International Journal for Housing Science and Its Applications
Publish August 3, 2025. Volume 46, Issue 3 Pages 29-43

https://doi.org/10.70517/ijhsa46303

Research on Intelligent Legal Sentencing Assisted Decision Making Model Combining Data Mining Algorithms and Legal Theory

Yang Yuan[1,*]
[1] Henan University of Economics and Law, Zhengzhou, Henan, 450003, China
Corresponding authors: (e-mail: ttl20250305@163.com).

Abstract With the rapid development of emerging technologies, legal text datatization provides the possibility of intelligent legal judgment assisted decision making. In this paper, a multi-task judgment prediction model based on data mining algorithm and Lawformer is designed to introduce Lawformer pre-training model, incorporate legal theories, and utilize HAN encoder to encode the case text at the word level and sentence level in order to better capture the semantic information. Then the support vector machine algorithm is integrated into the model to explore the crime composition, and a multi-task learning framework is constructed for the intelligent legal judgment assisted decision-making model. The experimental results show that the model in this paper performs well on the CAIL2018-Small public dataset, which significantly improves the model's legal judgment prediction accuracy and interpretability, with micro-averaged F1 values of 87.3 and 85.4 on the two tasks of crime prediction and law recommendation, and the overall legal case prediction correctness is as high as 99.48%. The research in this paper provides a new working path for the field of intelligent justice, which contributes to the scientific and intelligent development of legal judgment assisted decision-making.

Index Terms Lawformer, Support Vector Machine Algorithm, HAN Encoder, Multi-task Learning, Legal Judgment

I. Introduction

As the idea of "governing the country by law" has taken root in people's minds, they are more inclined to seek legal channels when disputes arise. In 2023, the number of litigation cases in China reached 45.574 million, with an average of about 23,000 new judicial documents added daily. By 2024, the average number of new judicial documents added daily will be about 26,000, and the number of judges with quota will be approximately 127,000. There are approximately 750,000 practitioners in the legal profession, and the judicial trial field shows a distinct phenomenon of "more cases than personnel" [1]-[3]. Litigation cases include civil, administrative, criminal, state compensation and other types of cases. However, judges hear different cases based on their nature, and their judicial authority is clearly demarcated, lacking universality. Moreover, in China, a judge needs to undertake a large amount of basic legal affairs work, such as litigation guidance, legal publicity, investigation and research, and answering questions after judgment, which further leads to the phenomenon of "more cases than fewer" [4]-[7]. The phenomenon of "many cases but few people" has brought about many problems. The first issue is the doubling of the pressure on judges in handling cases, as well as the balance between the efficiency and quality of case trials. Secondly, in the face of special cases, the written legal provisions may not be highly compatible, and judges need to make judgments based on a comprehensive review of the entire case and references from historical cases. Furthermore, there are differences in judicial trial standards in different regions, and different penalty results occur in the same type of cases [8]-[10]. Furthermore, due to the advanced development of the Internet, many cases have drawn public attention. While judges are under pressure in terms of the number of cases, they are also confronted with psychological pressure.

In recent years, the state has included judicial trial intelligence in the government's strategy for the development of artificial intelligence due to the fact that the judicial industry is characterized by high professionalism, large amounts of data and information, and strong internal logic [11]. Courts at all levels hope that through the further integration of artificial intelligence and justice, in the future to provide more professional and convenient legal aid services, can effectively reduce the basic transactional work of judicial personnel, improve the utilization rate of human resources in the court, so that the judges are more focused on the core work, better solve the difficult cases, to achieve the “burden” of the judges! To achieve the purpose of “reducing the burden” of judges [12]-[15]. In addition,

Bagaric et al. [16] mentioned the usefulness of intelligent algorithms in the criminal justice system, including from sentencing decisions to prison management, because the algorithms are detached from human emotions, avoiding subjective bias in decision-making. Currently judicial smart trial is an urgent and challenging scenario for AI application. Whereas sentencing decisions are the primary responsibility of judges in judicial trials, AI technology gives judges assistance in a variety of forms.

Artificial intelligence based legal aids such as case retrieval, judgment prediction, document review, and intelligent decision making are used in the practical aspects of the judicial field.Sil et al [17] introduced legal aids that automate legal models with the application of machine learning algorithms to help legal practitioners in text recognition and summary construction with 94% accuracy. Radhika et al [18] optimized a natural language processing model for legal document analysis using multilingual embedding, domain-specific knowledge, transfer learning, and fine-tuning strategies for legal document analysis in a multilingual environment. Wei [19] proposed a model for automatic generation of legal documents using knowledge graph technology, which not only shows advantages in the efficiency and quality of generation, but also improves the completeness of the generated content. Zhang et al [20] combined convolutional neural networks and gated recurrent unit neural networks to construct an outcome model for legal recommendation work in smart courts. Liu et al [21] used sparse and dense retrieval models to extract and filter unlabeled data for massive legal cases respectively, and constructed pseudo-labeled data by combining the labeled data, which improved the accuracy of retrieval and was a good method to save time for retrieval of historical cases for those involved in the judicial field. Zhang and Dou [22] constructed a legal judgment prediction model for past cases, which extracts the relevant features of the target case by searching, parsing, and learning from past cases, makes predictions, and assists the judge to make judgments, which reduces the judge's collection time for the relevant cases. While Shang [23] considered the litigation case process, feature capturing and feature dimensionality reduction of litigation process monitoring data with convolutional neural network and principal component analysis, respectively, so as to construct a judgment prediction model and optimize the model with genetic algorithm. Therefore, the research on judicial intelligence not only has significant theoretical research significance, but also has great practical demand.

In addition, in judicial decision making, Ulenaers [24] mentioned that AI assists trial decision making with assistants and intelligent judges, where the intelligent judge is making autonomous decisions in a fully automated court trial, while the assistants help the judge in trial decision making only in an auxiliary nature. Guo [25] used a bidirectional long and short-term memory network to construct a monitoring model for monitoring online public opinion in a judicial intelligent decision support system, which incorporated the intensity of popular sentiment, the time-series characteristics of the event, and the semantic similarity of the network into the monitoring considerations. Zhang et al [26] developed an intelligently-assisted legal discretionary system, which is supported by genetic algorithms and back-propagation neural networks, and can be used to predict the amount of penalties and damages in legal case judgments. Lian and Yang [27] designed a generative intelligent judicial trial assistance system through recurrent neural networks and long and short-term memory networks, which aided decision-making in judicial trials with the processes of case entry, prediction of judgment, and generation of legal documents. Ma [28] established an intelligent assisted legal judgment decision making method with the help of deep neural networks, which is carried out through three stages: crime prediction, relevant law recommendation, and judgment prediction, which contains a multi-task judgment prediction model and a sentence interval prediction model. Ng et al [29] mentioned that although artificial intelligence technology has improved the efficiency and quality of judicial case decisions, more consideration needs to be given to the transparency, privacy, adoptability, responsibility and obligation of the case in order to better ensure judicial justice. Moreover, most of the intelligent aids do not consider the legal principles, and there is data bias in the program setting, which leads to inconsistency between the intelligent fairness and the actual fairness, thus affecting the legal fairness. Therefore, there is a need to incorporate relevant legal principles and superior technology in intelligent legal judgment aid decision-making. Legal theory is the study of the connotation and extension of law, the origin and development of law, the function and effectiveness of law, the order and value of law and other issues, which is a powerful representation as the input of legal principles into intelligent decision-making. And data mining algorithms are the process of extracting hidden or potentially useful information and knowledge from a large amount of incomplete, noisy, fuzzy and random data, which is an effective tool to assist in solving the problem of transparency, reducing data bias and so on [30].

