Data & Knowledge Engineering 147 (2023) 102203

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Modified Hierarchical-Attention Network model for legal judgment predictions

G. Sukanya[1], J. Priyadarshini[2,*]

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

ARTICLE INFO

Keywords:
Legal judgement prediction
Modified Hierarchical Attention Network (M-HAN)
Sentiment polarity feature
Hybrid classifier
BI-LSTM
Self-Improved Recurrent Neural Network (SI-RNN)

ABSTRACT

The impact of Artificial Intelligence in Legal Research has reached a high level in simulating human thought processes. Case Pendency is a long-lasting problem in many countries. The judicial system has to be more competent and reliable to provide justice on time for any developing country. Litigants and attorneys devote more time and effort to trial case preparation in the courtroom. The task of decision prediction is to automatically forecast the type of charge, law article, and term of punishment. Most of the earlier works for verdict prediction focused to work on civil law jurisdictions. Some of the challenges in the task are case facts are highly unstructured lengthy documents with a lack of annotations and mainly used machine learning techniques. While most research works ignore the information loss at the encoding stage, our proposed MHAN overcomes the above issue and long-range dependency problem using the attention model over hierarchical encoders with three tiers namely Sentence encoder, word encoder, and character encoder. To avoid information loss, a brand-new judgment prediction framework called MHAN is developed in this study effort. It is built on a modified Hierarchical-Attention network and a specially designed domain-specific word embedding model. Additionally, it emphasizes the feature extraction phase by joining features obtained using MHAN with an improved cosine similarity feature. Finally, a hybrid Self Improved RNN is employed to provide the projected results. Furthermore, the proposed model is trained on 10 types of real-time criminal cases from the Madras High Court of India and Supreme Court of India. It has outperformed prior methods in terms of verdict prediction. By applying different variations of the deep learning model and ablation tests, the proposed model achieves consistent results over baseline models.

1. Introduction

Legal intelligence is in its infancy and has infiltrated all aspects of the legal world. Advanced Natural Language Processing principles have paved the way for a slew of AI applications in the legal field. Textual representations of legal information are common (e.g., legal cases, contracts, bills). Nowadays, legal text processing is a growing field in NLP. "Legal subject categorization, court opinion production and analysis, legal information extraction, and entity recognition" are just a few of the uses for legal text in NLP [1–5]. In countries that follow civil law jurisdiction systems, judgement [6] is given based on statutes and briefcase facts. They do not take into account the precedents of previous cases as such in common law systems followed in India. Most existing research

* Corresponding author.
E-mail addresses: Sukanya.g2020@vitstudent.ac.in (G. Sukanya), Priyadarshini.j@vit.ac.in (J. Priyadarshini).
1 Research Scholar.
2 Professor.

https://doi.org/10.1016/j.datak.2023.102203
Received 12 May 2022; Received in revised form 15 March 2023; Accepted 9 July 2023
Available online 28 July 2023

Nomenclature

HAN	Hierarchical Attention Network
M-HAN	Modified Hierarchical Attention Network
NPV	Negative Predictive Value
Bi-LSTM	Bi- Long Short-term Memory
NLP	Natural Language Processing
SI-RNN	Self-Improved Recurrent Neural Network
CNN	Convolutional Neural Networks
FDR	False Discovery Rate
LJP	Legal judgment prediction
DAG	Directed Acyclic Graph
MCC	Mathews Correlation Coefficient
FNR	False Negative Rate

works are based on civil law jurisdiction only. For the past several years, models that anticipate legal outcomes have proliferated, and these models have aided solicitors as well as citizens in reducing legal expenses and increasing access to justice. Lawyers and judges use the legal judgement prediction model to assess the chances of winning a case and make more consistent and informed decisions [7–9]. They can be used by human rights groups and law scholars to assess the fairness of court decisions and determine whether they are correlated without biases.

Artificial Intelligence is being transformed with the rapid growth of deep learning technologies. [10–15]. The objective of automatic charge prediction is to assess a case by analyzing the description of its textual reality, such as robbery, theft, or fraud. This is a key element of the legal aid system and also benefits numerous applications in the real world. It can give the expert judge for the legal professions with a convenient and dependable reference. It could also provide legal consultation services for regular people, which have been particularly beneficial to those who do not know the legal language and intricate processes [16–20].

Benefits of using ML in legal Judgment prediction

- Quick decision-making and results.
- Instantaneous data validation.
- The unprejudiced frame of view.
- The ability to quickly highlight the previous cases with comparable trends.
- Identifying situations with substantial variation in human and AI choices makes it easier to uncover corruption.

Machine learning has already begun to make changes in different sectors, and it will undoubtedly assist India's judicial system in producing more accurate and precise legal decisions. [18–20]. Manupatra, a legal analytics and visualization platform in India employs AI through its Case Map and Taxonomy and provides an interpretation of a judge's decision based on data processed by AI. Contract management, client due diligence, and forecasting legal outcomes are all areas where machine learning is already being used in law. The progress of machine learning in NLP is making decisions [18,20]. Deep learning has employed the attention mechanism in several situations, including image captioning, image creation, and language modeling and translation. It is even inspired study on representations for the machine counterpart of "consciousness". As a result, the attention mechanism is being used in the judgement prediction model to extract words that are relevant to the sentence's meaning and composite the description of those informative words to build a sentence vector. Furthermore, the facts in the judgements have a hierarchical structure, and not all words contribute equally to the presentation of the sentence content. As a result, to represent the hierarchical links of words, sentences, and documents for document categorization, a hierarchical attention model is necessary. The HAN [21–23] is a deep neural network used for document classification. A HAN tries to categorize a text using the information it can infer about this from its composite components or the phrases and words that make up the document. The term HAN derives from the fact that this knowledge is arranged hierarchically, commencing with the words in a phrase and progressing to the sentences in a document [19]. The word sequence encoder is used to encode the word vectors, and the word-level attention layer is used to aggregate the information of the informative words, which do not contribute evenly. The penalty prediction is considered as the legal task [24]. Automatically predicting the results of court decisions is a task known as legal judgement prediction [25]. The sentence-level attention mechanism has been used to reward attention to sentences that provide hints for accurately classifying a text [26]. Prior research has generally concentrated on ontology development, or pattern discovery to address the text-based prediction challenge. The major challenges of the existing system are misjudgement of the penalty prediction and the reduced accuracy of the prediction. Sometimes the process may be tedious due to the unstructured lengthy documents and overfitting is also occurred. To overcome all these drawbacks the proposed system is developed.

The major contribution of this research work is:

• Proposed a Modified Hierarchical Attention Network (M-HAN) with improved cosine similarity-based sentiment polarity feature and hybrid deep classifier for prediction, by merging Self-Improved Recurrent Neural Network (SI-RNN) and Bi-LSTM

This is the first attempt to model a judgment prediction for Indian High Court cases. The rest of this paper is organized as: The literature works on LJP are depicted in Section 2. In addition, Section 3 depicts the proposed judgment prediction model, Section 4 portrays pre-processing via tokenization, and Section 5 addresses about modified hierarchical attention network (M-HAN) with improved cosine similarity-based sentiment polarity predicted features. The results acquired with the proposed work are discussed comprehensively in Section 6. This paper is concluded in Section 7.

2. Literature review

2.1. Related works

In 2018, Zhong et al. [27] developed a topological multi-task learning framework, TOP JUDGE, through the formalization of the subtask dependence as a DAG. The TOP JUDGE proposal contained several tasks and DAG dependencies in the prediction of LJP. Furthermore, the data description was generated via the encoder based on CNN. The results of the TOP JUDGE model have shown that the existing models were more consistent.

In 2019, Chalkidis et al. [28] With cases from the European Court of Human Rights, they have created a new English legal decision prediction dataset. The suggested dataset was used to test a wide range of neural models, including BiGRU-Att, HAN, LWAN, BERT, and HIER-BERT. The suggested dataset outperformed current algorithms in terms of binary violation classification, multi-label classification, and case significance prediction.

In 2018, Long et al. [29] developed a legal reading comprehension strategy based on the legal situation. The goal of this approach was to be able to handle various and complicated textual inputs. They have also included the AutoJudge, which incorporates law articles for judgement prediction. In terms of consistency and dependability, the suggested model performed better than the baseline model.

