitization. Finally, a text classifier integrating TextCNN and GRU was constructed using a linearly weighted model fusion approach. To evaluate optimal model performance, the self-built CAIL-Multi dataset was utilized to determine the optimal fusion weights for each model, with performance comparisons against other models. Misclassification analysis provided insights into model optimization directions.

2 Deep Learning-Based Text Classification Algorithms

With societal development, the volume of texts across various fields has surged dramatically. Traditional text processing methods increasingly fail to meet the demands for informatization and intelligence in text processing. Therefore, this paper proposes applying deep learning-based text processing methods to specialized legal text processing, addressing the needs of the legal profession. The objective is to partially or fully replace feature extraction tasks with deep learning techniques during text processing, thereby enhancing efficiency and reducing manual intervention.

Deep learning-based text classification methods primarily consist of three modules: text preprocessing, text representation, and the classifier:

(1) The text preprocessing module primarily handles preliminary processing of raw corpora, cleaning irrelevant content from the text, performing word segmentation on the original text, and constructing a dictionary.

(2) The text representation module converts text into numerical vectors, serving as both a prerequisite for further text processing and a core component. This paper employs the Word2vec tool to represent text data as low-dimensional, dense word vectors, facilitating subsequent text classification tasks.

(3) The classifier module utilizes Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to learn textual features, performing feature extraction to obtain the text's feature representation.

2.1 Text Preprocessing Module

The original text data contains punctuation, stop words, and other elements. Before performing text processing tasks, the text must first undergo cleaning, word segmentation, and other preprocessing steps to convert it into the data input format required by deep learning models. The main steps in preprocessing Chinese text are as follows:

(1) Word Segmentation: Unlike languages such as English, Chinese text processing tasks require word segmentation. This is because character-level feature selection results in significant loss of n-gram information, and most algorithms directly ignore word order

information in text. Therefore, word-level features are preferable to character-level features. English words are separated by spaces, whereas Chinese characters are contiguous, necessitating complex segmentation processing. This paper employs the Jieba tokenizer for text segmentation.

(2) Stopword Removal: Stopword removal involves filtering out high-frequency words like conjunctions, pronouns, and prepositions that have minimal impact on text processing. For text classification tasks, a stopword list must be constructed to directly eliminate stopwords that contribute nothing to classification.

(3) Dictionary Construction: Count the frequency of words in the text to build a frequency-based dictionary, removing words with low occurrence rates.

The text preprocessing section of this paper primarily utilizes the open-source Jieba word segmentation tool. This tool's main function is to perform word segmentation on text and also includes a stopword list. After importing the text data, the Jieba tool can be used to segment the text and remove words from the stopword list, thereby creating a specialized dictionary for the text processing task at hand. This completes the text preprocessing work.

2.2 Text Representation Module

Text representation involves converting text into numerical vector form. Traditional methods employ the bag-of-words model, which disregards textual information such as word order and represents text using mutually independent vocabulary. Currently, the mainstream approach is Google's Word2vec method [40], with other methods like GloVe also used in practical applications. This paper directly adopts a pre-trained Word2vec model as the word vector. The Word2vec tool is utilized to convert text that has undergone preprocessing into numerical vectors.

Through a series of text preprocessing and representation steps, the original text dataset is ultimately transformed into a numerical matrix similar to digital images. Compared to text representation methods like one-hot encoding, this word vector representation not only reduces data volume but also preserves contextual information within the text. This enables more effective text processing tasks using methods such as neural networks.

2.3 Classifier Module

Compared to traditional machine learning approaches, the most significant difference in deep learning-based text classification methods lies in the use of neural network models to extract textual features. This paper primarily employs TextCNN and GRU to construct classifiers. The following sections describe the implementation of text classification models based on these two deep learning techniques.

2.3.1 TextCNN Model

TextCNN is a convolutional neural network model specifically designed for text classification tasks. It extracts n-gram features by applying one-dimensional convolution operations to word embeddings, effectively capturing local contextual information between words [41]. TextCNN typically employs multiple convolutional kernels of varying sizes to capture diverse feature patterns within text through distinct receptive fields. These features encompass structural characteristics of phrases and sentences. After processing through convolutional layers, the model selects the most prominent features under each kernel via max-pooling layers, thereby reducing feature dimensions while preserving critical information. Ultimately, these extracted feature vectors are concatenated and fed through one or more fully connected layers for classification predictions. The TextCNN model excels in short text classification tasks due to its simple structure, minimal parameters, and high computational efficiency. It finds extensive application in sentiment analysis, news classification, spam detection, and related fields. The results of the TextCNN model are shown in Figure 1.