The study proposes a Lawformer-based multi-task judgment-assisted decision-making model, MJP-Law. The model consists of three modules: case description encoding, relevant crime composition encoding, and multi-task learning. The case description encoding module uses Lawformer to generate dynamic word vectors containing semantic information of legal cases, and encodes the word vectors using HAN encoder to obtain the vector representation of legal sentences. In the relevant crime composition coding module, the support vector machine is utilized to obtain the charges related to the case, and then it is coded to embed them into the model. Finally, the

multi-task learning module is utilized to complete several legal case decision-making tasks such as crime prediction, law sentence recommendation and sentence prediction. The performance of the model in this paper is verified on the CAIL2018-Small public dataset, and case studies are utilized to verify the model's actual assisted decision-making effect.

II. Definition and methodology of multitasking legal judgment assistance decision-making

II. A.Problem definition

The task of legal sentence prediction is similar to a text classification problem, which aims to predict legal sentence outcomes by analyzing the input legal text. The input of a general LJP consists of three groups: case descriptions, legal text concepts, and penalty descriptions. And the output of LJP also consists of three groups: legal text, offense and penalty clause prediction. In this paper, the task of LJP can be formalized as $F(S,C,R)=(\hat{y}_1,\hat{y}_2,\hat{y}_3)$, where S, C, and R stand for the case descriptions, the keywords of the law text concepts, and the keywords of the case penalty description, respectively. The goal of this paper is to train a model $F(\cdot)$ to predict the sentence outcome $\hat{y}_1$ for the applicable legal provision $\hat{y}_1$, the offense $\hat{y}_2$, and the sentence $\hat{y}_3$.

II. A. 1) Description of cases

A case description is a case description of a case, including when the crime was committed, where the crime was committed, the outcome of the crime, the instrumentalities of the crime, and the sentence handed down by the court or judge.

In this paper, we use Jieba to disambiguate case descriptions and remove deactivated words. As a result, each case description can be described as a sequence of words, i.e., $S=\{s_1,s_2,...,s_n\}$, where n is the number of words in the case description and s_i denotes the i th word from the word sequence.

II. A. 2) The concept of a law

A legal concept is a description of the concept of an offense in an existing legal provision. The goal of this paper is to extract keywords from the legal concepts and integrate them into multitask prediction. According to the research, there are no existing keywords of law concepts, so this paper grabs the legal articles from China Find Law website according to the offense labels. For each offense, this paper takes only the first 10 words generated by TF-IDF to construct the keywords of legal concepts. If the number of keywords is less than 10 words, this paper uses 0 to fill the keyword list. Therefore, this paper can get the keywords of the concept of the law article $C=\{c_1,c_2,...,c_{|C|}\}$, where $c_i(1\le i\le|C|)$ and $|C|$ denote the i th keyword and the number of keywords extracted from the concept of law, respectively.

II. A. 3) Description of penalties

Penalty descriptions are descriptions of sentences in existing legal texts that deprive criminals of their personal freedom. Similarly, for all crimes, this paper extracts the keywords of penalty descriptions from the legal texts crawled on the web using regular expressions in the python library and notated as $R=\{r_1,r_2,...,r_{|R|}\}$, where $r_i(1\le i\le|R|)$ and $|R|$ denote the i th keyword and the number of extracted keywords, respectively.

II. B.Related Principles

II. B. 1) Chinese Segmentation

Chinese participle is the first step in natural language processing, Chinese participle is different from English participle only need to press space can be participle, and due to the complexity of Chinese itself, Chinese participle is also much more complex than English, much more difficult. There are several commonly used lexical tools: Jieba, Thulac, Pynlpir, SnowNLP, PyLTP, CoreNLP, among which Jieba is chosen as the lexer in this paper. Since Jieba can customize word lists, custom word lists can be added to increase the accuracy of segmentation in the scenarios of legal verdict prediction.Jieba can segment Chinese words, and customized words can also be added.

II. B. 2) Word vectors

The notation for representing natural language words in machine learning can be categorized into One-hot representation [31] and Distributed representation.One-hot representation represents each word as a long vector whose dimension is the size of the word list. Only one dimension of the vector has a value of 1, the other dimensions have a value of 0, and the dimension with a value of 1 represents the current word. Since there are two problems with the One-hot representation, one is that the data dimension will be too large when the text richness is large, and

the second is because a lexical divide will be created. Therefore distributed representation is usually used in computers. By representing words in a distributed way, it can effectively solve the lexical divide problem.

II. B. 3) Word2Vec

Distributed representations of words can be obtained by training the language model Word2Vec [32]. And Word2vec contains two models: the CBOW and Skip-gram [33] models. The CBOW model predicts the center word by the surrounding words. According to the prediction result of the center word, the gradient descent method is used to adjust the vectors of the surrounding words continuously. At the end of training, each word is used as the center word and the word vectors of the surrounding words are adjusted to finally get the word vectors of all the words in the whole text. In Skip-gram model, the center word will be used to predict the surrounding words and based on the prediction of the surrounding words.Skip-gram uses gradient descent method to adjust the word vectors of the center word continuously. After traversing all the text, the word vectors of all the words in the text are finally obtained.