In 2019, Anand et al. [30] have created simple general approaches for summarizing Indian court decision documents using neural networks. Two neural network topologies were used to learn the semantics of the input phrase. The authors addressed the problem of labeled data scarcity by allocating classes/scores to phrases while training, and this was based on their similarity to human-produced reference summaries.

In 2020, Guo et al. [31] have introduced the TenLa by combining the ideas of tensor decomposition with an improved Lasso regression model. The suggested approach investigates the commonalities between judicial cases and uses them as a predictor of future judgments. The suggested works used the following key processes: "(a) ModTen to represent legal cases as three-dimensional tensors, (b) ConTen to deconstruct tensors generated by ModTen into core tensors via an intermediate tensor, and (c) OLass to train with ConTen's Core tensors". The TenLa's results were more accurate than those of the traditional models.

In 2020, Guo et al. [32] have proposed TenRR, a model that combines tensor decomposition and ridge regression for predicting judicial judgments, and the suggested model included three key contributions. RTenr was created as a tensor representation method to describe legal situations as three-dimensional tensors in the first contribution. The ITend was used to deconstruct the original tensors representing legal cases into core tensors in the second contribution. The ORidge was created as part of the contribution to creating a judgement prediction model for judicial situations. The suggested work's results were more accurate than standard approaches for predicting judgments.

In 2019, Chen et al. [33] have used a deep learning model to assess the case's fundamental summary and predict the outcome. The deep learning model's output consisted of three elements: punishment, allegation, and legal provisions. The punishment prediction model was constructed using FastText and TextCNN, the legal provisions prediction model has been built using TextCNN, and the accusation prediction model was built using FastText and TextCNN. The suggested judicial decision-making approach produced more accurate and convincing results.

In 2019, Yang et al. [34] have presented a Multi-Perspective Bi-Feedback Network with a word collocation attention mechanism based on the topological structure among subtasks. A multi-perspective forward prediction and backward verification framework was also developed to manage the relationships between numerous subtasks. To differentiate situations with identical descriptions and distinct penalties, the suggested approach included the word collocations aspects of fact descriptions. Finally, it was observed that the suggested work has significantly improved the prediction accuracy. Table 1 shows the existing system Methodology, Dataset, Advantages, and Drawbacks.

3. Proposed judgment prediction model

As an effective and essential application in legal assistant systems, legal judgement prediction (LJP) seeks to determine the judgement outcomes based on information obtained from factual identification. In real-world settings, judges examine not just the facts of the case, but also external information including the defendant's fundamental knowledge and the court's perspective while dealing with criminal cases. Most existing studies, use the statement of the facts as the only input to LJP and disregard the external data.

Table 1
Review of collected research papers: Methodology, dataset, advantages, and drawbacks.

Author [Citations]	Methodology	Data collection	Features	Challenges
Zhong et al. [27]	TOP JUDGE	“**CJO, PKU**, and **CAIL**. CJO consists of criminal cases published by the Chinese government from China Judgement Online .PKU contains criminal cases published by Peking University Law Online CAIL(Chinese AI and Law Challenge)”	✓ Sustainable and persistent gains are achieved ✓ Topological learning to anticipate judgments ✓ Any kind of subtask reliant on DAG may be handled	⋄ The efficacy of TOP JUDGE must be investigated ⋄ need to investigate how the time element may be included in LJP
Chalkidiset al. [28]	English legal judgment prediction dataset	“English legal judgment prediction dataset”	✓ Classification of binary infringement by Surpass ✓ Classification of multi-label ✓ The prediction of case significance	⋄ Little-shot learning is not considered ⋄ need to break into separate subtasks the problem of charge prediction
Long et al. [29]	AutoJudge	“Chinese Referee Document Network”	✓ Improved F1 score, accuracy, and precision	⋄ have no access to elements of basis truth legislation ⋄ increasing the complexity of computing ⋄ Decreases precision and stability
Anand et al. [30]	neural network		✓ Address the problem of labeled data unavailability ✓ have a high-level coherence summarizes	⋄ More expenses ⋄ More complicated computing
Guo et al. [31]	TenLa	“3,000,000 legal cases in the past five years from multiple provinces and cities in China”	✓ higher accuracy ✓ eliminates information that is unnecessary, insignificant, and incorrect	⋄ Need to prevent overfitting ⋄ tedious
Guo et al. [32]	TenRR	“Chinese Referee Document Network”	✓ Reduce the initial tensor dimension considerably ✓ remove incorrect, irrelevant and superfluous tensor data;	⋄ the exactness of forecasts must be improved
Chen et al. [33]	FastText andTextCNN	“CAIL 2018 data set”	✓ more accurate and persuasive decision-making ✓ Better in accuracy, recall rate and other indicators.	⋄ Need to improve the accuracy of model prediction
Yang et al. [34]	Multi-Perspective based BiFeedback Network (MPBFN) and a Word Collocation Attention (WCA) mechanism	“Chinese AI and Law challenge (CAIL2018) *i.e.*, CAIL-small (the exercise stage data) and CAIL-big (the first stage data).”	✓ improves the overall performance ✓ improve the performance of multitasking	⋄ Need to reduce the misjudgment of penalty prediction

3.1. Proposed architecture MHAN

In this research work, a novel judgment prediction model based on the modified Hierarchical-Attention network is introduced to predict the judgment on Indian criminal cases of Madras High Court. The proposed judgment prediction model is made-up of 3 major phases: (a) pre-processing, (b) feature extraction and (c) judgment prediction phase. Initially, the collected raw data (collected judgment document data) is subjected to pre-processing phase. The collected input judgment document is often long document containing the long-term dependency problem. These bulkier data is complex to process and tends to decrease the prediction accuracy. Therefore, it is tokenized both in the sentence level, word level and character level (as the modified HAN). Improved cosine similarity-based sentiment polarity predicted features are computed based on these encodings and these features are considered for judgment prediction with a hybrid deep learning model. The hybrid classifier (SI-RNN and Bi-LSTM) is trained with the features extracted from M-HAN and computes the sentiment score. During the testing phase, the testing data is fed as input and the corresponding legal judgment predictions is based the pre-trained model. The overall architecture of the proposed work is shown in Fig. 1.

4. Pre-processing phase

4.1. Tokenization

Initially, the collected raw data (judgment document dataset) is pre-processed to transform the raw data into a meaningful format. Therefore, in this research work, the collected raw data D^{in} is read initially using the openCV function, and it is tokenized. Every task in NLP requires tokenization to enhance the meaning of the text by analyzing the sequence of words. In this research work, the raw data D^{in} is disintegrated into pieces called tokens. The tokens are often the input for parsing and text mining. The sentence, as well as word-level tokenization, is undergone in this research work. The extracted tokens acquired at the end of pre-processing phase are denoted as D^{token}.

5. Feature extraction phase (M-HAN)

Feature extraction is a technique of dimensionality decreasing the data into more manageable processing groups. A feature of those huge data sets is many variables that have to be processed using a significant amount of computer resources. Though deep learning models automatically learn the features from the whole corpus, if the emphasis is given to this phase important information will not be lost. This extraction procedure is beneficial to minimize the number of processing resources without losing vital or relevant information. The extraction of features also helps decrease the redundancy of data for a particular investigation. The reduction of data and efforts of the machinery in constructing variable combinations (functions) also contribute to the speed at which the machine is trained and generalized. Our proposed framework uses the MHAN algorithm for feature extraction. The algorithm uses sentence tokens from the corpus and is sent to MHAN Encoder, Muti-feature Selector, and Cosine Similarity Feature Selector. Then the above three features with static dimensions are concatenated and sent to hybrid classifier.

5.1. M-HAN (Modified Hierarchical Attention Network)

M-HAN is an extended version of standard HAN.