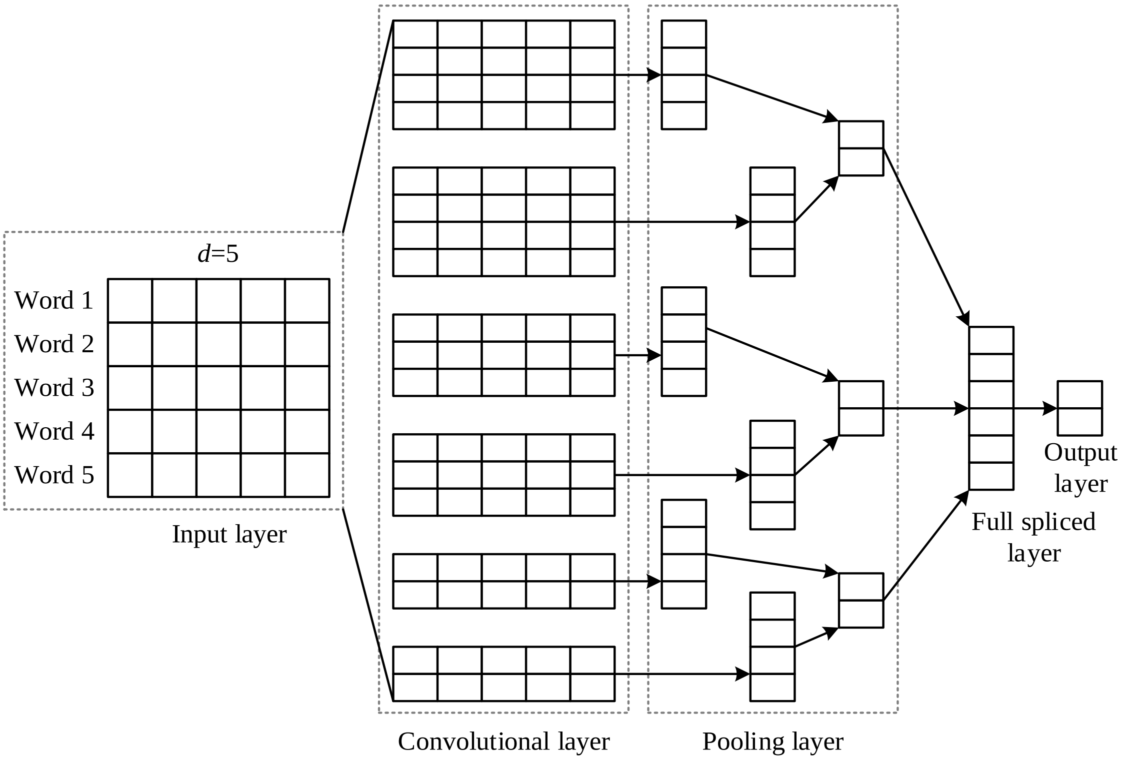


Figure 1: Diagram of the TextCNN model

The specific processing flow of the TextCNN model can be represented as follows:

The input to TextCNN consists of word embeddings representing text. Assuming the input text comprises n words, with each word's embedding vector having dimension d, the entire input can be represented as a matrix:

$$X = [x_1, x_2, \cdots, x_n] \in R^{n \times d} \tag{1}$$

Here, x_i represents the embedding vector of the i th word.

The convolution operation extracts features through the convolution kernel W. Suppose we have a convolution kernel of size h, whose weight matrix is:

$$W \in R^{h \times d} \tag{2}$$

The result of the convolution operation is a sliding window computation on the input matrix X, generating the feature map C:

$$C_i = f(w \cdot x_{i:i+h=1} + b) \tag{3}$$

Here, $x_{ii} + h - 1$ represents the input segment from positions i to $i + h - 1$, b denotes the bias term, and f is the nonlinear activation function (e.g., ReLU).

Pooling layers are used to select the most salient features, typically employing max pooling operations:

$$y = Softmax(W_{fc} \cdot z + b_{fc}) \tag{4}$$

Here, C denotes the feature map, and $\hat{C}$ represents the pooled feature.

For each convolution kernel size h, repeat the aforementioned convolution and pooling operations to obtain multiple feature maps.

Concatenate the pooled feature vectors generated by different convolution kernel sizes h to form the final feature vector:

$$Z = [\hat{C}, \hat{C}, \cdots, \hat{C}m] \tag{5}$$

Among these, $\hat{C}_i$ represents the pooling feature corresponding to the ith convolution kernel size, and m denotes the total number of convolution kernels.

Finally, through one or more fully connected layers, the feature vectors are classified using the Softmax function:

$$y = Softmax(W_{fc} \cdot z + b_{fc}) \tag{6}$$

Here, W_{fc} denotes the weight matrix of the fully connected layer, b_{fc} represents the bias term, and y signifies the classification probability distribution.

2.3.2 GRU Model

GRU (Gated Recurrent Unit) is an enhanced recurrent neural network model designed to address the vanishing gradient problem in traditional RNNs. It controls information flow by introducing two gating mechanisms: the update gate and the reset gate. The update gate determines the balance between the amount of information from the current state and that from the previous state, while the reset gate controls the influence of the previous state on the current computation. This design enables GRUs to capture long-term dependencies more efficiently while maintaining computational and memory advantages, making them highly effective for tasks such as sequence data processing and time series forecasting [42]. The GRU model structure is illustrated in Figure 2.

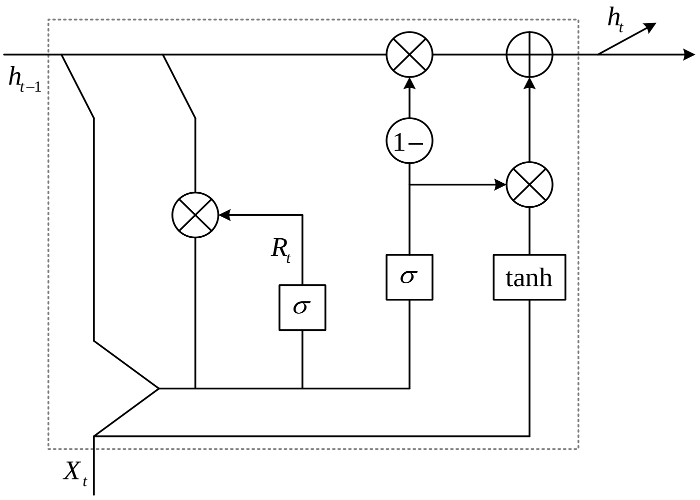


Figure 2: GRU model diagram

The update gate Z_t and reset gate r_t control how the hidden state from the previous time step influences the current time step. Their respective formulas are:

$$Z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{7}$$

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

Here, σ denotes the sigmoid activation function, x_t represents the input at the current time step, h_{t-1} denotes the hidden state from the previous time step, and W_z, U_z, W_r, U_r are the weight matrices, and b_z, b_r are the bias vectors.