II. B. 4) TF-IDF

Word Frequency-Inverse Document Frequency (TF-IDF) is a commonly used weighting technique in information retrieval and information exploration.TF-IDF is a statistical method for evaluating the meaning of a word in a document set or corpus. The meaning of a word increases proportionally with the frequency of its occurrence in a document, but decreases inversely with the frequency of its occurrence in a corpus. This number is usually normalized (usually word frequency divided by the total number of words in the article) to prevent it from being distorted in excessively long documents. High-frequency words in a particular document and low-frequency words in the entire document set can produce a highly weighted TF-IDF.Therefore, the TF-IDF tends to filter out commonly used words and retain important words.The TF-IDF is calculated as shown below. Namely:

$$TF_w = \frac{\text{The number of times the term t appears in a class}}{\text{The number of all terms in the class}} \tag{1}$$

$$IDF = \log\left(\frac{\text{The total number of documents in the corpus}}{\text{Number of documents containing term t} + 1}\right), \tag{2}$$

$$TF - IDF = TF * IDF \tag{3}$$

II. C.Related tools

II. C. 1) Python Language and Related Packages

Python is a language that represents the idea of simplicity.Python is very easy to use and the syntax is very simple. If you write a program in Python language, you don't need to think about the underlying details and it has good portability. In addition, Python has good interpretability and is object-oriented programming.Python also has a large and rich library and excellent extensibility.

II. C. 2) TensorFlow Framework

Tensorflow [34] is an open source deep learning framework provided by Google. Today, Tensorflow is widely used by many companies and startups to automate work tasks and develop new systems. It is highly praised for its support of distributed training, scalable production and deployment options, and support for multiple devices.

Tensorflow uses data flow models to describe computational processes and map them to various hardware platforms. Thanks to a unified architecture, Tensorflow can be deployed across multiple platforms, greatly reducing application delivery for machine learning systems.

III. Lawformer-based multitasking judgment prediction modeling

III. A. Lawformer word embeddings

Lawforme is a pre-training model based on Longformer. When using a general-purpose domain pre-training model to process legal text, it is necessary to truncate the text that exceeds the length of 512, an operation that will directly result in the loss of textual information, which will lead to unsatisfactory training results. Unlike mainstream pre-training models, Lawformer does not use the standard self-attention mechanism, it realizes the encoding of long legal texts by combining three different attention mechanisms. These three attention mechanisms are sliding window attention, extended sliding window attention, and global attention mechanism.

Given the case description text $d = \{s_1, s_2, ..., s_n\}$, n is the number of sentences, the i th sentence $s_i = \{w_{i1}, w_{i2}, ..., w_i\}$, w_{ij} denotes the j th word in the i th sentence, and t denotes the sentence length. The vector representation x_{ij} is obtained after w_i by Lawformer word embedding, which is computed as:

$$x_{ij} = Lawformer(w_{ij}) \tag{4}$$

III. B. HAN Encoder

The case description text belongs to long text, which contains multiple types of information elements such as the cause, passage, and result of the case with typical hierarchical structure, and the HAN model is utilized to capture the inter-sentence relationship of the text.The HAN encoder consists of word level coding module and sentence level coding module.

III. B. 1) Word-level coding

The word level coding module uses a model structure that combines BiGRU and Attention to obtain a vector representation of a sentence by coding a sequence of word levels.

The BiGRU model is able to encode from both forward and reverse directions to learn contextual information. The sentence $X_i = \{x_{i1}, x_{i2}, ..., x_{ii}\}$ as inputs to BiGRU, the hidden layer states $\vec{h}_{ij}, \overleftarrow{h}_{ij}$ are obtained after encoding in the two directions, respectively, and the outputs are obtained after splicing h_{ij}. The computational procedure is as follows:

$$\overrightarrow{h_{ij}} = \overrightarrow{GRU}(x_{ij}) \tag{5}$$

$$\overleftarrow{h_{ij}} = \overleftarrow{GRU}(x_{ij}) \tag{6}$$

$$h_{ij} = \left[\overrightarrow{h_{ij}}, \overleftarrow{h_{ij}}\right] \tag{7}$$

In order to find the words that contribute more to the semantic expression of the sentence, the attention mechanism is introduced for encoding. The attention mechanism assigns weights to each word in the sentence according to the degree of contribution, without affecting the core semantic expression of the sentence.

Specifically, the hidden representation u_{ij} is first obtained by a single-layer perceptron, and then the normalized exponential function is used to calculate the association similarity between the hidden representation u_{ij} and the vector representation u_w of the sentence context, which in turn obtains the corresponding probability weights α_{ij}. Finally, vector weighted summation is used to finally obtain the vector representation of the sentence S_i. The specific calculation process is as follows:

$$u_{ij} = \tanh(W_w h_{ij} + b_w) \tag{8}$$

$$\alpha_{ij} = \frac{\exp(u_{ij}^T u_w)}{\sum_j \exp(u_{ij}^T u_w)} \tag{9}$$

$$S_i = \sum_j a_{ij} h_{ij} \tag{10}$$

where W_w and b_w are the weight matrix and bias vector, respectively.

III. B. 2) Sentence-level coding

Similarly, the sentence level coding module also adopts the model structure combining BiGRU and Attention, where sentences form text paragraphs, and feature extraction of text paragraph information is accomplished by encoding sentence vectors S_i.

Firstly, the vector representation of sentence S_i is inputted into BiGRU, and the hidden layer states $\vec{L}_i, \overleftarrow{L}_i$ are obtained respectively after bidirectional encoding, and the two vectors are spliced to get the outputs L_i. The computational procedure is shown below:

$$\begin{aligned} \overrightarrow{L_i} &= \overrightarrow{GRU}(S_i) \\ \overleftarrow{L_i} &= \overleftarrow{GRU}(S_i) \end{aligned} \tag{11}$$

$$L_i = \left[\overrightarrow{L_i}, \overleftarrow{L_i}\right] \tag{12}$$

Firstly, the hidden representation u_i is obtained by a single-layer perceptron, and then the Softmax probability function is used to calculate the similarity of the association between the hidden representation u_i and the representation of the textual passage u_s to obtain the weights of the corresponding sentences α_i, and u_s represents the core semantics of the textual passage. It is obtained by random initialization, and the parameters are updated by backpropagation. Finally the vector representation h_i of each sentence and the corresponding weight α_i are weighted and summed to obtain the text passage representation P. The computational procedure is shown below:

$$u_i = \tanh(W_s L_i + b_s) \tag{13}$$

$$\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)} \tag{14}$$

$$P = \sum_i \alpha_i h_i \tag{15}$$

where b_s and W_s denote the bias vector and weight matrix of the single-layer perceptron, respectively.