HAN: Machine learning advancements in NLP are making decision-making capabilities extremely important. Bots have already been employed in the industry to "smart search" previous judgements and decisions to assist in the preparation of new lawsuits. The majority of contemporary judgement prediction models are based on standard machine learning techniques. Attention mechanisms are now employed in deep learning in a broad array of applications, including image captioning, image creation, and language processing and interpretation. As a result, the attention algorithm is implemented in the judgement prediction model to retrieve keywords from a big dataset by giving them higher weights. For document categorization, a hierarchical attention model is used to model the hierarchical relationships of words and sentences in a document. A HAN tries to characterize a text depending on the information it can glean from its composite parts or even the phrases and words which constitute the document. The term "hierarchical" in HAN is attributed to the idea that such information is organized hierarchically, beginning with the use of words in a phrase and progressing to the use of sentences in a document [22]. "A word sequence encoder, a word-level attention layer, a sentence encoder, and a sentence-level attention layer" are the components of HAN. The word sequence encoder is being used to encode the word vectors, and the word-level attention layer is being used to aggregate the information of informative words which do not participate proportionately. The sentence vectors are therefore stored in the word sequence encoder, with the sentence-level attention mechanism rewarding attention (weights) to the sentences that would provide hints to appropriately distinguish a text.

M-HAN: In order to enhance the document classification accuracy of HAN, a Modified Hierarchical Attention Network (M-HAN) is introduced in this research work. The proposed M-HAN is modeled with 3 level encoders: character encoder, word encoder, and sentence encoder. A stack of Bi-GRU encoders is used in M-HAN. The word encoder and sentence encoder are already present in HAN. We have introduced the character encoder within the HAN to enhance the document classification by overcoming the issue of Out Of Vocabulary Words(OOV). In addition, the weights W of encoders in M-HAN are normalized using the sigmoid function rather than the existing softmax function; therefore the classification problems can be solved. All these together aids in enhancing the judgment prediction performance of the projected model.

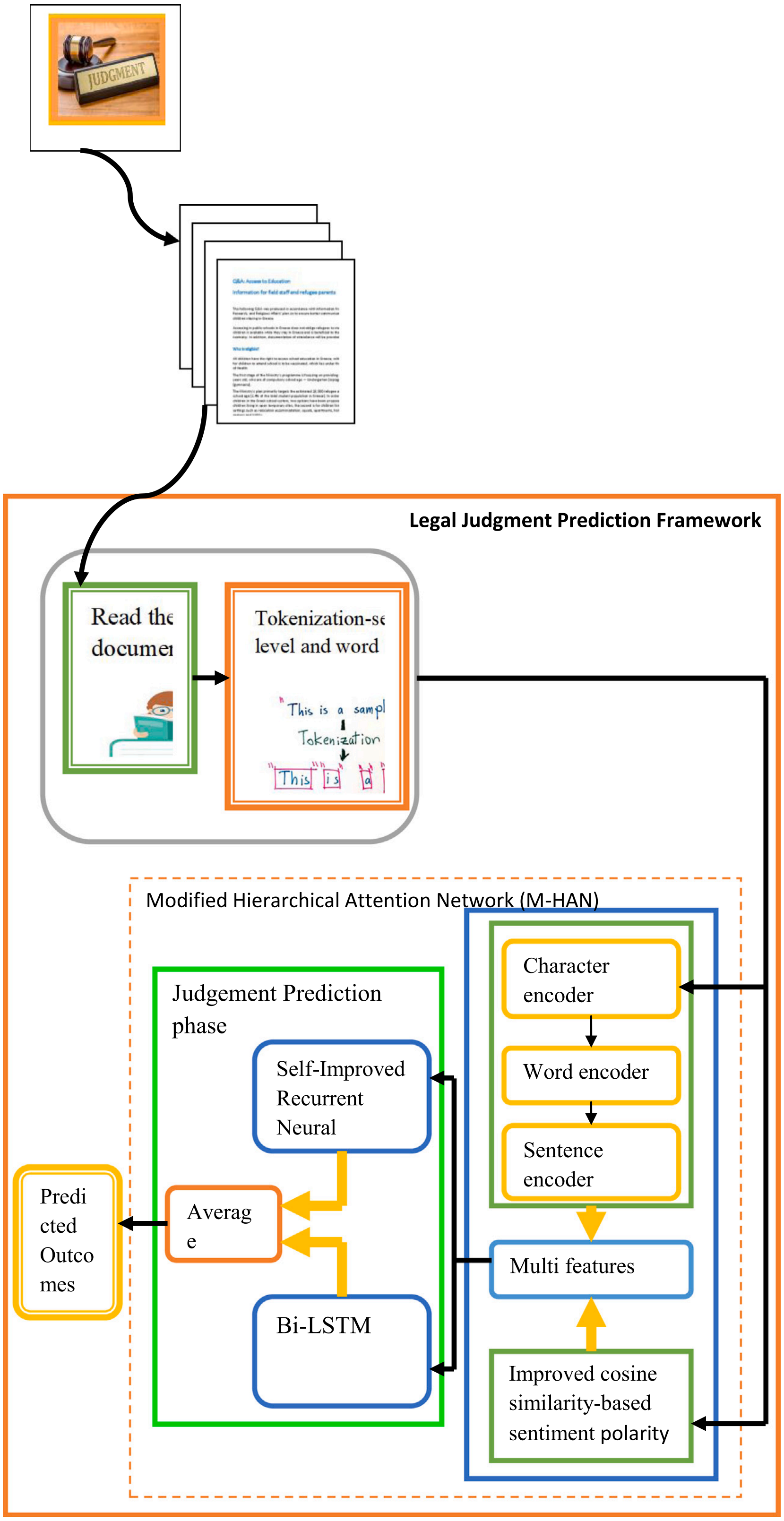

Fig. 1. Architecture of the MHAN Judge Prediction Model.

(a) Character attention: This is accomplished to characterize the characters in the documents. Character-level translation has several advantages over word-level translation. Character level approaches are devoid of out-of-vocabulary problems, can represent distinct, unusual morphological variations of a word, and do not need segmentation. Text segmentation is particularly difficult for several languages, and even for English because word tokenizers are either explicitly constructed or trained on a corpus using an objective function unrelated to the translation task given, resulting in a sub-optimal system. This is mathematically modeled in Eq. (1), Eq. (2), and Eq. (3), respectively.

$$u_i = \tanh\left(W_s h_i + b_s\right) \tag{1}$$

$$\alpha_i = \frac{\exp(u_i^{\mathrm{T}}.u_c)}{\sum_t \exp(u_i^{\mathrm{T}}.u_c)} \tag{2}$$

$$c = \sum_t \alpha_i.h_i \tag{3}$$

Here, c is the vector of the characters in the document. The features extracted from M-HAN are denoted as f^{M-HAN}.

(b) Word attention: All the tokens in D^{token} do not equally contribute to the meaning of a statement. This attention mechanism extracts the essential words that provide meaning to the sentence and aggregates the representation of those informative words to build a sentence vector. This is mathematically modeled in Eq. (4), Eq. (5), and Eq. (6), respectively.

$$u_{it} = \tanh\left(W_\omega h_{it} + b_\omega\right) \tag{4}$$

$$\alpha_{it} = \frac{\exp(u_{it}^{\mathrm{T}}.u_\omega)}{\sum_t \exp(u_{it}^{\mathrm{T}}.u_\omega)} \tag{5}$$

$$s_i = \sum_t \alpha_{it}.h_{it} \tag{6}$$

Here, h_{it} denotes the word annotation, which is fed as input to the one-layer MLP to attain the hidden representation u_{it} of h_{it} (hidden state) for ith sentence at a time t. Then, the similarity of u_{it} with words level context vector u_ω and α_{it}(normalized importance weight) through a softmax function. Further, the sentence vector s_i is computed as a weighted sum of the word annotations based on the weights. The context vector u_ω can be seen as a high-level representation of a fixed query "what is the informative word" over the words like that used in memory networks. The word context vector u_ω is randomly initialized and jointly learned during the training process.

(c) Sentence attention: Sentence Attention utilizes an attention mechanism to reward sentences that indicate to correctly categorize a document, and it is used to gauge the significance of the sentences. This is mathematically modeled in Eq. (7), Eq. (8), and Eq. (9), respectively.

$$u_i = \tanh\left(W_s h_i + b_s\right) \tag{7}$$

$$\alpha_i = \frac{\exp(u_i^{\mathrm{T}}.u_s)}{\sum_t \exp(u_i^{\mathrm{T}}.u_s)} \tag{8}$$

$$v = \sum_t \alpha_i.h_i \tag{9}$$

Where v is the vector of the document summarizing all sentence information in the document. Likewise, throughout the training phase, the context vector of the phrase level might be randomly initialized and trained along.