The reset gate r_t determines how much of the previous state information to discard, thereby computing the candidate hidden states $\tilde{h}_i$ for the current time step:

$$\tilde{h} = \tanh(W_{x1} + r_t \odot (Uh_{t-1}) + b \tag{9}$$

Here, $\odot$ denotes element-wise multiplication (Hadamard product), tanh is the hyperbolic tangent activation function, W and U are weight matrices, and b is the bias vector.

The update gate Z_t controls how much information in the current hidden state h_t originates from the previous hidden state and the current candidate hidden state:

$$h_t = (1 - Z_t) \odot h_{t-1} + Z_t \odot \tilde{h} \tag{10}$$

In particular, $(1 - Z_t)$ serves as the inverse vector of the update gate, preserving information from the previous time step, while Z_t is used to incorporate new information from the candidate hidden states.

3 Improvements to Deep Learning Models in Text Classification

In the experiments conducted in this paper, both categories of deep learning-based methods exhibit distinct characteristics. Convolutional neural networks perform one-dimensional convolutions on word vectors using multiple distinct convolutional kernels—a process analogous to extracting textual features within sliding window sizes—and are particularly adept at capturing local features or key information within text. Conversely, recurrent neural network classification models are better suited for learning contextual relationships between textual elements. Consequently, the feature types extracted by these two approaches may exhibit certain differences. Building upon this foundation, this paper fuses the two models to construct a combined model, TextCNN-GRU. The specific fusion process is illustrated in Figure 3.

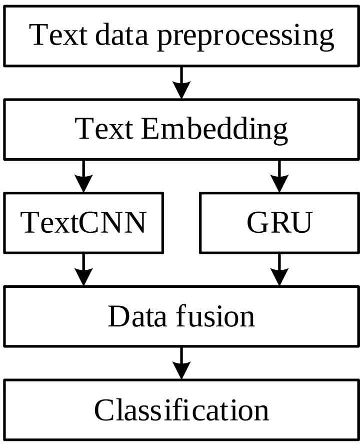


Figure 3: A text classification structure based on model fusion

Common model ensemble methods include bagging and stacking, while this paper employs a linear weighting approach for multi-model fusion. Specifically, during training, each deep learning model outputs a probability prediction matrix. The probability matrices from all models are then weighted averaged. For each sample, the top five values with the highest probabilities are selected as the final prediction result. The accuracy of model classification is improved by manually adjusting the weight values during training.

Text data preprocessing follows the same approach as the two methods above. At the text representation layer, non-static word vector training methods are applied to each model separately, meaning distinct word vector representations are used to enable differentiation during training at the word embedding layer. For simplicity in model training, all models in this paper employ a unified word embedding layer. In the training layer, the word vectors are input into both the TextCNN model and the GRU model to generate feature mapping vectors for the input text. These vectors are then fused using weight matrices before being fed into the Softmax fully-connected layer to predict the text category.

4 Experimental Results and Analysis

4.1 Experimental Dataset

The dataset used in this study is CAIL 2018, which comprises CAIL 2018-Small and CAIL 2018-Large. The CAIL 2018-Small dataset features a higher proportion of single-label data and a smaller dataset size, while CAIL 2018-Large offers a larger dataset that can serve as a supplement to CAIL 2018-Small.

To better evaluate model performance on multi-label legal text classification tasks, this paper reconstructs a new dataset—CAIL-Multi. The specific construction method is as follows: First, high-frequency labels for criminal charges and legal provisions are filtered, retaining only the top 100 most frequently occurring criminal charges and the top 105 most frequently occurring legal provisions from the CAIL 2018-Small dataset. Next, the CAIL 2018-Large dataset is incorporated to extract multi-label cases containing these high-frequency labels, thereby constructing a more representative multi-label dataset. Finally, label distribution is balanced to ensure a relatively even proportion of multi-label cases, enhancing the model's generalization capability in multi-label learning tasks. The dataset is categorized into three label types: crime type, legal provision, and sentence length.

4.2 Evaluation Indicators

Evaluate the model using accuracy:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

Among these, TP denotes the number of correctly identified entities, TN represents the number of misidentified entities, FP indicates the number of other labels incorrectly classified as this label, and FN signifies the number of this label incorrectly classified as other labels.

For multi-class text classification tasks, evaluating the performance of classifiers across different categories involves calculating the macro-weighted average using the Classification-Report evaluation function from the Sklearn module. denoted as macro-precision (P_w) macro-recall (R_w) and macro-F_t value (F_w) as shown in Equations (12) to (14).

