LegalHANOpt: A Hierarchical Attention Network with BOHB Optimization for Predicting and Explaining Legal Case Decisions

Aijun Wang
School of Economics and Management, Jining university, Qufu, Shandong, 273155, China
E-mail: wangaijun188@outlook.com

Keywords: legal case prediction, hierarchical attention network, explainable AI (XAI), Bayesian Optimization, hyperband, legal document analysis, judicial decision support

Received: July 28, 2025

Legal documents are often lengthy and complex, making it challenging and time-consuming for experts to accurately predict case outcomes. Older methods are not well-suited to the structure and language used in legal texts. This paper proposes a model called LegalHANOpt (Legal Hierarchical Attention Network with Optimized Parameters) to make accurate predictions about legal case decisions and explain how those decisions are made. LegalHANOpt utilizes a Hierarchical Attention Network (HAN) that analyzes legal documents by examining individual words and sentences, much like lawyers typically do. To improve the model's performance, Bayesian Optimization with Hyperband (BOHB) is utilized. This sophisticated method automatically determines the optimal settings for training the model, such as learning rate and dropout. The LegalHANOpt is trained on an extensive collection of past legal cases, including relevant facts, laws, and decisions. Results show that LegalHANOpt gives more accurate predictions than older methods achieving superior performance with an accuracy of 0.91%, macro F1-score of 0.83%, and AUC-ROC of 0.92%. It also highlights essential parts of the text, helping users understand why the model made a particular decision. In short, LegalHANOpt is a valuable and explainable tool to support legal experts in making better and faster decisions.

Povzetek: Prispevek predstavi razložljiv model LegalHANOpt (hierarhična pozornost + BOHB optimizacija), ki na podlagi pravnih dokumentov natančneje napoveduje sodne odločitve in hkrati označi ključne dele besedila, ki utemeljujejo napoved.

1 Introduction and related works

Judicial law cases are binding rulings that courts deliver upon consideration of legal submissions, facts, and applicable legislation [1]. Legal conclusions play a crucial role in the formation of judicial precedents and influence subsequent decisions [2]. The use of AI in the legal community has revolutionized legal processing, access, and analysis. As digital law information spreads more widely, AI technologies are also utilized for automating mundane tasks, such as contract examination, legal research, and case outcome prediction [3]. These technologies provide greater efficiency, accuracy, and access to the law, enabling professionals to make better decisions while being relieved of the laborious task of hand-reviewing documents [4]. Despite this advancement, legal documents are complicated by the field's specific language, hierarchical structure, and conditional meanings [5]. Conventional machine learning processes tend not to perform as subtly, and therefore they have low predictive strength and generalizability [6]. This has led to the emergence of deep models, most of which use the attention mechanism and hierarchical processing as first-line approaches to improve legal text comprehension. Nevertheless, they should address the model interpretability problem, and optimization problems are critical in high-risk real-world applications [7].

Deep models have outperformed in natural language processing applications such as sentiment analysis, machine translation, and text classification [8]. Deep learning models can be utilized to learn patterns and make predictions based on extensive historical case data [9]. Architectures such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformer-based architectures have recently shown very promising performances in most legal text classification tasks. Legal texts differ from regular language in that they are formal, multi-level in structure, and grounded in legal precedents and citations [10].

Precedent cases help lawyers forecast legal outcomes. Due to the rising amount and complexity of legal documents, human interpretation and prediction are inconvenient and difficult. Surface-level learning systems struggle to understand legal documents' hierarchical and contextual character, limiting predictive ability. In the legal world, openness is essential, and most modern deep learning models are black boxes with minimal explainability. Such models' hyperparameters must be manually tweaked, which is computationally and time-consuming. Unreliable model tuning can cause overfitting or failure to generalize, rendering the model ineffective. This study addresses the requirement for a prediction model that can accurately predict legal cases and explain their decision-making motives. Creating a system that respects the law's semantic and structural richness, simplifies interpretation, and can be automatically modified for optimal prediction performance is difficult. Thus, forecasting legal outcomes with an accurate, interpretable, and computationally efficient model is the main research challenge. The LegalHANOpt processes words and sentences using a Hierarchical Attention Network (HAN) to reflect legal document organization. Such architecture lets the model catch the most important text elements and make an informed judgment. Bayesian Optimization with Hyperband (BOHB) finds the appropriate learning rate, dropout rate, and attention size to improve model prediction. The model is trained utilizing a huge annotated legal corpus comprising factual descriptions, pertinent legislation, and court judgments. Attention visualizations help people understand model outputs and monitor reasoning steps for each prediction.