III. C. Relevant Criminal Composition Codes

Crawl the corresponding criminal composition of the charges on China Find Law, go to the classification of the charges through the binary classification method, input the case description into the support vector machine classifier, according to the output of the probability size of each charge, to get the m charges that have relevance to the case, and find the corresponding criminal composition through these charges. The relevant crime composition C_i is embedded to get the word embedding representation $E_i = \{e_1, e_2, ..., e_k\}$. where $i \in [1, m]$ and k is the number of words. It is expressed by the following formula:

$$E_i = Lawformer(C_i) \tag{16}$$

On the one hand, the length of the crime composition text is shorter compared to the description of the case, and the information type is single, without text hierarchy. On the other hand, CNN adopts the weight sharing mechanism of convolutional layer to be able to capture the local features in the text, so as to learn the key information of the text. Therefore, CNN is used to learn the text features of relevant crime composition. The calculation formula is shown below:

$$h_i = W_m \otimes e_{j:j+h-1} + b_m, j \in [1, k-h+1] \tag{17}$$

Where " $\otimes$ " is the convolution operation, W_m and b_m denote the convolution matrix and the bias vector, respectively, and h is the size of the upper and lower sliding windows.

Since these m crime components are associated with the same case description, in order to capture the associated features with the case description, vector summation is used to obtain the associated crime component coding representation M of the case description. The formula is shown below:

$$M = \sum_m h_i \tag{18}$$

The vector representation P obtained at the case description coding end is further vector spliced with the vector representation M obtained at the crime composition coding end. This is represented by the following equation:

$$Q = [P, M] \tag{19}$$

III. D. Multi-task learning module

In order to simulate the judge's trial thinking, multiple fully connected layers are used to construct the relationship between the three subtasks.

In predicting the charge, the splicing vector Q is obtained as f_1 by multilayer perceptron, f_1 is used as a specific representation of the charge prediction task, and the fully connected layers are used as classifiers, and the output is transformed into the prediction probability of each category of the charge using the sigmoid activation function. The formula is calculated as follows:

$$\hat{y}_1 = sigmoid(W_1 f_1 + b_1) \tag{20}$$

where b_1 and W_1 denote the bias vector and weight matrix of the classifier for the offense prediction task, respectively, and $\hat{y}_1$ denotes the prediction probability of each category of the offense.

Similarly, in predicting the statute, the splicing vector Q is passed through a multilayer perceptron to obtain f_2, and f_2 is used as a specific representation of the statute prediction task, and the output is transformed into the predicted probability of each category of the statute using the fully-connected layer as the classifier and the sigmoid activation function. The computational formula is shown:

$$\hat{y}_2 = sigmoid(W_2 f_2 + b_2) \tag{21}$$

where b_2 and W_2 denote the bias vector and weight matrix of the classifier for the French bar prediction task, respectively. $\hat{y}_2$ denotes the prediction probability of each category of the French bar.

In the prediction of sentence, the splicing vector Q is obtained by multilayer perceptron to obtain f_3, f_1, f_2 and f_3 are spliced, and the splicing is used as a specific representation of the sentence prediction task. The calculation formula is as follows:

$$\hat{y}_3 = sigmoid(W_3 \cdot (f_1, f_2, f_3) + b_3) \tag{22}$$

Where W_3 and b_3 are the weight matrix and bias vector of the classifier for the sentence prediction task, respectively, and $\hat{y}_3$ denotes the predicted probability of each category of the sentence.

For each sub-task, the loss is calculated separately here using the binary cross entropy loss function, assigning weights to the loss of each task, and finally the total loss is calculated using weighted summation. The calculation formula is as follows:

$$Loss = -\sum_{i=1}^{3} \lambda_i \sum_{j=1}^{|Y_i|} (y_{i,j} \log_2(\hat{y}_{i,j}) + (1 - y_{i,j}) \log_2(1 - \hat{y}_{i,j})) \tag{23}$$

where $y_{i,j}$ and $\hat{y}_{i,j}$ are the probabilities of one-hot coding and predictive labeling corresponding to the j th class of true labels for the i th task, respectively, and $|Y_i|, \lambda_i$ denote the total number of labels and the weight parameter for the i th task, respectively.

IV. Validation of the effectiveness of multi-task judgment prediction models

IV. A. Experimental data set

The dataset used for the experiments in this paper is the public dataset CAIL2018-Small from the Challenge of Artificial Intelligence in Justice (CAIL2018), which contains 154,592 pieces of training data, 17,131 pieces of validation data, and 32,508 pieces of test data, involving 202 charges and 183 articles. 183 legal articles. The cases in the dataset are stored in json format, which contains case descriptions, applicable legal provisions, charges and sentences, etc. Since the model in this paper aims to explore the dependency relationship between the charge prediction and law recommendation, the sentence prediction problem is not considered for the time being.

IV. B. Data pre-processing

The data are easily disturbed by noisy data, vacant data, and inconsistent data, resulting in data of varying quality. The quality of data directly affects the prediction and generalization ability of the model to a certain extent. Therefore, in order to improve the data quality, this paper does some data preprocessing work on the dataset. Data processing mainly includes five aspects: input preprocessing, labeling preprocessing, special influence factor processing, interference information processing and interference information processing.

IV. C. Benchmarking model and assessment indicators

In order to verify the prediction ability of the Lawformer-based multi-task decision prediction model, this subsection selects six benchmark models to compare the effects with the Lawformer-based multi-task decision prediction model, including the RNN-based model, the CNN-based model, the attention mechanism-based model, the model integrating legal information, and the model constructing the dependencies between subtasks, which are BiLSTM, CNN, DPCNN, HAN, Fact-Law and TopJudge.

For the multi-classification task of crime prediction and legal recommendation, this paper adopts macro-mean F_1 value and micro-mean F_1 value as the evaluation indexes of the model, in which the macro-mean F_1 value is more concerned with the categories with small amount of sample data, and the micro-mean F_1 value is more concerned with the categories with large amount of sample data.

Assuming that the number of case documents is N and the number of categories is C. Use y_{ij} to denote the model true value for labeling the j document with the i th category, and can only take the value of 0 or 1, and use $\overline{y}_{ij}$ to denote the corresponding model predicted value. Then we can get the true case (TP), false positive case (FP), false negative case (FN), and true negative case (TN) for the i category:

$$\begin{aligned} TP_i &= \sum_{j=1}^{N}\left[y_{ij}=1,\overline{y}_{ij}=1\right] \\ FP_i &= \sum_{j=1}^{N}\left[y_{ij}=0,y_{ij}=1\right] \\ FN_i &= \sum_{j=1}^{N}\left[y_{ij}=1,\overline{y}_{ij}=0\right] \\ TN_i &= \sum_{j=1}^{N}\left[y_{ij}=0,\overline{y}_{ij}=0\right] \end{aligned} \tag{24}$$