$$P_w = \sum_{i=1}^{n} \frac{\omega_i}{n} P_i \tag{12}$$

$$R_w = \sum_{i=1}^{n} \frac{\omega_i}{n} R_i \tag{13}$$

$$F_w = \frac{2 P_i R_i}{P_i + R_i} \tag{14}$$

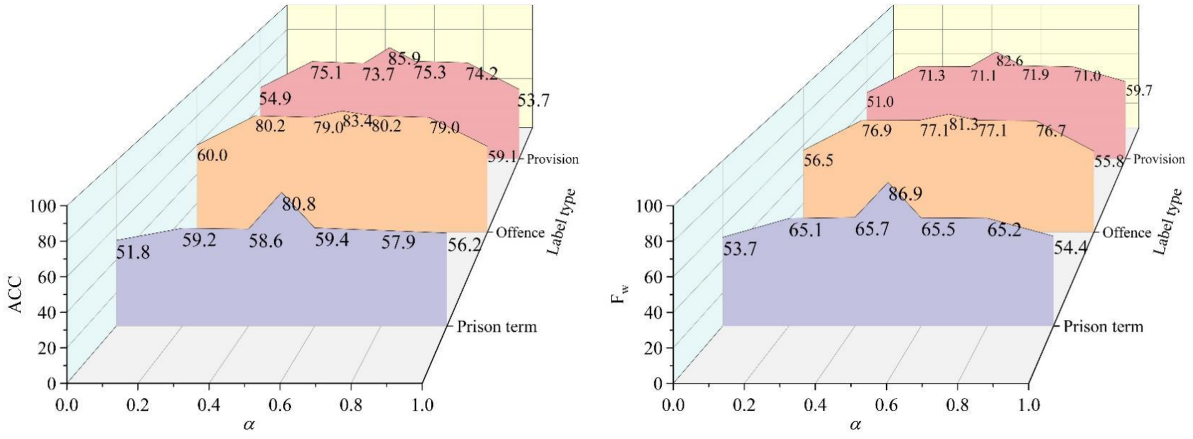
where ω_i, P_i, and R_i, with $i = 1, 2, \cdots, n$, represent the weight, precision, and recall of each category, respectively.

4.3 Fusion Weight

To validate the contribution of each module to the overall performance of the model in this study, experiments were designed to analyze the fusion weights α of TextCNN and the fusion weights β of GRU separately.

(1) TextCNN Fusion Weight α

This study altered the fusion weight α sequentially to values of 0.0, 0.2, 0.4, 0.5, 0.6, 0.8, and 1.0 to observe its impact on the accuracy and macro F1 score of the TextCNN-GRU model. The effect of fusion weight α on model performance is shown in Figure 4. (a) and (b) respectively illustrate the impact of different α values on accuracy for legal clause classification, crime classification, and sentence classification tasks, as well as the impact on macro F1 scores for these three tasks. The figures reveal that an α value of 0.5 achieves optimal accuracy and macro F1 scores across all tasks. Lower α values cause accuracy fluctuations, while excessively high α values introduce redundant semantic information that diminishes the importance of co-occurrence relationships, thereby impairing label relationship modeling. Ultimately, a weight ratio of 0.5 strikes a favorable balance between semantic similarity and co-occurrence relationships.

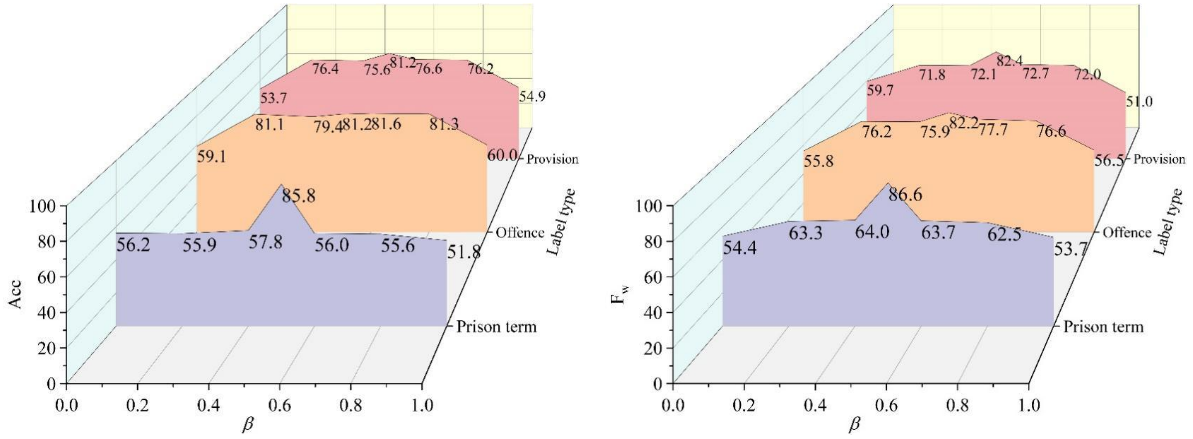


(a) Acc (b) F$_w$

Figure 4: The effect of α on model performance

(2) GRU Fusion Weight β

Similarly, the GRU fusion weight β was varied sequentially through values of 0.0, 0.2, 0.4, 0.5, 0.6, 0.8, and 1.0 to evaluate its impact on the accuracy and macro F1 score of the TextCNN-GRU model across different tasks. The effect of fusion weight β on model performance is shown in Figure 5. (a) and (b) respectively illustrate the impact of different β values on accuracy and macro F1 scores for the legal clause classification, crime classification, and sentence classification tasks. The figures reveal an overall trend of accuracy and macro F1 scores first increasing and then decreasing. When β is set to 0.5, the model achieves optimal performance across multiple tasks, with particularly noticeable improvements in sentence classification. This indicates that setting both TextCNN and GRU weights to 0.5 yields the highest classification accuracy for the model.



(a) Acc (b) F$_w$

Figure 5: The effect of β on model performance

4.4 Comparative Experiments

To validate the effectiveness of TextCNN-GRU in complex legal text classification tasks, experiments were conducted to classify crime charges, legal provisions, and sentencing terms against other models (CNN, RNN, TextCNN, GRU, CNN-HAN[43]). The output categories of the Softmax function were modified to three classes. The training set with these three labels was input into the comparison models for training. The classification performance of each model was validated on the test set, with results shown in Table 1. The combined model demonstrated superior classification performance compared to single models. The classification accuracy of the TextCNN-GRU model designed in this paper exceeded that of the CNN-HAN model by 10%.