1.1 Related works

Sil & Roy introduced a machine learning-based legal document classification method [11]. Support vector machines and decision trees were chosen because they excel in structured classification. The model classified legal materials moderately well. Its inability to handle semantic meaning and structural complexity limited its use in more complex legal reasoning tasks like outcome prediction and explanation. Shang presented a computational intelligence approach for legal prediction and assistance [12]. Fuzzy logic and rule-based systems addressed legal document subjectivity and ambiguity. System consistency and forecast accuracy were good. Scalability was its key shortcoming since manually written rules made it hard to respond to complex legal scenarios. ILDC for CJPE is an Indian legal corpus built by Malik et al. [13] to interpret and forecast judgments. Deep learning models BERT and BiLSTM were utilized for semantic representation. The models explained and predicted court rulings well. The dataset imbalance and low performance in underrepresented case categories limited generalization.

Niklaus et al. [14] recommended the Swiss-decision-Prediction dataset for multilingual court decision prediction. Transformer models handled contextual representation and multilingual input. These models performed well in Swiss legal texts. Variability across jurisdictions and languages reduces accuracy, making it difficult to forecast how legal concepts would align across languages. Zafar [15] examined the ethical and practical issues of AI in legal decision-making. A conceptual and analytical approach assessed AI's implications on bias, transparency, and legal justice. The findings stressed legal AI responsibility. The study lacked empirical confirmation and focused more on theoretical difficulties than proven models or quantitative data. Lam et al. [16] created a legal analytics algorithm to predict employment law notification durations. Regression models with NLP preprocessing collected valuable contract data. The method yielded accurate legal forecasts. The short sample size and difficulty modeling court ruling exceptions, which sometimes involve complicated, case-specific aspects, were limitations.

Safat et al. [17] suggested a deep learning-machine learning hybrid crime forecasting technique. We used Random Forest, CNN, and LSTM models since they work well with temporal and geographical data. Crime patterns were accurately predicted by the system. Its focus on criminal events rather than court judgment results limits its usefulness to legal decision prediction tasks. Kumar proposed a machine learning-based legal outcome forecasting methodology [18]. Decision trees, logistic regression, and ensemble techniques were used for interpretability and computing efficiency. The algorithms' prediction performance was good across legal datasets. They lacked a deep comprehension of legal language's contextual meaning, making them less capable of comprehending complex legal papers or reasoning. Abimbola et al. [19] developed a CNN-LSTM hybrid model for legal document sentiment analysis. CNNs recorded spatial patterns, LSTMs sequential data dependencies. On legal datasets, the model classified sentiment accurately. Its key limitations were weak explainability and poor performance on lengthy, complex legal texts with layered phrase structures.

Anand & Wagh proposed deep learning for legal text summarization [20]. Long legal narratives were recorded and condensed using attention-based encoder-decoder designs. The results showed that the model produced logical summaries with legal details. Poor performance on multi-topic or lengthy court matters and difficulty managing complex rhetorical patterns were limitations. Yue et al. presented NeurJudge, a neural framework for situation-aware legal judgment prediction [21]. The model used contextual embeddings and legal knowledge graphs to increase legal comprehension. It outperformed benchmarks in case categorization and decision prediction. Structured knowledge increases system complexity, making it difficult to scale and generalize across legal systems or jurisdictions. Deep learning-based Arabic text court judgment support system TaSbeeb was developed by Almuzaini & Azmi [22] using LSTM networks. Because it records legal sequences and dependencies, LSTM was chosen. The model predicted verdict categorization accurately. Using language-specific tuning reduced cross-lingual transferability and accuracy in ambiguous legal scenarios.