The macro F_1 -value is the sum of the F_1 -values of all categories, as follows:

$$\begin{aligned} P_i &= \frac{TP_i}{TP_i+FP_i} \\ R_i &= \frac{TP_i}{TP_i+FN_i} \\ F_i &= \frac{2\times P_i\times R_i}{P_i+R_i} \end{aligned} \tag{25}$$

$$F_{macro} = \frac{\sum_{i=1}^{C}F_i}{C}\times 100 \tag{26}$$

For the micro F_1 values, a global confusion matrix is constructed for the instances in the dataset, and then the micro precision and micro recall are computed, which in turn are computed to obtain the micro F_1 values as follows:

$$TP_{micro} = \sum_{i=1}^{C}TP_i \tag{27}$$

$$FP_{micro} = \sum_{i=1}^{C}FP_i \tag{28}$$

$$FN_{micro} = \sum_{i=1}^{C}FN_i \tag{29}$$

$$P_{micro} = \frac{TP_{micro}}{TP_{micro}+FP_{micro}} \tag{30}$$

$$R_{micro} = \frac{TP_{micro}}{TP_{micro}+FN_{micro}} \tag{31}$$

$$F_{micro} = \frac{2\times P_{micro}\times R_{micro}}{P_{micro}+R_{micro}}\times 100 \tag{32}$$

IV. D. Experimental results and analysis

This experiment is the construction of Lawformer-based multi-task judgment prediction model based on Anaconda development platform using Keras deep learning framework with TensorFlow as the backend.

Table 1 shows the experimental results of the model proposed in this paper and the other six comparative models on the CAIL2018-Small dataset on the 2 tasks of charge prediction and related lawformer prediction.

The performance of the model proposed in this paper is significantly better than all the other models on both tasks of offense prediction and statute recommendation, and the micro-averaged F1 values of this paper's model on the tasks of offense prediction and statute recommendation reach 87.3 and 85.4, respectively.From the performance comparison between this paper's model, the TopJudge model, and the other five benchmark models, it can be seen that the performance of this paper's model based on Lawformer and the TopJudge model have substantially improved evaluation metrics than single models on both the task of crime prediction and law recommendation, and the multi-task learning model has a stronger feature representation ability after utilizing the intrinsic linkage information between the crime and the related law, compared with the single model that only considers a single task. In addition, all models have higher micro-average F1 values than macro-average F1 values in the offense prediction and statute recommendation tasks, which also indicates that the distribution of offenses and related statutes in the CAIL2018-Small dataset is unbalanced.

Table 1: Different model performance compared with experimental results

Model	Crime prediction		Legal recommendation	
	Micromean F1	Macroid F1	Micromean F1	Macroid F1
BiLSTM	68.2	63.2	68.3	61.6
CNN	77.8	71.8	76.0	70.2
HAN	67.6	64.0	67.0	63.9
DPCNN	81.3	72.2	77.4	70.6
Fact-Law	76.1	69.7	73.5	69.2
Top Judge	83.7	74.1	79.9	73.7
This model	87.3	78.6	85.4	76.2

Parameter-sensitive experiments on the values of hyperparameters in the model are used to obtain reasonable hyperparameter values. The experiments are mainly divided into the following two groups: the first group of experiments is conducted on the dimension of word vector representation, and the second group of experiments is conducted on the number of relevant Lawformer bars extracted. Meanwhile, due to the complex network structure of the Lawformer-based model proposed in this paper, in order to avoid excessive computational costs, this paper selects the commonly used hyperparameter values to train the network model, and then selects the optimal parameters from them by comparing and analyzing the experimental results of the model on the tasks of charge prediction and law recommendation.

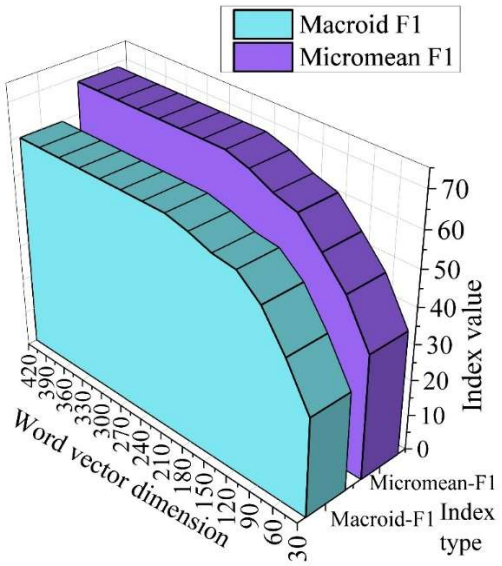

Figure 1: The influence of the word vector on the crime

The effects of word vector dimensions on the charge prediction and law recommendation tasks are shown in Figures 1 and 2, respectively. In the experiments, this paper fixes the other hyperparameters, and the word vector

dimensions are chosen as 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 390, and 420, respectively, and observes the corresponding models on the charge prediction and law recommendation tasks with micro average F1 value, and macro average F1 value. Figure 1 shows the impact of word vector dimension on the task of charge prediction and law recommendation. From the experimental results, it can be seen that the evaluation index has been rising sharply during the process of word vector dimension from 30 to 210, while the word vector dimension gradually tends to stabilize after 210, so taking into account the performance of the model and the difficulty of training, this paper chooses to set the word vector dimension to 210.

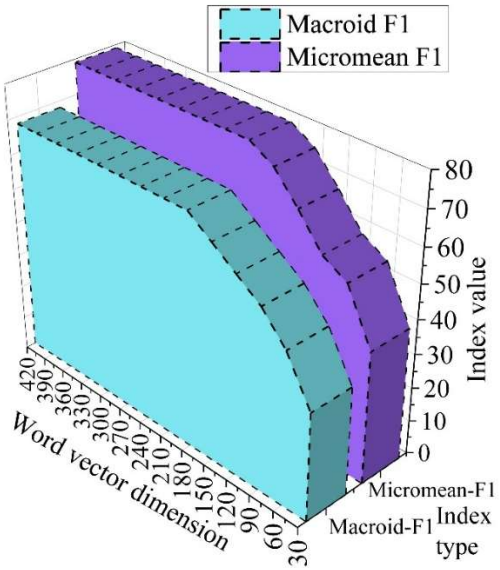

Figure 2: The influence of the word vector on the recommended task

Finally, this paper conducts parameter-sensitive experiments on the number of relevant legal strips extracted. The incorporation of relevant legal information can substantially improve the model performance, so the number of extracted relevant legal articles is also crucial for the model. In the experiments, this paper fixes other influential parameters, and the number of extracted relevant legal articles is set to 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20, respectively, and observes its impact on the tasks of offense prediction and legal article recommendation. Figures 3 and 4 demonstrate the experimental results. It can be seen that the evaluation metrics are in an increasing stage during the change of the number of relevant legal articles from 2 to 10, until they level off after 10. Therefore, in this paper, the number of extracted legal articles is chosen to be 10.