In addition to this, we have also computed the polarity of the sentiment scores of D^{token} using the improved cosine similarity-based sentiment polarity predicted features

5.2. Proposed improved cosine similarity based sentiment polarity predicted features and Multi-Feature Extraction

The sentiment polarity score of D^{token} is estimated using word2Vec embedding. The score is computed using the improved cosine similarity, rather than the existing cosine similarity. Since in the traditional cosine similarity the magnitude of vectors is not taken into account. Therefore, to overcome the drawbacks of traditional cosine similarity, improved cosine similarity has been developed in this work. And, to make the document more influential, a weightage is given for the document and it is included along in the improved cosine similarity to make it efficient. Mathematically, the improved cosine similarity can be expressed as per Eq. (10), from which the polarity score is calculated.

$$f^{I-POL} = 1 - \frac{\left(\sum_{i=1}^{n} A_i * B_i\right)}{\left(\sum_{i=1}^{n} A_i^2\right)^{1/2} * \left(\sum_{i=1}^{n} B_i^2\right)^{1/2}} * W \tag{10}$$

Here, the weight of the document is denoted as W, and it ranges between [0–1] via the Chaotic map function. The features extracted from improved cosine similarity-based sentiment polarity predicted features are denoted as f^{I-POL}.

5.3. Multi-features

The features acquired from M-HAN f^{M-HAN} and Improved cosine similarity-based sentiment polarity predicted features f^{I-POL} is subjected to the judgment prediction phase. The extracted overall features are denoted as $F = f^{I-POL}+f^{M-HAN}$, using which the hybrid classifiers in the judgment prediction phase have been trained.

6. Judgement prediction phase

The judgment prediction phase is modeled with two deep learning classifiers Bi-LSTM and SI-RNN, respectively. Both classifiers are trained concurrently using the extracted features F. Both classifiers execute parallel and produce their corresponding outcomes: out^{SI-RNN} from SI-RNN and out^{Bi_LSTM} from BI-LSTM. The final legal judgment predicted outcome is the mean of out^{SI-RNN} and out^{Bi_LSTM}.

6.1. SI-RNN

The RNN, is a well-known deep learning model with three layers: input, hidden state, and output state. The extracted feature F is fed into the SI-RNN's input layer $I(t)$. The hidden layer $B(t)$ of the RNN detects the features, and the resulting outcomes are revealed in the form of labels through the output layer $O(t)$. The time step index is denoted by t. Furthermore, in SI-RNN, the hidden layer functions as a memory device. At each t^{th} time step, the hidden state $B(t)$ is computed using Eq. (11), which corresponds to non-linear transformations fun such as tanh and ReLu. SI-RNN also has three weight matrices $\left(W = \left\{W_{I,B}, W_{B,B}, W_{\mathbf{B},O}\right\}\right)$. Here, $W_{I,B}$ establishes the link between the input and the hidden layer; $W_{B,B}$ establishes the interlining between the hidden layers; and $W_{\mathbf{B},O}$ establishes the connection between the hidden and the output states. All of these weight parameters $\left(W = \left\{W_{I,B}, W_{B,B}, W_{\mathbf{B},O}\right\}\right)$are exchanged over time.

$$B(t) = fun.(W_{I,\mathbf{B}}.F(t) + W_{B,B}.B(t-1) \tag{11}$$

The SI-RNN architectural phases are as follows:

- Step 1: Set the bias function to $b, c = 0$. and initialize $W_{I,B}, W_{B,B}, W_{B,O}$.
- Step 2: The forward propagation function is used to select the appropriate features, which is represented in Eq. (12) to (15), respectively. At the post-processing step, the softmax function is utilized to obtain the normalized probabilities of the observed outcomes with less computation difficulty. The final selected features are denoted as $\hat{O}$.

$$a(t) = b + W_{I,B}.F(t) + W_{B,B}.B(t-1) \tag{12}$$

$$B(t) = \tanh.(a(t)) \tag{13}$$

$$O(t) = c + B(t).W_{B,O} \tag{14}$$

$$\hat{O}(t) = soft\max(O(t)) \tag{15}$$

In the traditional RNN, the mean square error is computed. As a novelty. the new loss function $Loss$ is calculated as in Eq. (16). Since the MSE makes a great loss metric for a model to optimize. Thus, a new loss is computed based on the adaptive weight factor. Here, $f(y)$ is the function of output and $ground$ is the ground truth label, and D is the adaptive weight factor that is computed as per Eq. (17).

$$D(f(y), ground) = \begin{cases} \dfrac{|(f(y), ground)|}{2}; & if\ |(f(y), ground)| < C \\ |(f(y), ground)| & 0 \end{cases} \tag{16}$$

$$\begin{aligned} Loss(f(y), ground) &= mean(D(f(y), ground)) \\ &* mean([(predicted - actual)]^2) \end{aligned} \tag{17}$$

Here, weight is halved, when the absolute difference between the predicted and generated truth is less than a particular constant C. The value of C is fixed as 10 in this work.

6.2. BI-LSTM

A bi-directional LSTM or Bi-LSTM is a sequence processing model composed of two Bi-LSTM's: one takes the entry forward, the other reverse. Bi-LSTM'sefficiently enhance the quantity of information the network offers, and improve the algorithm context (e.g. knowing what words immediately follow and precede a word in a sentence). Bi-LSTM [35] is indeed a deep learning architecture based on an artificial recurrent neural network (RNN). In terms of memory, Bi-LSTM is indeed a sort of recurrent neural network that significantly outperforms standard RNN. BI-LSTMs perform dramatically superior when it comes to memorizing specific patterns. Bi-LSTM, like any other NN, can have multiple hidden layers, and as it goes through each layer, the relevant information remains retained while the irrelevant data is eliminated in each cell. The extracted feature F is fed into Bi-LSTM.

1. FORGET Gate: This gate determines which data should be retained for computing the cell state and what should be discarded. The information from the preceding hidden stage (previous cell) is represented by hid_{t-1}, while the data from the current cell is represented by x_t. The Forget gate receives exactly two inputs. They're run via a sigmoid function, with the ones that tend to 0 getting eliminated and the rest being sent on to determine the cell state.
2. INPUT Gate: The Input Gate modifies the state of the cell and determines which data is relevant and which is not. The input gate, like the forget gate, aids in the identification of crucial data as well as the storage of relevant material in the memory. The inputs hid_{t-1} and x_t are sent through the sigmoid and tanh functions, respectively. The tanh function controls the network and minimizes bias.
3. Cell State: The new cell state gets computed using all of the information gathered. The cell state is initially multiplied by the forget gate's output. If multiplied by values close to 0, this would have the potential to drop values in the cell state. The cell state would then be updated to new values that perhaps the neural network considers relevant using a point-wise addition with the output from the input gate.
4. OUTPUT Gate: The final gate, the Output gate, determines whether the next concealed phase will be. A sigmoid process is defined using the parameters hid_{t-1} and x_t. The newly implemented cell state is then sent through the tanh function and multiplied by the sigmoid output to determine what information should be stored inside the hidden state.

X_t = Input vector at the t-time.
Hid_{t-1}= Previous Hidden state.
Mem_{t-1}= Previous Memory state.
Hid_t= Current Hidden state.
C_t= Current Memory state.
[*] = multiplication operation.
[+] = addition operation.

As a result, each BI-LSTM module's inputs are X_t (current input), Hid_{t1}, and C_{t1}. The outputs are Hid_t and C_t.

The gates Inp, for, out are the input, forget as well as output gates. The function formula (sigmoid) is the same as for input, forget, and output, with the matrix parameters being the sole difference. This indicates that the gate's output is indeed a vector of values ranging from 0 to 1. A value of 0 indicates that almost all information is blocked, whereas a value of one indicates that all information has been included. The gate input determines how many states have been passed for the current input that has just been computed. The forget gate determines how many prior states have been let through. Ultimately, the gate output determines how many internal states are accessible to the network (higher layer & next time step). As a measurement for the hidden state, all gates have the same dimensions as the hidden state dimension. The sigmoid gate's output will be multiplied by that other value to determine how much of that value is consumed. Mathematically, Inp, for, out are given in Eq. (18), Eq. (19), and Eq. (20), respectively.

$$Inp = \sigma(x_t.U_{inp} + s_{t-1}.W_{inp}).inp = \sigma(x_t.U_{inp} + s_{t-1}.W_{inp}) \tag{18}$$

$$for = \sigma(x_t.U_{for} + s_{t-1}.W_{for}).inp = \sigma(x_t.U_{for} + s_{t-1}.W_{for}) \tag{19}$$

$$out = \sigma(x_t.U_{out} + s_{t-1}.W_{out}).inp = \sigma(x_t.U_{out} + s_{t-1}.W_{outr}) \tag{20}$$

1. g is a hidden "candidate" state that is also calculated using current and hidden information.
2. C_t is the unit's inner memory. A mixture of the former C_{t-1} memory, the forgot gate, and the hidden state g, the newly computed, multiplied by the input gate. It is therefore an obvious mix of how we integrate the earlier memory with the fresh input. Mathematically, C_t is given in Eq. (21).

$$C_t = for_t + C_{t-1} + inp_t * g \tag{21}$$

The extracted features from Bi-LSTM are denoted as out^{Bi_LSTM}.