Deep neural network-based AI-assisted model for court decision prediction by Ma [23]. Fully connected layers handled high-dimensional lawful feature inputs. The model performed well on binary legal tasks. However, its inability to handle complex or multiclass rulings and lack of transparency raised concerns about its reliability in high-stakes legal applications. Medvedeva et al. rigorously evaluated automated court decision prediction

[24]. Statistics and deep learning were tested for real-world applicability. Results suggested that fairness and interpretability in legal situations are usually disregarded in favor of high accuracy metrics. The field's downside was its over-reliance on quantitative performance above ethical and legal thought.

Morić et al. [25] developed a judicial decision support system based on Bayesian Networks, which enables probabilistic decision-making and uncertainty modeling. By improving transparency and systematic inference, the study also demonstrated that it was difficult to integrate legal knowledge and maintain ethical, interpretable AI behavior within the context of actual judicial decision-making. A model of online-judicial-public-opinion control and intelligent decision-making was proposed by Guo [26] using Bi-LSTM. The method was helpful in terms of capturing the temporal dynamics of legal discourse, but encountered problems related to the quality of data and cross-jurisdiction generalization as well as demanded more interpretable reasoning processes to be used in actual legal contexts. Wang et al. [27] proposed LegalReasoner, which integrates large language models with formal legal knowledge to improve the prediction of legal judgments. Despite being more accurate and able to reason contextually, the framework was challenged on knowledge incorporation, computational costs, and the production of legally relevant, explainable results for practical use in courts. Table 1 shows the comparative overview of model capabilities.

The primary significance of the paper is

- To develop a deep learning model tailored for hierarchical legal text analysis using a Hierarchical Attention Network.
- To integrate Bayesian Optimization with Hyperband for efficient and automated hyperparameter tuning.
- To enhance interpretability through attention mechanisms that highlight critical parts of the legal text.
- To evaluate the model on real-world legal datasets with performance comparison against traditional and modern baselines.
- To provide a scalable and transparent decision-support system for legal professionals and judicial institutions.

Table 1: Comparative overview of model capabilities

Ref	Dataset	Model Type	Accuracy	F1-Score	Key Limitation
[11]	Legal document classification dataset	SVM, Decision Tree	0.70	0.68	Shallow models; cannot capture semantic or hierarchical context
[13]	35K Indian legal cases	BERT, BiLSTM	0.78	0.76	Class imbalance; limited explanation capability
[14]	Swiss multilingual legal cases	Transformer models	0.78	0.74	Cross-lingual variability reduces generalization
[21]	Criminal legal case dataset	Knowledge Graph + Neural Model	0.82	0.80	Heavy model complexity; difficult to scale across jurisdictions
[22]	Arabic judicial decisions	LSTM classifier	0.82	0.78	Language-specific tuning; limited cross-domain transfer
[19]	Legal sentiment analysis dataset	CNN–LSTM Hybrid	0.84	0.79	Weak performance on long multi-sentence legal documents

1.2 Research questions

RQ1: Does the integration of Bayesian Optimization with Hyperband (BOHB) significantly improve model performance compared to manual or conventional hyperparameter tuning in legal judgment prediction tasks?

RQ2: Can a Hierarchical Attention Network (HAN) effectively capture both word-level and sentence-level legal semantics to outperform existing deep learning baselines (e.g., BERT/BiLSTM, NeurJudge, TaSbeeb)?

RQ3: Do dual-level attention layers provide meaningful, human-interpretable explanations that align with legally significant sentences and phrases?

2 LegalHANOpt model structure

This section outlines the methodology of the LegalHANOpt model to enhance prediction accuracy and interpretability in legal case verdicts. LegalHANOpt utilizes the hierarchical nature of legal documents by parallel processing word-level and sentence-level information using an HAN. Bidirectional GRUs acquire contextual information per level, and attention mechanisms select the most relevant words and sentences that determine the judgment. The model also hyperparameter-tunes learning rate, dropout rate, GRU units, and batch size with Bayesian Optimization with Hyperband (BOHB) for further tuning. The entire methodology encompasses data preprocessing steps, hierarchical feature generation, model training, prediction creation, and explanation through attention. Figure 1 shows the Architecture of the LegalHANOpt model.