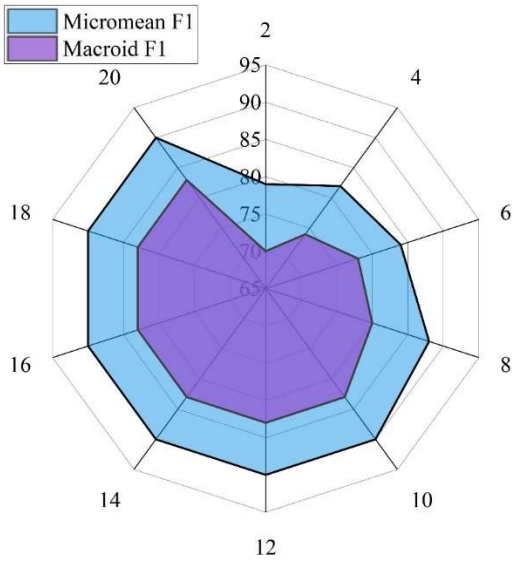

Figure 3: The effect of the relevant law on the prediction of the crime

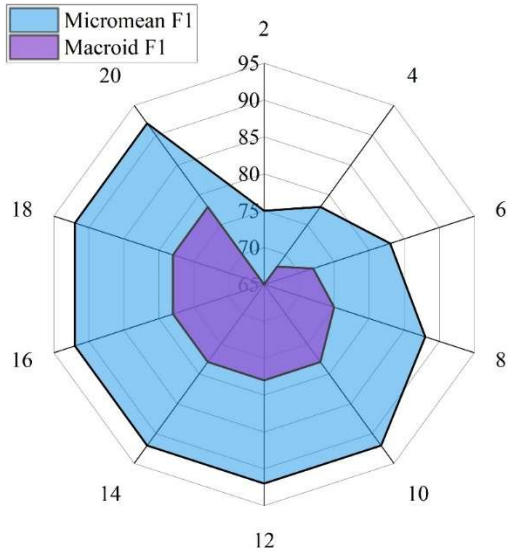

Figure 4: The effect of the method on the recommendation of the method

V. Interpretability analysis of predictions of legal judgments

This chapter will analyze the interpretability of the model in this paper from both macro and micro perspectives. On the micro level, some cases are selected to analyze the decision-making process of the model through the visualization of the attention heat map, to observe the key words on which the model bases its decision-making, and to explore whether these words are consistent with human logic; on the macro level, through the statistical analysis methods such as the error type analysis, and the lexicon of the charge attention value, we will discuss whether the knowledge learned by the model is consistent with the human cognition, and whether it is able to make judgments in accordance with correct decision-making logic, so as to validate the comprehensiveness and objectivity of the model interpretable method. Thus, the comprehensiveness and objectivity of the model's interpretable methods are verified.

V. A. Case Studies

The following shows an example of the model of this paper in the test set for the offense of tolerating drug use. The facts of the case are described as follows:

Ducheon County People's Procuratorate alleges that: on September 18, 2015, the defendant Zhu Shuangquan in Ducheon County, Dingcheng Office Airport New District XII of his rented room content to stay in Xiao Mou to take drugs, by the Ducheon County Public Security Bureau of the police on the spot, and the next day on his administrative detention for five days. in March 2016, the defendant Zhu Shuangquan in Ducheon County, Dingcheng Office Airport New District XII of his rented room to stay in Xiao Mou again. in May 2016, the defendant Zhu Shuangquan in the Ducheon County, Dingcheng Office Airport New District XII of his rented room again to stay in Xiao Mou to take drugs. In May 2016, Defendant Zhu Shuangquan again allowed Tian to take drugs in his rented room in the twelfth district of the new airport area of the Dingcheng Office in Ducheon County. Figure 5 shows the first level of conceptual labeling attention heat map, the case text, the higher the attention value obtained the darker the background color of the text. There are similarities and differences in the feature content of each level of attention attention, and the feature content of each level of attention attention basically conforms to the nested logical relationship of each label in the model design.

In the first level of conceptual labels, i.e., the first level of labels categorized by the object of infringement, the object of infringement was identified as the order of social administration through the act of “drug use” and “taking drugs”, which corresponds to the chapter crime of disrupting the order of social administration. In the model's prediction of what to focus on at this level of conceptual labeling, “drug use” receives the majority of the attention, and the same words appearing in different locations in the case description receive different intensities of attention. To some extent, this indicates that Lawformer, as a feature extraction component, is able to take the whole context into account better. The model gives more attention to the last two behaviors of “taking drugs again” and “taking drugs again” than the initial one.

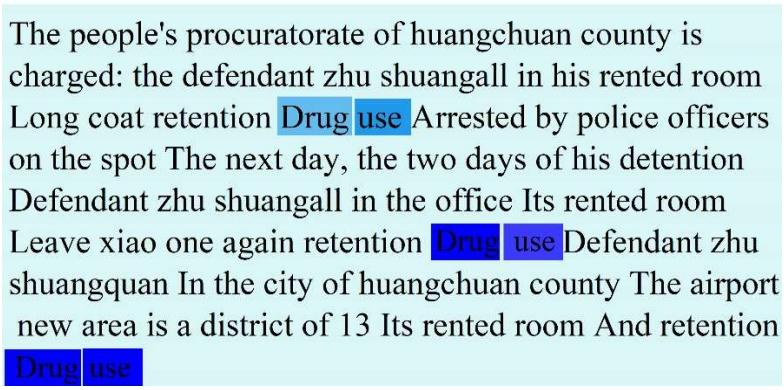

Figure 5: The first level concept tag

Figure 6 shows the second-level conceptual label attention heat map, in which the model still focuses mainly on "drug use" in the second-level conceptual label, i.e., the second-level labels classified by subjective and objective aspects. But at the same time, it also gives more attention to the three "stay" words than the first-level label. Through the acts of "drug use" and "drug consumption", it was determined that the state drug control order was obstructed, and the crime of smuggling, selling, transporting, and manufacturing drugs should be charged with the corresponding crimes.

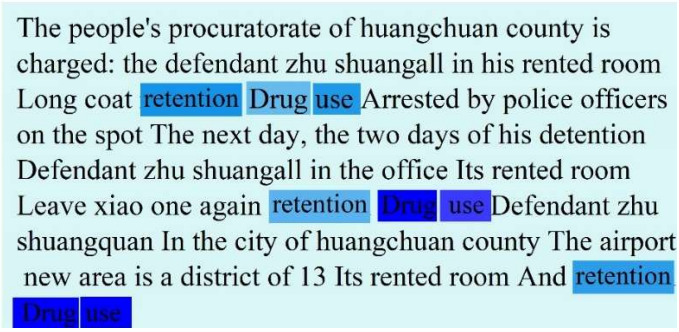

Figure 6: The second level concept tag

Figure 7 shows the attention heat map of the third-level conceptual labels, in which the third-level conceptual labels, that is, the classification of confusing crimes and the determination of the final crime level of the conceptual labels, the model still focuses on words such as "staying" and "taking drugs". However, compared with the second-level label, the focus on "drugs" has decreased significantly, while the attention to the word "containment" has increased significantly. This is because it has been confirmed to be a drug-related crime after the classification of the second-level label, and the main purpose of the third-level concept label in drug-related crimes is to distinguish between the crimes of smuggling, trafficking, transportation, and manufacturing drugs, and the crime of allowing others to take drugs and the crime of illegal possession of drugs. The key to the crime of accommodating others to take drugs lies in identifying the act of "accommodating".