7. Results and discussion

7.1. Dataset description

The proposed judgment prediction model has been implemented in PYTHON. The proposed work consists of two datasets namely Madras High Court dataset 1 and Supreme Court of India dataset 2. The dataset1 uses real-time criminal cases from Madras High Court published publicly in the Manupatra platform dataset acquired for Indian cases. Among the collected 500 cases, 15 different case files have been collected inclosing: (a) Madras HC abortion cases having 32 files, (b) Madras HC child abuse cases having 30 files, (c) Madras HC Confiscation property cases with 37 files, (d) Madras HC corruption cases with 30 files, (e) Madras HC felony cases with 30 files, (f) Madras HC Fraud cases enclosing 30 files, (g) Madras HC kidnap cases enclosing 29 files, (h) Madras HC organ theft cases with 31 files, (i) Madras HC rape cases with 29 files, (j) Madras HC trafficking cases with 30 files, (k) Madras HC White Collar crimes Cases with 30 files, (l) Madras High Court Animal cruelty cases with 30 files, (m) Madras High Court Forgery Cases encapsulating 30 files, (n) Madras High court Murder Cases with 30 files and (o) Madras High Court Theft cases enclosing 30 files within it. Dataset 2 uses the Supreme Court of India and it holds 200 cases, 12 different case files have been

Table 2
Sample judgment.

Sample judgement	Class value
"1 . This Criminal appeal arises out of the Judgment of conviction and sentence dated\n01.03.2007 in S.C. No. 121 of 2002 on the file of the Additional Sessions Court (Fast\nTrack Court), Dharmapuri, whereby the appellant/accused was convicted and sentenced\nas follows:"	1
"1 . The sole accused in S.C. No. 278 of 2009 on the file of the MahilaCourt,\nChengalpattu, is the appellant herein. He stood charged for the offences under Sections\n376 and 506(ii) IPC. When the appellant was questioned as to the charge, he pleaded\nnot guilty and therefore, he was put on trial. The Mahila Court, Chengalpattu, after full-\nfledged trial, found him guilty of the offences and convicted the appellant under Section\n376 IPC and sentenced him to undergo Rigorous Imprisonment for 7 years and to pay a\nfine of Rs. 2,000/-, in default, to undergo Rigorous Imprisonment for one (1) year and\nhe was found not guilty of the offence under Section 506(ii) IPC. Challenging the said\nconviction and sentence, the appellant is before this Court with this Criminal Appeal."	1
"Police, Sivagangai District and Others, dated 15.9.2008, and in W.P. No. 5677 of 2007,\nL. Ravindran v. The Commissioner of Police, Chennai and Others, dated 22.3.2012, Mr.\nPrasannaVinodh, learned counsel for the petitioner submitted that the Deputy\nSuperintendent of Police, Karur Town Sub Division, Karur, has failed to consider that\nthe petitioner was not involved in any serious crime affecting peace and public\ntranquility, and that therefore, there is non application of mind on the part of the\nDeputy Superintendent of Police, Karur Town Sub Division, Karur, the second\nrespondent herein, to the provisions warranting opening and retention of History Sheet.\nHe further submitted that had the Inspector of Police, Karur Town Police Station, Karur,\nforwarded the details of acquittal in Crime Nos. 2773 of 2009 and 80 of 2010, which are\nfavourable to the petitioner, then the second respondent would not have retained the\npetitioner in the History sheet. Thus, he submitted that there is a failure on the part of\nthe second respondent in taking into consideration, Police Standing Order 758 which\nreads that when any information favorable to an individual for whom a History Sheet is\nbeing kept is received, it shall be entered therein."	0
"65. I find that the appellant has not discharged the burden of proof to distance himself\nfor an acquittal but on the contrary has further exposed himself of having committed the\noffence by asking questions during course of cross-examination of PW 2 minor victim."	0

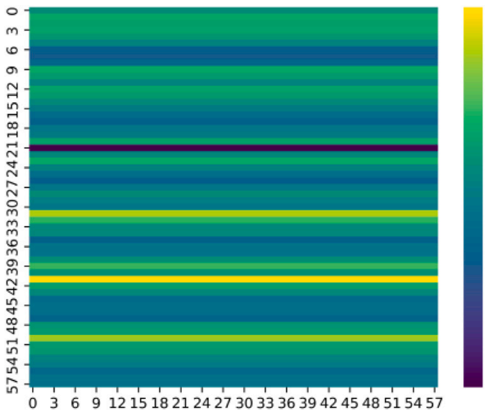

Fig. 2. Heatmap for M-HAN features.

collected inclosing (a) Labor matters include 14 files, (b) Rent case matters include 9 files, (c) Direct tax matters include 24 files (d) Indirect tax matters include 36 files (e) Land Acquisition and Requisition matters include 5 files (f) Service matters include 21 files (g) Academic matters include 7 files (h) Letter petition and pill master includes 18 files (i) Election matters include 12 files (j) Company Law, Mrtp, Trai, Sebi, Idrai & Rbi include 10 files (k) Compensation matters include 7 cases (l) Criminal matters include 39 cases. Among these data, 70% of the data have been utilized for training the projected model, while the rest 30% of the data have been utilized to test the projected work for judgment prediction. The 70% required for training (considered as 100%) was split into 4 groups of training percentages: 60%, 70%, 80%, and 90%, respectively. All these 4 groups of training percentages have been utilized to validate the projected model. To prove that the projected model has higher prediction performance than the existing models like DBN, SVN, NN, NB, LSTM, CNN, GRU, and English legal judgment prediction [21], M-AttBLSTM-CNN [19], two neural network architectures [30], respectively. The evaluation is mainly focused on the like "accuracy, precision, F_1-score and MCC", and statistical performance, respectively. The sample judgment and the predicted outcomes (0- not a crime, 1- crime) acquired by the proposed work are depicted in Table 2. The heat map for the extracted M-HAN features is shown in Fig. 2. The database information is provided in Table 3.

7.2. Performance analysis of dataset 2

The proposed work using dataset 2 is evaluated in terms of Positive metrics like "accuracy precision, F-measure, and MCC". The outcomes acquired are manifested in Tables 4–7. The accuracy of the proposed work at TP = 90 is 4.24%, 25.42%, 4.19%,

Table 3
Database information including training data count, testing data count, words count and facts count for dataset 1.

Subset	Cases in numbers	Words in numbers	Facts in numbers
Train	22	11 032	1. Madras HC abortion cases
Test	10	1254	
Train	21	10 354	2. Madras HC child abuse cases
Test	9	2663	
Train	26	18 324	3. Madras HC Confiscation property cases
Test	11	3025	
Train	21	8238	4. Madras HC corruption cases
Test	9	2177	
Train	20	10 734	5. Madras HC felony cases
Test	10	1789	
Train	21	19 523	6. Madras HC Fraud cases
Test	9	2430	
Train	20	12 254	7. Madras HC kidnap cases
Test	9	2012	
Train	22	9845	8. Madras HC organ theft cases
Test	9	1845	
Train	20	11 225	9. Madras HC rape cases
Test	9	3741	
Train	21	13 147	10. Madras HC trafficking cases
Test	9	2307	
Train	21	10 140	11. Madras HC White Collar crimes Cases
Test	9	2369	
Train	21	11 074	12. Madras High Court Animal cruelty cases
Test	9	1974	
Train	21	10 048	13. Madras High Court Forgery Cases
Test	9	1174	
Train	21	11 124	14. Madras High court Murder Cases
Test	9	2558	
Train	21	9787	15. Madras High Court Theft cases
Test	9	3112	

Table 4
Analysis on the proposed hybrid classifier for varying Test Percentage in terms of accuracy.