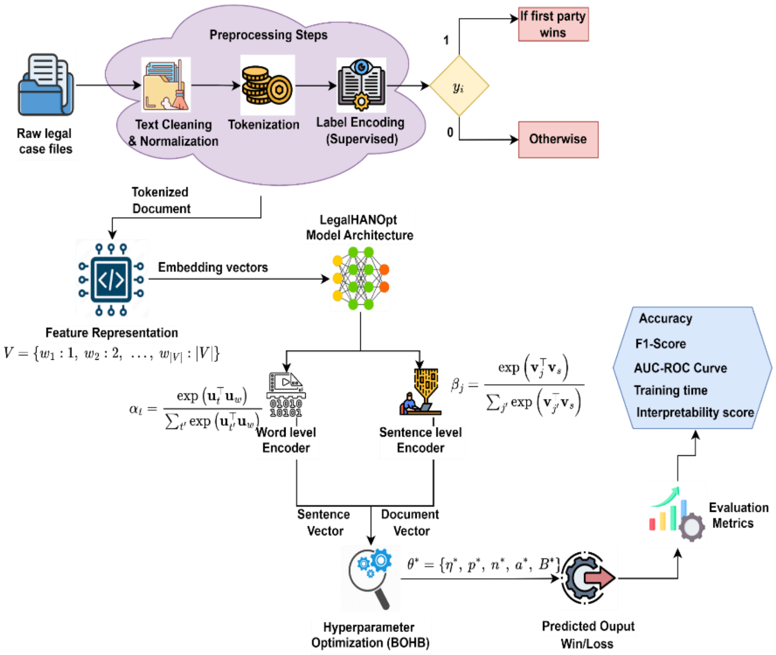
Preprocessing Steps
Raw legal case files
Text Cleaning & Normalization
Tokenization
Label Encoding (Supervised)
y_i
1
If first party wins
0
Otherwise
Tokenized Document
LegalHANOpt Model Architecture
Embedding vectors
Feature Representation
$V = \{w_1 : 1,\ w_2 : 2,\ \dots,\ w_{|V|} : |V|\}$
$\alpha_t = \frac{\exp(\mathbf{u}_t^\top \mathbf{u}_w)}{\sum_{t'} \exp(\mathbf{u}_{t'}^\top \mathbf{u}_w)}$
Word level Encoder
Sentence level Encoder
$\beta_j = \frac{\exp(\mathbf{v}_j^\top \mathbf{v}_s)}{\sum_{j'} \exp(\mathbf{v}_{j'}^\top \mathbf{v}_s)}$
Accuracy
F1-Score
AUC-ROC Curve
Training time
Interpretability score
Sentence Vector
Document Vector
Evaluation Metrics
$\theta^* = \{\eta^*,\ p^*,\ n^*,\ a^*,\ B^*\}$
Hyperparameter Optimization (BOHB)
Predicted Ouput Win/Loss

Figure 1: Architecture of the LegalHANOpt model

2.1 Data preprocessing and collection

Preparation and collection of the legal text data are the initial steps of LegalHANOpt to make it suitable for deep learning modeling. Case files, i.e., fact descriptions, statute citations, and outcomes of judgments, are gathered from public sources in the Supreme Court Judgment Prediction dataset [28]. Case documents contain a narrative component (case facts) and an associate label about the outcome of the court judgment.

2.1.1 Text cleaning and normalization

To preprocess the text to eliminate noise and normalize the input text, a series of text preprocessing steps is employed:

✓ All characters converted to lowercase in order not to be case-sensitive.

✓ Special character removal (e.g., punctuation marks, symbols) to eliminate noisy patterns.

✓ Stopword removal, where common words (e.g., "the", "is", "and") that do not contribute significantly to judgment prediction are eliminated.

✓ Lemmatization may be optionally employed to convert inflected words to their base or dictionary form (e.g., "arguing" → "argue").

Let $D = \{d1, d2, \dots, dN\}$ represent the corpus of legal documents, where each d_i is a raw text document. After cleaning, each d_i is converted into a sequence of sentences $S_i = \{s_1, s_2, \dots, s_T\}$, and each sentence s_j into a sequence of words $W_j = \{w_1, w_2, \dots, w_L\}$, preserving the hierarchical structure, $di \rightarrow \{s1, s2, \dots, sT\}, sj \rightarrow \{w1, w2, \dots, wL\}$. This structured form is necessary for input into the Hierarchical Attention Network (HAN), which processes text at both word and sentence levels.