Table 1: Comparison of Evaluation Indicators under three classifications

Model	Acc	P_w	R_w	F_w
CNN	0.417	0.423	0.437	0.431
RNN	0.493	0.433	0.402	0.447
TextCNN	0.517	0.552	0.513	0.58
GRU	0.521	0.564	0.515	0.536
CNN-HAN	0.857	0.814	0.842	0.881
TextCNN-GRU	0.867	0.816	0.873	0.887

To further illustrate the classification performance of the TextCNN-GRU model, a three-dimensional plot was generated showing the accuracy difference between TextCNN-GRU and CNN-HAN under identical batch_size and learn_rate settings. The results are depicted in Figure 6. TextCNN-GRU achieved higher test set accuracy than CNN-HAN across all hyperparameter tests, indicating that the proposed hybrid model maintains superior classification accuracy even when compared to the high-performing CNN-HAN model.

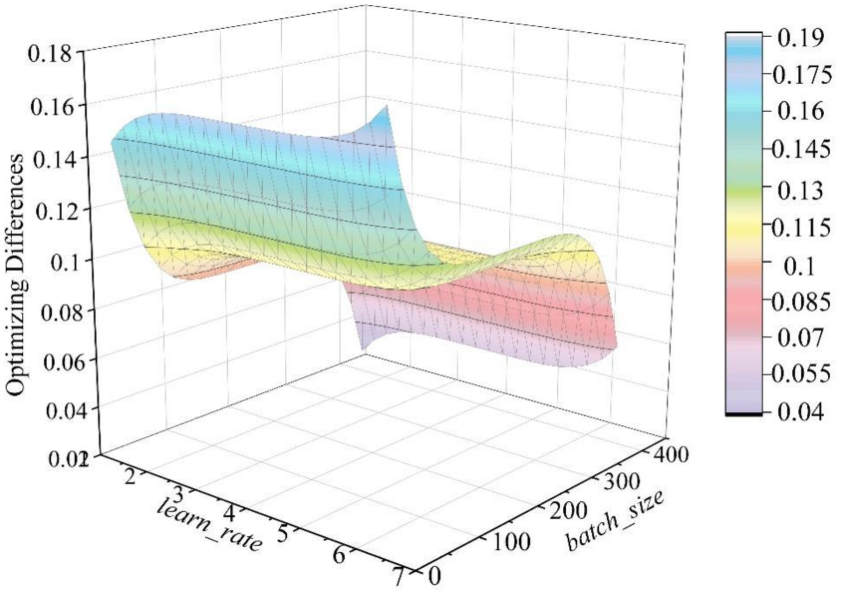


Figure 6: Classification experiment accuracy

4.5 Generalization Experiments

To validate the model's generalization capability, dataset generalization experiments were conducted on the self-built dataset. During testing, the test set data remained unchanged while 1000, 2000, and 3000 data points were randomly sampled from the dataset. These were similarly allocated to the training and validation sets at a 9:1 ratio to measure the model's generalization performance. For each sample size, the experiment was repeated five times, and the average results were calculated to obtain a more reliable evaluation. The average results are shown in Figures 7 to 9.