Training percentage (TP)	DBN	SVM	NN	NB	LSTM	CNN [33]	GRU	English legal judgment prediction [28]	M-AttBLSTM-CNN [19]	Two neural networkar-chitectures [30]	Hybrid classifier
60	74.15867	76.38165	75.26544	75.26544	83.22843	75.97549	72.78026	77.00408	76.11489	74.40956	87.75046
70	76.29273	76.68554	76.11489	76.18685	85.90275	76.64841	74.8835	78.81569	76.33051	75.40354	89.02564
80	77.37172	77.37172	76.38165	76.29273	85.91621	76.73733	76.20381	79.38599	77.35976	76.11489	91.79633
90	80.59082	78.60463	77.25323	79.49382	86.07385	77.01626	76.29273	80.38302	78.08264	76.68554	91.88998

11.33%, 3.32%, 2.37%, 3.04%, 1.43%, 3.48% and 2.92% better than the existing models like DBN, SVN,NN, NB, LSTM, CNN, GRU, English legal judgment prediction [28], M-AttBLSTM-CNN [19], two neural network architectures [30], respectively. Moreover, the precision of the proposed work had also attained the highest value at every variation in TP. The precision (96%) of the proposed work has recorded the highest value at every variation in TP, respectively. In addition, the F1-score of the proposed work at TP = 90 is 96.17, which is the best score while compared to existing models like DBN = 80.32, SVN = 78.70, NN = 76.51, NB = 77.44, LSTM = 95.93, CNN = 91.08, GRU = 75.81, English legal judgment prediction [28] = 86.33, M-AttBLSTM-CNN [19] = 80.64, two neural network architectures [30] = 95.93. Thus, from the overall evaluation, it is vivid that the proposed work is much applicable for legal judgment prediction.

7.3. Performance analysis of dataset 2

The suggested study employing dataset2 is assessed using "accuracy precision, F-measure, and MCC" as positive metrics. Tables 8 to 11 show the manifestation of the results attained. For the best outcomes, metrics like "accuracy, precision, MCC, and F-measure" should be maintained at a higher level. The introduction of the new M-HAM, which extracts the character level, word level, and sentence level encoding, is responsible for all of these advances. In comparison to existing models like DBN, SVN, NN, NB, LSTM, CNN, GRU, English legal judgment prediction [28], M-AttBLSTM-CNN [19], and two neural network architectures [30], the proposed work's accuracy at TP = 90 is 14 percent, 16.89 percent, 18.93 percent, 15.58 percent, 6.75 percent, 19.30 percent, 20.43 percent, 14.30 percent, and 19.82 percent better. Additionally, the planned work's precision has always achieved the highest value

Table 5
Analysis of proposed hybrid classifier for varying Test Percentage in terms of precision for dataset 1.

Training percentage (TP)	DBN	SVM	NN	NB	LSTM	CNN [33]	GRU	English legal judgment prediction [28]	M-AttBLSTM-CNN [19]	Two neural networkar-chitectures [30]	Hybrid classifier
60	65.43465	74.5153	75.10561	82.8131	75.164	69.68887	95.933	71.74351	75.89175	66.57902	96.17678
70	73.7011	75.852	75.68628	86.15289	78.60138	75.17816	95.933	84.14933	76.11489	69.96493	96.37447
80	95.933	78.49576	76.90285	86.49207	78.81406	78.1629	95.933	89.35736	76.77743	75.30371	96.34601
90	95.933	78.62204	77.67508	89.24501	79.94696	91.92602	95.933	90.54997	76.89145	79.48999	96.1738

Table 6
Analysis of proposed hybrid classifier for varying Test Percentage in terms of F1-score.

Training percentage (TP)	DBN	SVM	NN	NB	LSTM	CNN [33]	GRU	English legal judgment prediction [28]	M-AttBLSTM-CNN [19]	two neural networkar-chitectures [30]	Hybrid classifier
60	75.85504	77.00652	75.01497	65.58632	84.53347	72.87368	74.84817	65.58632	73.60242	75.67882	96.17678
70	76.153	77.61047	76.11389	69.95874	84.69267	76.74412	75.25395	68.86563	78.12169	76.153	96.37447
80	76.33539	78.30421	76.51737	76.87955	95.933	78.94649	75.43467	74.35112	78.99807	82.59018	96.34601
90	80.32577	78.70358	76.51737	77.44157	95.933	91.08113	75.81635	86.33542	80.64688	95.933	96.1738

Table 7
Analysis of proposed hybrid classifier for varying Test Percentage in terms of MCC.

Training percentage (TP)	DBN	SVM	NN	NB	LSTM	CNN [33]	GRU	English legal judgment prediction [28]	M-AttBLSTM-CNN [19]	two neural networkar-chitectures [30]	Hybrid classifier
60	69.83055	76.36279	75.80492	72.2664	84.66749	73.3895	67.46531	76.0725	75.15641	75.72597	86.24
70	71.65797	77.25148	75.89948	72.9858	86.80556	75.42568	67.49925	77.78454	77.10523	75.75513	88.82397
80	78.12501	77.8059	76.32185	73.19983	86.9351	77.01626	69.68096	78.36599	77.93053	77.5376	91.21765
90	83.73143	78.05061	76.70962	77.21587	87.61989	78.96176	69.89567	78.94737	78.19709	78.93494	91.30924

Table 8
Analysis of the proposed hybrid classifier for varying Test Percentage in terms of accuracy for dataset 2.

Training per-centage(%)	DBN	SVM	NN	NB	LSTM	CNN	GRU	English legal judgment prediction [2]	M-AttBLSTM-CNN [28]	Two neural network ar-chitectures [4]	Hybrid classifier
60	75.35223	74.4257	76.21585	71.7728	77.24131	78.29847	77.5425	76.85652	74.9422	76.67776	86.43183
70	80.82669	74.68273	80.91838	74.70896	81.89058	82.64054	82.2099	86.91219	81.08844	81.74969	88.3769
80	84.59255	74.86852	85.44674	80.64653	86.61029	87.22972	86.91266	89.39945	85.69812	85.83541	90.92317
90	90.10188	74.88794	90.14927	84.36567	90.90735	91.7483	91.15575	92.59737	90.76178	91.25956	93.92919

at different TPs. The suggested work's precision (97%) consistently reported the maximum value for each change in TP. Therefore, it is clear from the overall assessment that the suggested work is very applicable for judicial judgment prediction.

7.4. Statistical analysis for dataset 1 and dataset 2

The statistical performance like mean, median, Standard Deviation, the Worst and best performance is computed for the proposed work for dataset 1 and dataset 2. The results acquired are manifested in Table 12. This evaluation has been done at 70th TP for dataset 1 and dataset 2. On observing the outcomes, the proposed had attained the highest value under all the considered performance. The mean performance recorded by the dataset 1 is 91.4703, which is 8.37%, 20.70%, 7.66%, 7.7%, 7.58%, 6.79%, 7.63%, 5.68%, 8% and 7.95% better than the existing models like DBN, SVN,NN, NB, LSTM, CNN, GRU, English legal judgment prediction [28], M-AttBLSTM-CNN [19], two neural network architectures [30], respectively. When compared to other conventional approaches, dataset 2's mean performance is excellent. The Self-Improved Recurrent Neural Network (SI-RNN) and Bi-LSTM to create the Modified Hierarchical Attention Network (M-HAN), which has an improved cosine similarity-based sentiment polarity feature

Table 9
Analysis of the proposed hybrid classifier for varying Test Percentage in terms of Precision for dataset 2.

Training percentage(%)	DBN	SVM	NN	NB	LSTM	CNN	GRU	English legal judgment prediction [2]	M-AttBLSTM-CNN [28]	Two neural network architectures [4]	Hybrid classifier
60	78.98218	74.3323	75.96456	81.30082	79.02779	82.31379	95.933	68.61652	74.90205	72.09122	97
70	84.04745	74.53779	80.84481	82.40318	81.76662	83.50989	95.933	89.37969	80.9414	83.0637	97
80	87.09308	74.8214	85.03276	86.30284	85.65256	90.33602	95.933	92.50879	85.69812	89.57938	97
90	91.52437	75.65367	90.46515	87.59824	90.7438	90.8379	95.933	95.20963	90.92095	90.00131	97

Table 10
Analysis of the proposed hybrid classifier for varying Test Percentage in terms of F-measure for dataset 2.