2.1.2 Label encoding supervised learning

All legal cases carry a result label yi, which is the outcome of the verdict. For tasks like "First Party Winner" prediction, target labels are represented in binary format as in equation (1):

$$y_i = \begin{cases} 1, & if\ the\ first\ party\ (plaintiff/appellant)\ wins \\ 0, & otherwise \end{cases} \quad (1)$$

This reduces the task of prediction to a binary classification task for training the network's output layer using cross-entropy loss as shown in equation (2):

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i log\hat{y}_i + (1 - y_i)log(1 - \hat{y}_i)] \quad (2)$$

where $\hat{y}_i$ is the predicted probability by the model for the positive class (first party wins).

2.1.3 Tokenization and vocabulary construction

Following the hierarchical organization and cleaning of the legal text, every sentence is tokenized into individual words or subwords based on the embedding model used. In this paper, word-level tokenization has been employed such that every sentence (s_j) in the document d_i is broken as $s_j = \{w_1, w_2, \dots, w_n\}$. A fixed vocabulary V is then built by gathering all the distinct tokens in the corpus and assigning to every token $w \in V$ a distinct index such that $V = \{w_1: 1, w_2: 2, \ldots., w_{|V|}: | V |\}$. This allows tokenized sequences to be mapped into indexed numerical sequences, for instance, $\{w1, w2, w3\} \rightarrow \{1,5,17\}$. These indices are then each used to look up a dense vector in an embedding lookup from a pre-trained embedding matrix $E \in R^{|V| \times d}$, where d is the embedding dimension. A sequence $[w_1, w_2, ., w_n]$ thus becomes $[\vec{e}_1, \vec{e}_2, ., \vec{e}_n]$, with each $\vec{e}_i \in \mathbb{R}^d$. To ensure equal input dimensions for the neural network, all sequences are padded with a special $< PAD >$ token or truncated to size L with a tensor shape of (B, T, L, d), where B is the batch size, T is sentences per document, and L is words per sentence. This organized input is subsequently presented as a hierarchical attention architecture, preserving word-level and sentence-level semantics pertinent to legal judgment prediction.

Once each sentence is encoded and filtered through attention mechanisms, it is then summarized as a fixed-length sentence vector $\vec{s}_j \in \mathbb{R}^h$, where h is the hidden size of the encoder. The document is then considered to be a sequence of such sentence vectors $D_i = [\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_T] \in \mathbb{R}^{T \times h}$. This input is passed to the sentence-level encoder, which absorbs the document's context. Sentence-level attention is subsequently used to weight each sentence according to how relevant it is to the final decision, resulting in a final document vector $\vec{d}_j \in \mathbb{R}^h$. The hierarchical process from word embeddings to document vectors retains both the local linguistic features (local to the word) and global contextual features (global to the sentence). This architecture is well-suited for legal documents, where certain words and phrases carry disproportionate legal weight.

2.1.4 Textual preprocessing pipeline

A uniform and reproducible pipeline was followed in all the preprocessing steps. Segmentation of the sentence was done with the en_core_web_sm model of spaCy that is known to split long legal texts into structured sentences. The tokenizer used was also spaCy at the word-level, where there were regular token boundaries and treatment of punctuation. Everything was in lower case and the number patterns were brought to the standard forms. A custom-built legal stopword list was used to apply domain-specific stopwords with 146 words (e.g., petitioner, respondent, bench, learned counsel) that was compiled based on the existing legal NLP literature and through an inspection of their manuals to eliminate lexical items that were not informative. There were rare tokens that did not appear 3 times, so they were eliminated and the vocabulary was limited to 50, 000 tokens. It did not use any subword tokenization (e.g. WordPiece/BPE) because the model uses standard word-level embeddings. To make sure that all the preprocessing scripts behave deterministically, they were all implemented through spaCy v3.7.

2.2 Feature representation

The accurate legal text emulation of LegalHANOpt is based on converting unstructured text data into semantically rich vector representations suitable for hierarchical neural network processing. The conversion closes the gap between legible legal text and machine-readable numerical formats.