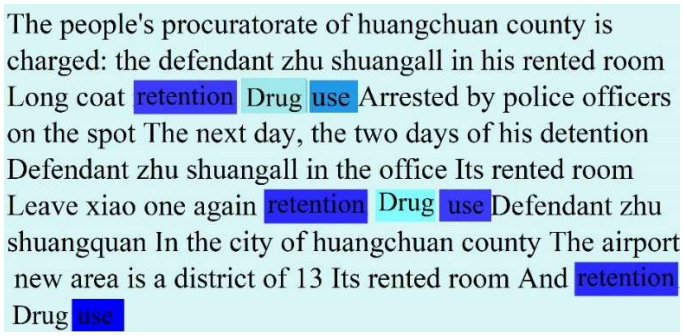

Figure 7: The third level concept tag

V. B. Overall analysis

In order to observe whether the labeling of the three-level hierarchical concepts in the Lawformer-based multitasking judgment prediction model proposed in this paper is effective and whether the model learns the knowledge of such hierarchical concepts, this paper intends to analyze the misclassification produced by the Lawformer-based multitasking judgment prediction incorporating the knowledge of the legal articles on the test set, and the prediction statistics are shown in Figure 8. There are a total of 5000 cases. Of these, the number of cases predicted incorrectly by the Lawformer-based multitasking judgment prediction totaled 26 cases. The correct rate of model charge prediction is 99.48%. Overall, the prediction effect of the model in this paper is good.

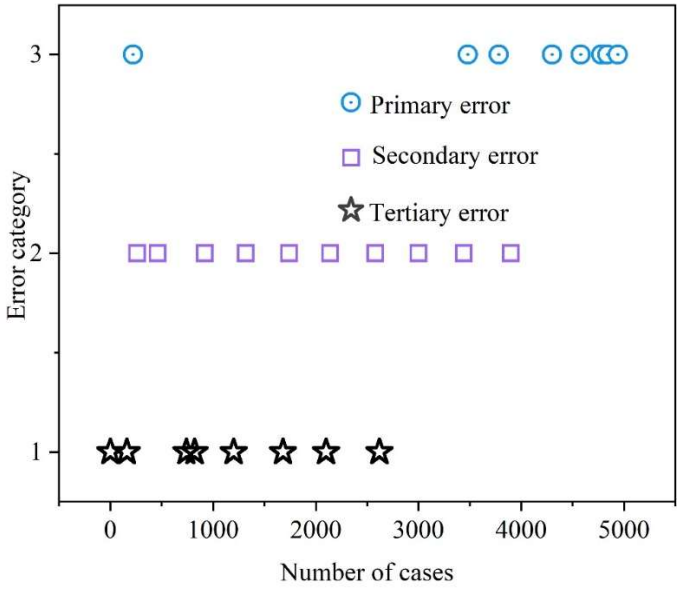

Figure 8: Misclassification analysis

Based on the data in the model training set, or the cases in the database, the words that are generally given higher attention by the model in the cases of this offense are shown in Table 2. However, these words do not necessarily appear in the content of the case entered by the user. From the macroscopic point of view of the entire dataset, sorted by the size of the mean value of attention, there is an inflection point on a particular crime, a level of conceptual labeling, but the inflection point location does not appear among the first few words, but rather the location of the first hundreds or even thousands of words, the number of words taken on the labeling of different crimes and levels is not a certain number of words.

In practical application, for specific charges and level labels, the specific number of words can be set according to the user's needs, in order to display the first n words with higher attention value. As the number of words with higher attention value corresponding to each crime is different, in order to facilitate the analysis, the words with higher attention value taken for each crime and level concept label are set to 5 here, and the words with duplicates in the concept labels of each level are removed and arranged in descending order of attention value.

Table 2: All kinds of charges are high in the different level of concept tags

Offence	Chapter offence/violation	Crime/objective aspect	Judgment crime/differentiation
	label1	label2	label3
Dangerous driving	Ethanol, transportation, driving, alcohol, blood	Content, wine, drink, drunk, traffic	Driving, identification, motor vehicles, accidents, motorcycles
Traffic accident	Traffic, accident, death, main, driving	Driving, responsibility, serious injury, collision, rescue	Injuries, charges, collisions, injuries, escape
Credit card fraud	Credit card, overdraft, bank, urge, arrears	Bank card, card number, zhang (card number), consumption, urge	Apply for, handle, renminbi, malicious, application
Contract fraud	Contract, sign, fraud, mortgage, performance	Fraud, fraud, fraud, fiction, collection	Money, give, lie, property, yuan

VI. Conclusion

This chapter proposes a new Lawformer-based multi-task judgment prediction model, MJP-Law, to deeply utilize the legal text corpus to assist legal judgments.

The model in this paper performs well on the two tasks of charge prediction and law recommendation, with micro-averaged F1 values of 87.3 and 85.4 on the two tasks, respectively, which illustrates its ability to allow the model to accurately capture key information and improve the performance of the prediction model. The values of the evaluation indexes of both the model in this paper and the TopJudge model are improved substantially compared with the single model, indicating that the multi-task learning model is able to deeply excavate the relevant information between the charges and the relevant legal articles, and improve the extraction ability of the features of the legal text.

The attention mechanism guides the model to increase attention to the key features of legal information, with different levels of conceptual labels paying different attention to the facts of the case, and different attention intensities for the same words in different positions, illustrating that Lawformer takes into account the full context in a more balanced way. Overall, the number of cases predicted incorrectly by the multi-task judgment prediction based on Lawformer is only 26 cases, and the prediction correctness rate is as high as 99.48%, which illustrates that the model in this paper shows excellent potential in the field of legal judgment prediction, and it has a more important application value.

Funding

This work was supported by Henan Philosophy and Social Sciences Planning Project: Research on Legal Regulation of the Digital Cultural Industry from the Perspective of Artificial Intelligence Technology (2024CFX007).

About the Author

Yang Yuan, doctor of law, from Henan University of Economics and Law, Zhengzhou, Henan, China.