Training percentage(%)	DBN	SVM	NN	NB	LSTM	CNN	GRU	English legal judgment prediction [2]	M-AttBLSTM-CNN [28]	Two neural network architectures [4]	Hybrid classifier
60	74.15156	74.71673	76.29774	69.23594	72.71273	75.33158	72.99626	80.23047	74.95701	78.35056	87.3548
70	80.1743	74.94651	80.93374	73.11263	79.47882	82.70476	79.78873	86.6056	81.11929	81.47887	88.984
80	84.2538	75.08552	85.49844	80.05098	85.60442	87.67519	85.89481	89.12947	85.69812	85.30831	91.21092
90	90.02173	75.28817	90.13182	84.19053	90.62829	91.79478	90.86338	92.47703	90.75375	91.35228	93.98776

Table 11
Analysis of the proposed hybrid classifier for varying Test Percentage in terms of MCC for dataset 2.

Training percentage(%)	DBN	SVM	NN	NB	LSTM	CNN	GRU	English legal judgment prediction [2]	M-AttBLSTM-CNN [28]	Two neural network architectures [4]	Hybrid classifier
60	66.93009	65.34409	66.95028	73.09538	70.94986	69.12326	71.22652	77.17418	67.05052	67.60273	78.19998
70	74.34991	65.82933	76.01145	76.28436	77.76156	77.65126	77.81411	77.49953	75.89358	76.10153	81.54012
80	76.1771	77.34822	76.48502	77.3936	78.78211	78.73363	79.02897	82.21319	76.26923	76.68245	86.05704
90	85.2647	77.89134	85.41421	80.08846	86.34788	87.35543	86.49169	88.5425	86.39713	86.64575	91.63866

and a hybrid deep classifier for prediction. This improvement is owing to the M-HAN, extracted eight features, and polarity level score prediction with improved cosine similarity.

7.5. Ablation test for proposed work with M-HAN: with and without hybrid classifier for dataset 1 and dataset 2

The ablation test is conducted for the proposed work with M-HAN output and proposed work with M-HAN+Hybrid classifier output for dataset 1 and dataset 2. The outcomes acquired are manifested in Table 13. The accuracy of the proposed work with M-HAN is 4% better than the accuracy of the proposed work with the M-HAN+Hybrid classifier. In addition, the proposed work with M-HAN has achieved the highest precision compared to HAN + Hybrid classifier. For dataset 2, the accuracy of the proposed work is high when compared to the HAN features. Therefore, from the evaluation, it is vivid that M-HAN has enhanced the performance of the proposed work in making legal judgments.

7.6. Ablation test of proposed work: HAN vs M-HAN for dataset 1 and dataset 2

The outcome from the proposed work with HAN and with M-HAN for dataset 1 and dataset 2 is evaluated, and the results acquired are tabulated in Table 14. The accuracy of the proposed work with M-HAN is 95%, which is the best value when compared to the proposed work with HAN feature for dataset 1. For dataset 2, the accuracy is high for a proposed model when compared to the HAN features.

8. Conclusion

This paper introduced a new judgment prediction model with 3 major phases: (a) pre-processing, (b) feature extraction, and (c) judgment prediction phase. The acquired raw data was initially tokenized at both the sentence and word levels during the pre-processing phase. The features were then extracted from these tokens using the M-HAN-based features and the proposed improved cosine similarity-based methods. The hybrid classifier constructed in the judgment prediction framework was trained using the extracted sentiment polarity characteristics. In the end, a hybrid classifier is used to model the judgment prediction stage. The suggested hybrid classifier combines the SI-RNN and Bi-LSTM algorithms. The proposed hybrid classifier's output displays the

Table 12
Statistical analysis of the proposed hybrid classifier.

Dataset 1											
Performance	DBN	SVM	NN	NB	LSTM	CNN [33]	GRU	English legal judgment prediction [28]	M-AttBLSTM-CNN [19]	two neural networkarchitectures [30]	Hybrid classifier
Mean	84.4065	75.7833	84.9669	84.9313	85.0202	85.6518	84.9847	86.5591	84.6911	84.7356	91.4703
Median	84.3976	75.8487	84.9669	84.7178	85.1092	85.6073	85.0914	88.2759	84.9669	84.6466	91.2004
Standard deviation	5.4919	0.2049	5.2910	5.3727	5.1646	5.0644	5.1251	5.8974	5.9420	5.4093	2.8606
Worst	76.8900	75.4630	77.8507	78.2777	78.0286	78.9181	78.0286	76.9612	76.3563	77.4593	87.9266
Best	91.9407	75.9725	92.0830	92.0119	91.8340	92.4744	91.7272	92.7235	92.4744	92.1898	95.5536
Dataset 2											
Performance	DBN	SVM	NN	NB	LSTM	CNN [33]	GRU	English legal judgment prediction [28]	M-AttBLSTM-CNN [19]	two neural networkarchitectures [30]	Hybrid classifier
Mean	82.71833	74.7162	83.18255	77.8734	84.16238	84.97925	84.45519	86.44138	83.12263	83.8806	89.9152
Median	82.7096	74.7756	83.18255	77.6777	84.25043	84.935	84.5612	88.1558	83.3932	83.79255	89.6500
Standard deviation	5.3820	0.1858	5.17984	4.92622	5.11246	5.02461	5.0932	5.8893	5.8319	5.35472	2.8119
Worst	75.3522	74.4257	76.2158	71.772	77.2413	78.29846	77.5424	76.8565	74.942	76.67775	86.4318
Best	90.1018	74.88794	90.1492	84.3656	90.9073	91.74829	91.15574	92.5973	90.7617	91.2595	93.9291

Table 13
Ablation test for proposed work with M-HAN: with and without hybrid classifier for dataset 1 and dataset 2.

Dataset 1		
Metrics	SI-RNN+Bi-LSTM with HAN features	SI-RNN+Bi-LSTM with M-HAN features
Accuracy	0.883107	0.90641
Precision	0.928098	0.949282
F1-score	0.874933	0.898997
MCC	0.807874	0.843241
Dataset 2		
Metrics	SI-RNN+Bi-LSTM with HAN features	SI-RNN+Bi-LSTM with M-HAN features
Accuracy	0.866513	0.884081
Precision	0.866513	0.898038
F1-score	0.866513	0.883294
mcc	0.771175	0.836618

Table 14
Ablation test for proposed work with HAN and M-HAN.

Dataset 1		
Metrics	HAN	MHAN features
Accuracy	0.869325	0.90641
Precision	0.933585	0.949282
F1-score	0.920729	0.898997
MCC	0.551093	0.843241
Dataset 2		
Accuracy	0.879272	0.884081
Precision	0.880814	0.898038
F1-score	0.879194	0.883294
mcc	0.836988	0.836618

expected results. The pre-trained model is used to anticipate the associated legal judgments based on the testing data, which is provided as input during the testing phase. The evaluation is mainly focused on the prediction measures like "accuracy, precision F_1-score, and MCC", and statistical performance, respectively. By investigating the automated examination of other resources we hope to broaden the study's focus. These resources might then be used in a multi-input manner to enhance performance and support system choices. Additionally, we intend to use data from other courts and neural approaches. In future work, investigating the automated examination of other resources we hope to broaden the study's focus. These resources might then be used in a multi-input manner to enhance performance and support system choices. Additionally, we intend to use data from other courts and neural approaches.

Findings: The accuracy of the proposed work at TP = 90 is 14%, 16.89%, 18.93%, 15.58%, 6.75%, 19.30%, 20.43%, 14.30%, 17.67% and 19,82% better than the existing models like DBN, SVN,NN, NB, LSTM, CNN, GRU, English legal judgment prediction [28], M-AttBLSTM-CNN [19], two neural network architectures [30], respectively. Moreover, the specificity, sensitivity and precision of the proposed work had also attained the highest value at every variation in TP.

CRediT authorship contribution statement

G. Sukanya: Conceptualization, Methodology, Resources, Data curation. **J. Priyadarshini:** Formal analysis, Investigation.

Declaration of competing interest

The authors declare that they have no conflict of interest.