2.2.1 Pre-trained legal text embedding

For encoding individual words as points in a high-dimensional semantic space, the paper employs pre-trained word embeddings, such as GloVe, which map the distributional properties of language from word co-occurrence in large text corpora. For a vocabulary V, each word $w \in V$ is associated with a distinct continuous vector $\vec{e}_w \in \mathbb{R}^d$ where d is the size of the embedding (typically 100–300 dimensions). These vectors capture semantic similarity—words with similar meanings are encoded as vectors close to one another in Euclidean or cosine space.

Each sentence s_j composed of L words is represented as an embedding matrix $S_j \in \mathbb{R}^{L \times d}$. This is passed to the word-level encoder, which learns contextualized word representations. Likewise, the resulting vectors sj from word-level encodings are pooled to provide a document-level matrix. $D_i \in \mathbb{R}^{T \times h}$, with T sentences and h being the sentence embedding space dimension.

2.2.2 Hierarchical feature modeling

Legal papers have intra-sentence and inter-sentence links, which this hierarchical organization preserves. This arrangement allows the model to master localized lexical patterns and discourse flow simultaneously, unlike flat designs, which consider the content as a single sequence of tokens.

The model determines word-level syntactic features like legal jargon and negations ("not liable," "granted bail"). Causality, juridical precedence, and procedural activities are determined at the sentence level. The multi-granularity of this depiction is vital in juridical reports, where each clause can considerably affect the final decision.

2.3 Model architecture – hierarchical attention network (HAN)

The LegalHANOpt model uses a Hierarchical Attention Network (HAN) to read legal documents as they appear linguistically—words organized into sentences and sentences organized into documents. In contrast to sequence models with flat input, HAN employs two levels of attention-based encoding: word-level and sentence-level, allowing the model to capture both local and global context effectively.

2.3.1 Word encoder with attention

Each sentence $s_j = \{w_1, w_2, \ldots, w_L\}$ of a text is initially converted to a matrix $S_j \in \mathbb{R}^{L \times d}$, where d is the dimension for word embeddings. This matrix is subsequently input into a Bidirectional Gated Recurrent Unit (BiGRU) to capture contextual dependencies in both directions, as shown in equation 3.

$$\left.\begin{aligned} h_t &= [\vec{h}_t; \overleftarrow{h}_t] \\ \vec{h}_t &= GRU_{fw}(\vec{e}_t) \\ \overleftarrow{h}_t &= GRU_{bw}(\overleftarrow{e}_t) \end{aligned}\right\} \quad (3)$$

Here, $\vec{e}_t \in \mathbb{R}^d$ is the embedding of a word t, and $h_t \in \mathbb{R}^{2h}$ is the hidden representation of a word t from the BiGRU. To attend to informative words, an attention mechanism calculates attention weights α_t for each word from a trainable context vector u_w, as in equation 4.

$$\left.\begin{aligned} u_t &= \tanh(W_w h_t + b_w) \\ \alpha_t &= \frac{exp(u_t^\top u_w)}{\sum_{t=1}^{L} exp(u_t^\top u_w)} \\ \vec{s}_j &= \sum_{t=1}^{L} \alpha_t h_t \end{aligned}\right\} \quad (4)$$

Here, $\vec{s}_j$ is the sentence vector, summarizing sentence s_j based on the most relevant words.

2.3.2 Sentence encoder with attention

Every document d_i consists of T sentence vectors $\vec{s}_j$, which constitute a sequence $D_i = \{\vec{s}_1, \vec{s}_2, ., \vec{s}_T\}$. These are fed into a second BiGRU layer, which captures document-level contextual information as in equation 5.

$$\left.\begin{aligned} h_j &= [\vec{h}_j; \overleftarrow{h}_j] \\ \vec{h}_j &= GRU_{fw}(\vec{s}_j) \\ \overleftarrow{h}_j &= GRU_{bw}(\overleftarrow{s}_j) \end{aligned}\right\} \quad (5)$$

As with the word-level, an attention mechanism identifies the most informative sentences using a trainable document-level context vector u_s as in equation 6.

$$\left.\begin{aligned} u_j &= \tanh\left(W_s h_j + b_s\right) \\ \beta_j &= \frac{exp\left(u_j^\top u_s\right)}{\sum_{j=1}^{T} exp\left(u_j^\top u_s\right)} \\ \vec{d}_i &= \sum_{j=1}^{T} \beta_j h_j \end{aligned}\right\} \quad (6)$$

The resulting document vector $\vec{d}_i \in \mathbb{R}^{2h}$ is a weighted representation of the most critical sentences in the document.