References

[1] Duan, J., & Zhang, B. (2024). Abusive litigation in China: comparative insights and practical approaches for reform. Humanities and Social Sciences Communications, 11(1), 1-11.

[2] Li, H., Shao, Y., Wu, Y., Ai, Q., Ma, Y., & Liu, Y. (2024, July). Lecardv2: A large-scale chinese legal case retrieval dataset. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 2251-2260).

[3] Zhang, S., Luo, J., & Guo, P. (2025). The Legal Profession in China. In Technology, Legal Education and Legal Profession in China and Australia: Opportunities and Challenges (pp. 197-263). Singapore: Springer Nature Singapore.

[4] Chen, T., Xu, W., & Yu, X. (2024). Administrative litigation in China: Assessing the chief officials' appearance system. The China Quarterly, 259, 744-764.

[5] Shui, B. (2024). Chinese Constitutonal Performance Unveiled: Text Mining Insights in Civil Litigations. ICL Journal, 18(3), 429-456.

[6] Zhan, C., & Qiao, S. (2024). Workload, legal doctrine, and judicial review in an authoritarian regime: A study of expropriation judgments in China. International Review of Law and Economics, 80, 106232.

[7] Yam, J. (2024). When judges are not judging. University of Toronto Law Journal, 75(1), 2-44.

[8] He, X. (2021). Pressures on Chinese judges under Xi. The China Journal, 85(1), 49-74.

[9] Lei, L. (2017). Legal Methods, Legal Certainty and the Rule of Law. Renmin Chinese L. Rev., 5, 25.

[10] Zheng, Y., Ruan, M., Su, M., & Li, H. (2025). Territorial differences, development trends and influencing factors of judicial transparency in China. Territory, Politics, Governance, 1-22.

[11] Barman, R. (2023). Unveiling the Future: The Intersection of Artificial Intelligence and the Judicial System. Indian J. Integrated Rsch. L., 3, 1.

[12] Re, R. M., & Solow-Niederman, A. (2019). Developing artificially intelligent justice. Stan. Tech. L. Rev., 22, 242.

[13] Rigano, C. (2019). Using artificial intelligence to address criminal justice needs. National Institute of Justice Journal, 280(1-10), 17.

[14] Gans-Combe, C. (2022). Automated justice: Issues, benefits and risks in the use of artificial intelligence and its algorithms in access to justice and law enforcement. Ethics, Integrity and Policymaking: The Value of the Case Study, 175-194.

[15] Ejjami, R. (2024). AI-driven justice: Evaluating the impact of artificial intelligence on legal systems. Int. J. Multidiscip. Res, 6(3), 1-29.

[16] Bagaric, M., Svilar, J., Bull, M., Hunter, D., & Stobbs, N. (2021). The solution to the pervasive bias and discrimination in the criminal justice: transparent artificial intelligence. American Criminal Law Review, 59(1).

[17] Sil, R., Alpana, Roy, A., Dasmahapatra, M., & Dhali, D. (2021). An intelligent approach for automated argument based legal text recognition and summarization using machine learning. Journal of Intelligent & Fuzzy Systems, 41(5), 5457-5466.

[18] Radhika, A., Bhasin, N. K., Raju, Y. R., Satyanarayana, K. N. V., & Raj, I. I. (2024, March). Optimization of Natural Language Processing Models for Multilingual Legal Document Analysis. In 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS) (pp. 1-6). IEEE.

[19] Wei, H. (2024, May). Intelligent Legal Document Generation System and Method Based on Knowledge Graph. In Proceedings of the 2024 International Conference on Machine Intelligence and Digital Applications (pp. 350-354).

[20] Zhang, Y., Zhao, Y., & Zhao, Y. (2022). The Application of Artificial Intelligence Decision-Making Algorithm in Crisis Analysis and Optimization of the International Court System. Mobile Information Systems, 2022(1), 8150122.

[21] Liu, Y., Tan, T. P., & Zhan, X. (2024). Iterative Self-Supervised Learning for legal similar case retrieval. IEEE Access, 12, 17231-17241.

[22] Zhang, H., & Dou, Z. (2023, August). Case retrieval for legal judgment prediction in legal artificial intelligence. In China National Conference on Chinese Computational Linguistics (pp. 434-448). Singapore: Springer Nature Singapore.
[23] Shang, X. (2022). A computational intelligence model for legal prediction and decision support. Computational Intelligence and Neuroscience, 2022(1), 5795189.
[24] Ulenaers, J. (2020). The impact of artificial intelligence on the right to a fair trial: towards a robot judge?. Asian Journal of Law and Economics, 11(2), 20200008.
[25] Guo, H. (2024). Design of judicial public opinion supervision and intelligent decision-making model based on Bi-LSTM. PeerJ Computer Science, 10, e2385.
[26] Zhang, N., Pu, Y. F., Yang, S., Gao, J., Wang, Z., & Zhou, J. L. (2018). A Chinese legal intelligent auxiliary discretionary adviser based on GA-BP NNs. The Electronic Library, 36(6), 1135-1153.
[27] Lian, Y., & Yang, Y. (2024). Development and application of intelligent judicial trial assistance system based on generative artificial intelligence and machine learning technology. Applied and Computational Engineering, 75, 223-229.
[28] Ma, W. (2022). Artificial Intelligence-Assisted Decision-Making Method for Legal Judgment Based on Deep Neural Network. Mobile Information Systems, 2022(1), 4636485.
[29] Ng, Y. F., Windholz, E. L., & Moutsias, J. (2023). Legal considerations in machine-assisted decision-making: Planning and building as a case study. Bond Law Review, 35(1), 143-164.
[30] Haoxiang, W., & Smys, S. (2021). Big data analysis and perturbation using data mining algorithm. Journal of Soft Computing Paradigm (JSCP), 3(01), 19-28.
[31] Han Yu ,Xingjie Li ,Xue Hao ,Zhaowei Song ,Shangyu Liu ,Xinyue Li ... & Huasheng Xie. (2024). Improving Scattered Defect Grading in Castings Digital Radiographs via Smoothing the One-Hot Encoding. International Journal of Metalcasting,19(1),1-13.
[32] Zhan Zerui. (2025). Comparative Analysis of TF-IDF and Word2Vec in Sentiment Analysis: A Case of Food Reviews. ITM Web of Conferences,70,
[33] Enes Celik & Sevinc Ilhan Omurca. (2024). Skip-Gram and Transformer Model for Session-Based Recommendation. Applied Sciences,14(14),6353-6353.
[34] Yusung Sim,Wonho Shin & Sungho Lee. (2025). Automated code transformation for distributed training of TensorFlow deep learning models. Science of Computer Programming,242,103260-103260.