Funding

This research did not receive any specific funding.

References

[1] C. Jin, G. Zhang, M. Wu, S. Zhou, T. Fu, Textual content prediction via fuzzy attention neural network model without predefined knowledge, China Commun. 17 (6) (2020) 211–222.
[2] T. Goto, K. Sano, S. Tojo, Modeling predictability of agent in legal cases, in: 2016 IEEE International Conference on Agents, ICA, Matsue, 2016, pp. 13–18.
[3] Y. Yin, F. Zulkernine, S. Dahan, Determining worker type from legal text data using machine learning, in: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, DASC/PiCom/CBDCom/CyberSciTech, Calgary, AB, Canada, 2020, pp. 444–450.
[4] K. Kowsrihawat, P. Vateekul, P. Boonkwan, Predicting judicial decisions of criminal cases from thai supreme court using bi-directional GRU with attention mechanism, in: 2018 5th Asian Conference on Defense Technology, ACDT, Hanoi, 2018, pp. 50–55.
[5] K. Zhu, B. Ma, T. Huang, Z. Li, H. Ma, Y. Li, Sequence generation network based on hierarchical attention for multi-charge prediction, IEEE Access 8 (2020) 109315–109324.
[6] A.B. Calgary, L. Yuan Canada, et al., Automatic legal judgment prediction via large amounts of criminal cases, in: 2019 IEEE 5th International Conference on Computer and Communications, ICCC, Chengdu, China, 2020, pp. 2087–2091, 361–365.
[7] B. Chen, Y. Li, S. Zhang, H. Lian, T. He, A deep learning method for judicial decision support, in: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion, QRS-C, Sofia, Bulgaria, 2019, pp. 145–149.
[8] J. Guo, B. Wu, P. Zhou, BLHNN: A novel charge prediction model based on bi-attention BI-LSTM-CNN hybrid neural network, in: 2020 IEEE Fifth International Conference on Data Science in Cyberspace, DSC, Hong Kong, Hong Kong, 2020, pp. 246–252.
[9] Chang Yin, Cuiqing Jiang, Zhao Wang, Evaluating the credit risk of SMEs using legal judgments, Decis. Support Syst. (2020).
[10] D. Huang, W. Lin, A model for legal judgment prediction based on multi-model fusion, in: 2019 3rd International Conference on Electronic Information Technology and Computer Engineering, EITCE, Xiamen, China, 2019, pp. 892–895.
[11] S. Li, B. Liu, L. Ye, H. Zhang, B. Fang, Element-aware legal judgment prediction for criminal cases with confusing charges, in: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence, ICTAI, Portland, OR, USA, 2019, pp. 660–667.
[12] L. Chen, N. Xu, Y. Wang, Legal judgment prediction with label dependencies, in: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, DASC/PiCom/CBDCom/CyberSciTech.
[13] R. Sil, A. Roy, A novel approach on argument based legal prediction model using machine learning, in: 2020 International Conference on Smart Electronics and Communication, ICOSEC, Trichy, India, 2020, pp. 487–490.
[14] X. Yang, G. Shi, J. Lou, S. Wang, Z. Guo, Interpretable charge prediction with multi-perspective jointly learning model, in: 2019 IEEE 5th International Conference on Computer and Communications, ICCC, Chengdu, China, 2019, pp. 1850–1855.
[15] V.G. Pillai, L.R. Chandran, Verdict prediction for Indian courts using bag of words and convolutional neural network, in: 2020 Third International Conference on Smart Systems and Inventive Technology, ICSSIT, Tirunelveli, India, 2020, pp. 676–683.
[16] Virgilio Galdo, Yue Li, Rama Martin, Identifying urban areas by combining human judgment and machine learning: An application to India, Journal of Urban Economics 125 (2021) 103229.
[17] Rafe Athar Shaikh, Tirath Prasad Sahu, Veena Anand, Predicting outcomes of legal cases based on legal factors using classifiers, Procedia Comput. Sci. (2020).
[18] Tom R. Kupfer, Yoel Inbar, Joshua M. Tybur, Reexamining the role of intent in moral judgements of purity violations, J. Exp. Soc. Psychol. (2020).
[19] S. Li, B. Liu, L. Ye, H. Zhang, B. Fang, Element-aware legal judgment prediction for criminal cases with confusing charges, in: IEEE 31st International Conference on Tools with Artificial Intelligence, ICTAI, Portland, OR, USA, 2019, pp. 660–667.
[20] Sajad Mousavi, Fatemeh Afghah, U. Rajendra Acharya, HAN-ECG: An interpretable atrial fibrillation detection model using hierarchical attention networks, Comput. Biol. Med. (2020).
[21] Fa Li, Zhipeng Gui, Yichen Lei, A hierarchical temporal attention-based BI-LSTM encoder–decoder model for individual mobility prediction, Neurocomputing (2020).
[22] Yirong Zhou, Jun Li, Luo Chen, A spatiotemporal hierarchical attention mechanism-based model for multi-step station-level crowd flow prediction, Inform. Sci. (2021).
[23] Shuning Xing, Fang'ai Liu, Tianlai Li, A hierarchical attention model for rating prediction by leveraging user and product reviews, Neurocomputing (2019).
[24] Jingzhe Liu, Peng Wu, Penalty prediction based on modulated hierarchical attention coupled with legal attribute recognition, in: 2022 IEEE International Conference on Big Data, Big Data, IEEE, 2022.

[25] Yao Chi, et al., Legal judgement prediction of sentence commutation with multi-document information, in: Computer Supported Cooperative Work and Social Computing: 16th CCF Conference, ChineseCSCW 2021, Xiangtan, China, November 26–28, 2021, Revised Selected Papers, Part I, Springer Nature Singapore, Singapore, 2022.
[26] Prameela Madambakam, Shathanaa Rajmohan, A study on legal judgment prediction using deep learning techniques, in: 2022 IEEE Silchar Subsection Conference, SILCON, IEEE, 2022.
[27] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liuy, Maosong Sun, Legal judgment prediction via topological learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3540–3549.
[28] Ilias Chalkidis, Ilias Chalkidi, Nikolaos Aletras, Legal judgment prediction in english, 2019, arXiv:1906.02059v1 [cs.CL].
[29] Shangbang Long, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Automatic judgment prediction via legal reading comprehension, 2018, arXiv:1809.06537v1 [cs.AI].
[30] Deepa Anand, Rupali Wagh, Effective deep learning approaches for summarization of legal texts, J. King Saud Univ. Comput. Inf. Sci. (2019).
[31] Xiaoding Guo, Hongli Zhang, Lin Ye, Shang Li, TenLa: an approach based on controllable tensor decomposition and optimized lasso regression for judgement prediction of legal cases, Appl. Intell. (2020).
[32] X. Guo, H. Zhang, L. Ye, S. Li, G. Zhang, TenRR: An approach based on innovative tensor decomposition and optimized ridge regression for judgment prediction of legal cases, IEEE Access 8 (2020) 167914–167929, http://dx.doi.org/10.1109/ACCESS.2020.2999522.
[33] Baogui Chen, Yu Li, Shu Zhang, Hao Lian, A deep learning method for judicial decision support, in: IEEE 19th International Conference on Software Quality, Reliability and Security Companion, QRS-C, 2019.
[34] Wenmian Yang, Wenmian Yang, Xiaojie Zhou, Yutao Luo, Legal judgment prediction via multi-perspective bi-feedback network, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, 2019.
[35] Sivaram Rajeyyagari, Automatic speaker diarization using deep BI-LSTM in audio lecturing of e-Khool Platform, J. Netw. Commun. Syst. 3 (4) (2020).

Sukanya G. is from Chennai India and pursuing Ph.D in VIT University in the field of NLP and Law. The career objective of the author is to work for the development of digital assistance for litigants in India in general. She has completed bachelor's degree in Computer Science and Engineering at Madras University and Master's Degree in Anna University, Chennai. The author has attended many national and international conferences and has written survey papers on legal judgment prediction system.

Dr. J. Priyadarshini is currently Professor at school of Computer Science and Engineering, VIT University, Chennai, TamilNadu, India. She obtained her Ph.D in Computer Science at Anna University, MIT campus, Chennai in 2014. Her areas of interests are Artificial Intelligence, Image Processing and Natural Language Processing. Also she has published articles in various national and International conferences and journals.