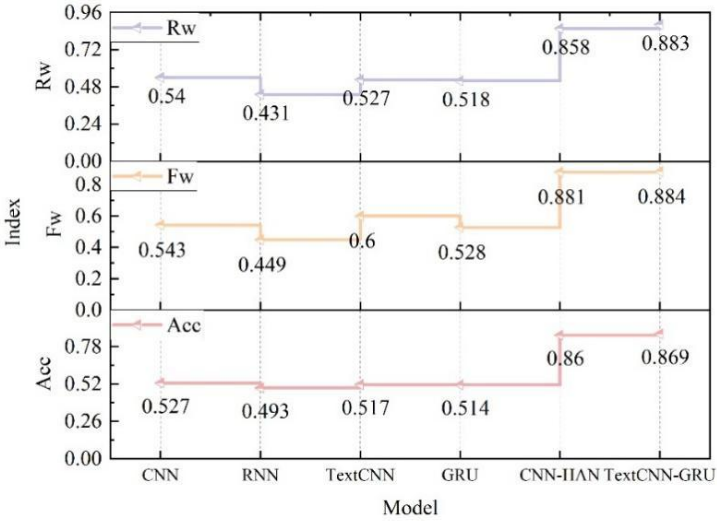


Figure 7: The data cluster is 1000

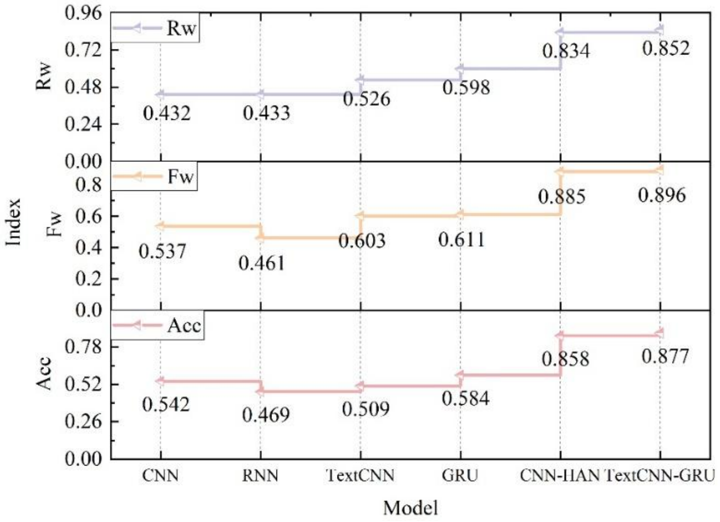


Figure 8: The data cluster is 2000

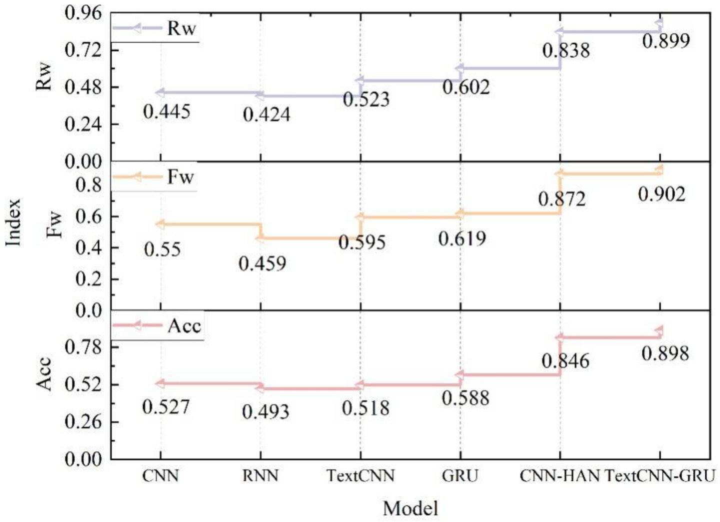


Figure 9: The data cluster is 3000

The above experiments demonstrate:

(1) With limited data (1000 training data experiment group), TextCNN significantly outperforms GRU, and even traditional CNN methods prove superior to GRU.

(2) GRU performance improves as data volume increases, but the 3000 training data experiment group shows no substantial improvement over the full dataset GRU training results, indicating GRU algorithm performance reaches a plateau with increasing data volume.

(3) When compared to CNN-HAN with limited data, TextCNN-GRU shows only marginal improvement initially, but this advantage grows as the dataset size increases.

(4) TextCNN-GRU consistently outperforms all other methods in experiments, validating the algorithm's effectiveness.

4.6 Misclassification Analysis

To thoroughly investigate the bottlenecks and optimization directions of the model in legal text classification tasks, quantitative metrics were further introduced for misclassification analysis. By randomly stratified sampling of 500 representative legal text samples, a confusion matrix was constructed to analyze discrepancies between actual labels and predicted results. This systematically evaluated the classification accuracy and confusion distribution of the TextCNN-GRU model across various crime categories, with the confusion matrix shown in Figure 10. The 500 legal texts encompassed 10 major categories of offenses: crimes endangering national security, crimes endangering public safety, crimes disrupting economic order, crimes infringing upon personal rights, crimes infringing upon property rights, crimes obstructing social management order, crimes endangering national defense interests, crimes of corruption and bribery, crimes of dereliction of duty, and crimes of military personnel violating their duties, denoted as C1 to C10 respectively. Through confusion matrix visualization analysis, the model exhibits significant performance variations across certain crime categories. It demonstrates higher recognition accuracy for high-frequency offenses such as “Crimes Against Personal Rights (C4),” “Crimes Against Property (C5),” and “Crimes Against Social Management Order (C6).” However, its accuracy for “Corruption and Bribery (C8)” is notably lower, with 40 and 28 cases misclassified as “Dereliction of Duty (C9)” and “Disruption of Economic Order (C3)” respectively. This reflects the model's limited ability to handle crimes with ambiguous boundaries. Additionally, significant semantic overlap exists between “Crimes Endangering National Defense Interests (C7)” and “Crimes Endangering National Security (C1)”, resulting in 23 instances of confusion.

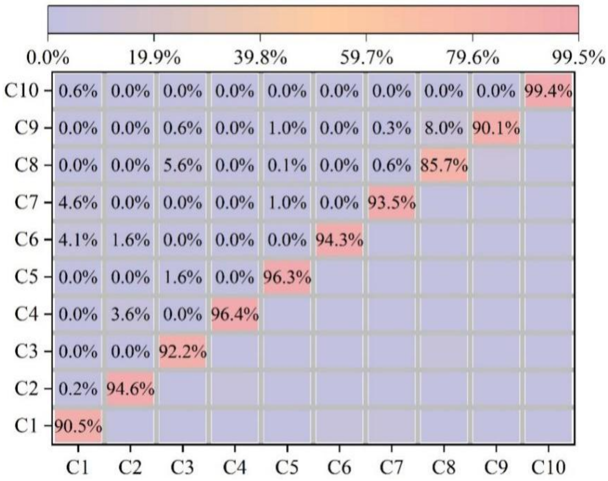


Figure 10: Confusion matrix

Based on the textual characteristics of misclassified samples, three typical misclassification patterns were identified:

(1) Blurred category boundaries. Analysis revealed that some crime texts exhibit unclear category boundaries, making accurate classification challenging for the model.

(2) Non-standard textual expressions and interference from obscure terms. Certain texts contain rare vocabulary and unstructured expressions, preventing word vectors from effectively representing their semantic meanings.

(3) Mixed narratives involving multiple events and roles. Criminal charge texts may describe multiple behaviors or roles, causing the model to overlook primary actions.

In summary, classification errors primarily stem from unclear category boundaries, ambiguous semantics, and non-standard expressions. Future improvements should focus on optimizing the dataset and adding secondary classifications to enhance the model's accuracy and practicality.

5 Conclusions and Outlook

This paper constructs a classification model integrating TextCNN and GRU for legal text classification tasks. Research results indicate that the proposed TextCNN-GRU model outperforms traditional deep learning models in terms of accuracy, macro-precision, macro-recall, and macro F_t scores. Compared to the high-performing CNN-HAN ensemble model, it achieves a 10% higher classification accuracy. However, the model exhibits certain limitations, including misclassifications when handling ambiguous category boundaries, non-standard textual expressions, and mixed narratives involving multiple events.

Future work will focus on further optimizing the dataset by enhancing data diversity and annotation granularity to improve the model's ability to distinguish ambiguous categories. Additionally, incorporating more domain knowledge and secondary classification schemes will strengthen the model's semantic understanding and application adaptability. Exploring more efficient algorithms and strategies will also enhance the model's real-time response capability in low-computing-power environments, promoting its widespread application in legal text classification tasks.

About the Author

Kunchi Wang was born in Zhanjiang, Guangdong, China P.R. China, in 1990. He received the Master degree from Guangdong University of Finance & Economics, P.R. China. Now, he is a doctor student Studies in Xi’an Jiaotong university. His research interests include Jurisprudence, Legisprudence.

Enchang Liu was born in Yanji, Jilin, China P.R. China, in 1996. She is a Master's student Studies inEast China University of Political Science and Law, P.R. China. Her research interests include Jurisprudence, Intellectual Property Law, Finance, Accounting.

Yuchen Meng was born in Xi’an, Shaanxi, P.R. China, in 1992. He holds an LLM from Gansu University of Politics and Law. Now, he is a Ph.D. candidate in computational law, with particular research interests in the jurisprudence of artificial intelligence and algorithmic power in justice.

References

[1] Winkels, R. (2015, February). The openlaws project: Big open legal data. In Proceedings of the International Legal Informatics Symposium (IRIS 2015) (pp. 189-196).

[2] Custers, B., & Leeuw, F. L. (2017). Legal Big Data (Legal Big Data). Custers BHM & Leeuw F.(2017), Legal big data: Toepassingen voor de rechtspraktijk en juridisch onderzoek, Nederlands Juristenblad, 34, 2449-2456.

[3] Filtz, E., Kirrane, S., & Polleres, A. (2021). The linked legal data landscape: linking legal data across different countries. Artificial Intelligence and Law, 29(4), 485-539.

[4] Grady, K. A. (2017). Mining legal data: Collecting and analyzing 21st Century gold. J. INTERNET L., 20, 1-12.

[5] Nishikawa-Pacher, A., & Hamann, H. (2023). Is every law for everyone? Assessing access to national legislation through official legal databases around the world. Oxford Journal of Legal Studies, 43(2), 298-321.

[6] Xu, C., & Pei, Z. B. (2021). Research on the construction of an intelligent court in the era of big data. In Computational Social Science (pp. 3-10). CRC Press.

[7] Gupta, A., Wang, A. Z., Lin, K., Hong, H., Sun, H., Liebman, B. L., ... & Roberts, M. E. (2017). Toward building a legal knowledge-base of Chinese judicial documents for large-scale analytics. In Legal knowledge and information systems (pp. 135-144). IOS Press.

[8] Liebman, B. L., Roberts, M. E., Stern, R. E., & Wang, A. Z. (2020). Mass digitization of Chinese court decisions: How to use text as data in the field of Chinese law. Journal of Law and Courts, 8(2), 177-201.

[9] Wang, Z., Wang, B., Duan, X., Wu, D., Wang, S., Hu, G., & Liu, T. (2019, November). IFlyLegal: a Chinese legal system for consultation, law searching, and document analysis. In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp): System demonstrations (pp. 97-102).

[10] Olkiewicz, M., & Cabaj, M. (2022). The problem of burnout in the judiciary. Zeszyty Naukowe. Organizacja i Zarządzanie/Politechnika Śląska, (155), 413-430.

[11] Opeskin, B. (2017). The supply of judicial labour: Optimising a scarce resource in Australia. Oñati Socio-Legal Series, 7(4).

[12] Schrever, C., Hulbert, C., & Sourdin, T. (2024). The privilege and the pressure: Judges' and magistrates' reflections on the sources and impacts of stress in judicial work. Psychiatry, Psychology and Law, 31(3), 327-380.

[13] Geyh, C. G. (2025). The Pressure Points of Professional Identity for Judges in the Modern Era. Mercer Law Review, 76(3), 16.

[14] Kelechi, A. H., Alsharif, M. H., Bameyi, O. J., Ezra, P. J., Joseph, I. K., Atayero, A. A., ... & Hong, J. (2020). Artificial intelligence: An energy efficiency tool for enhanced high

performance computing. Symmetry, 12(6), 1029.

[15] Liu, J., Qian, Y., Yang, Y., & Yang, Z. (2022). Can artificial intelligence improve the energy efficiency of manufacturing companies? Evidence from China. International Journal of Environmental Research and Public Health, 19(4), 2091.

[16] Qudus, L. (2025). Leveraging Artificial Intelligence to Enhance Process Control and Improve Efficiency in Manufacturing Industries. International Journal of Computer Applications Technology and Research, 14(02), 18-38.

[17] Waltersmann, L., Kiemel, S., Stuhlsatz, J., Sauer, A., & Miehe, R. (2021). Artificial intelligence applications for increasing resource efficiency in manufacturing companies—a comprehensive review. Sustainability, 13(12), 6689.

[18] Martins, V. S., & Silva, C. D. (2022). Text classification in law area: a systematic review. Journal of Information and Data Management, 13(6).

[19] Costa, J. A. F., Dantas, N. C. D., & Silva, E. D. S. (2023, November). Evaluating Text Classification in the Legal Domain Using BERT Embeddings. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 51-63). Cham: Springer Nature Switzerland.

[20] Rigutini, L., Globo, A., Stefanelli, M., Zugarini, A., Gultekin, S., & Ernandes, M. (2024). Performance, energy consumption and costs: a comparative analysis of automatic text classification approaches in the Legal domain. INTERNATIONAL JOURNAL ON NATURAL LANGUAGE COMPUTING, 13(1), 19-35.

[21] McCarroll, N., McShane, P., O'Connell, E., Curran, K., Singh, M., McNamee, E., ... & Brammer, A. (2025). Evaluating Shallow and Deep Learning Strategies for Legal Text Classification of Clauses in Non-Disclosure Agreements. SN Computer Science, 6(7), 784.

[22] Coelho, G. M., Ramos, A. C., de Sousa, J., Cavaliere, M., de Lima, M. J., Mangeth, A., ... & Casanova, M. A. (2022). Text Classification in the Brazilian Legal Domain. In ICEIS (1) (pp. 355-363).

[23] Chalkidis, I., & Kampas, D. (2019). Deep learning in law: early adaptation and legal word embeddings trained on large corpora. Artificial Intelligence and Law, 27(2), 171-198.

[24] Clavié, B., & Alphonsus, M. (2021). The unreasonable effectiveness of the baseline: discussing SVMs in legal text classification. In Legal Knowledge and Information Systems (pp. 58-61). IOS Press.

[25] Csányi, G., & Orosz, T. (2022). Comparison of data augmentation methods for legal document classification. Acta Technica Jaurinensis, 15(1), 15-21.

[26] Wei, F., Qin, H., Ye, S., & Zhao, H. (2018, December). Empirical study of deep learning for text classification in legal document review. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 3317-3320). IEEE.

[27] Li, P., Zhao, F., Li, Y., & Zhu, Z. (2018, June). Law text classification using semi-

supervised convolutional neural networks. In 2018 Chinese control and decision conference (CCDC) (pp. 309-313). IEEE.

[28] Undavia, S., Meyers, A., & Ortega, J. E. (2018, September). A comparative study of classifying legal documents with neural networks. In 2018 Federated conference on computer science and information systems (FedCSIS) (pp. 515-522). IEEE.

[29] Bansal, N., Sharma, A., & Singh, R. K. (2019). An Evolving Hybrid Deep Learning Framework for Legal Document Classification. Ingénierie des Systèmes d'Information, 24(4).

[30] Lee, J., & Lee, H. (2019, October). A Comparison Study on Legal Document Classification Using Deep Neural Networks. In 2019 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 926-928). IEEE.

[31] Qiu, M., Zhang, Y., Ma, T., Wu, Q., & Jin, F. (2020). Convolutional-neural-network-based Multilabel Text Classification for Automatic Discrimination of Legal Documents. Sensors & Materials, 32.

[32] Hammami, E., Faiz, R., & Akermi, I. (2021, June). A dynamic convolutional neural network approach for legal text classification. In International Conference on Information and Knowledge Systems (pp. 71-84). Cham: Springer International Publishing.

[33] Han, Q., & Snaidauf, D. (2021, December). Comparison of deep learning technologies in legal document classification. In 2021 IEEE international conference on big data (big data) (pp. 2701-2704). IEEE.

[34] Akça, O., Bayrak, G., Issifu, A. M., & Ganiz, M. C. (2022, August). Traditional machine learning and deep learning-based text classification for turkish law documents using transformers and domain adaptation. In 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) (pp. 1-6). IEEE.

[35] Chen, H., Wu, L., Chen, J., Lu, W., & Ding, J. (2022). A comparative study of automated legal text classification using random forests and deep learning. Information Processing & Management, 59(2), 102798.

[36] Song, D., Vold, A., Madan, K., & Schilder, F. (2022). Multi-label legal document classification: A deep learning-based approach with label-attention and domain-specific pre-training. Information Systems, 106, 101718.

[37] Shukla, B., Gupta, S., Yadav, A. K., & Yadav, D. (2023, September). Challenges and issues in legal documents classification. In AIP Conference Proceedings (Vol. 2754, No. 1, p. 020001). AIP Publishing LLC.

[38] Rusiya, S., & Jamatia, A. (2023, May). Implementation of legal documents text summarization and classification by applying neural network techniques. In Machine Intelligence Techniques for Data Analysis and Signal Processing: Proceedings of the 4th International Conference MISP 2022, Volume 1 (pp. 329-341). Singapore: Springer Nature Singapore.

[39] Du, Z. (2023). Research on the Text Classification of Legal Consultation Based on Deep

Learning. International Journal of Advanced Computer Science and Applications, 14(7).

[40] Johnson S. Joshua, Murty M. Ramakrishna & Navakanth I. (2023). A detailed review on word embedding techniques with emphasis on word2vec. Multimedia Tools and Applications,83(13),37979-38007.

[41] Bao Guo, Chunxia Zhang, Junmin Liu & Xiaoyi Ma. (2019). Improving text classification with weighted word embeddings via a multi-channel TextCNN model. Neurocomputing, 363,366-374.

[42] Poluru Eswaraiah & Hussain Syed. (2024). A Hybrid Deep Learning GRU based Approach for Text Classification using Word Embedding. EAI Endorsed Transactions on Internet of Things,10(1),

[43] Pillai Aditya R & Arun Biri. (2024). A feature fusion and detection approach using deep learning for sentimental analysis and offensive text detection from code-mix Malayalam language. Biomedical Signal Processing and Control,89,