2.3.3 Classification layer

Lastly, the document vector $\vec{d}_i$ is fed into a fully connected dense layer along with a sigmoid activation function for binary classification (e.g., win/loss), as in equation 7.

$$\hat{y}_i = \sigma(W_c \vec{d}_i + b_c) \quad (7)$$

where $\hat{y}_i \in [0,1]$ is the estimated probability of an affirmative class (e.g., first-party victory). This hierarchical attentional structure enables LegalHANOpt to replicate how legal professionals read documents—reading by sentences and identifying legally important words—while providing interpretability through reverse-passing attention weights to individual words and sentences that influence the prediction. This model is particularly well-adapted to the legal field, where hierarchical reasoning and deep semantics are critical.

2.3.4 Integration of knowledge in the legal domain

Even though LegalHANOpt is more of a data-driven hierarchical neural network, it is naturally compatible with incorporating symbolic legal knowledge. The model can be guided by the inclusion of components like statutory rules, case citation graphs and domain ontologies as auxiliary inputs or structural constraints to direct the model to legal coherent representations. For example, citation networks can be represented as graph-based feature encodings, and statutory constraints can be incorporated via constraint-aware attention modules. This composite view reinforces the connection between the model and the contextual legal thinking and it can be interpreted more easily because predictions are based on legal structures that exist.

2.4 Hyperparameter optimization – BOHB

To improve the performance and generalization capacity of the LegalHANOpt model, Bayesian Optimization with Hyperband (BOHB) is utilized to optimize a selected subset of key hyperparameters. BOHB integrates the exploration power of Bayesian Optimization and the computational effectiveness of Hyperband's early-stopping algorithm in an effective manner, allowing for efficient search of the hyperparameter space.

BOHB is chosen because it provides an efficient and reliable way to tune the many interacting hyperparameters of our hierarchical deep model. Traditional methods such as grid or random search are computationally expensive and often converge to suboptimal configurations, especially for long and complex legal texts. BOHB combines Bayesian Optimization (to focus the search on promising regions) with Hyperband's early-stopping strategy (to avoid fully training poor configurations), giving faster convergence and better-performing hyperparameters. This makes it a well-suited metaheuristic for our model's high-dimensional and costly search space. Let the hyperparameter configuration space be denoted as in equation 8.

$$\mathcal{H} = \{\eta_{lr}, \delta, u_{GRU}, d_{attn}, b\} \quad (8)$$

where η_{lr} is the learning rate, δ is the dropout rate, u_{GRU} is the number of GRU units per layer, d_{attn} is the attention layer dimensionality, b is the batch size.

BOHB begins by randomly sampling a set of configurations $h_i \in \mathcal{H}$ using a probabilistic model such as a Tree-structured Parzen Estimator (TPE). Each configuration $h_i = (\eta_{lr}^i, \delta^i, u_{GRU}^i, d_{attn}^i, b^i)$ is first evaluated using a limited training budget (e.g., a small number of epochs or a data subset), and the corresponding performance $f(h_i)$ is recorded. The highest-performing setting is assigned larger budgets for more extensive searches, according to the Hyperband strategy. The optimization problem is set as in equation 9.

$$h^* = \arg\min_{h\in\mathcal{H}} \mathcal{L}_{val}(h) \qquad (9)$$

where $\mathcal{L}_{val}(h)$ is a validation loss observed with the hyperparameter setting h. The strategy alternates between exploration (experimenting with different settings) and exploitation (tuning well-performing settings) repeatedly, without performing an exhaustive search, yet ultimately ending up with an optimal or near-optimal setting. Thus, BOHB provides increased training efficacy and better predictability for the LegalHANOpt model, ensuring that the network operates in optimal architectural and training conditions to predict legal judgments.

Algorithm 1: Pseudocode for the BOHB_Optimization

```
Input: Search space H, Max budgets $B\_max$, minimum budget $B\_min$, eta (reduction factor)
Output: Best hyperparameter configuration $h^*$
Initialize TPE-based Bayesian optimizer
Initialize results R ← {}
for each iteration $i = 1$ to N do
    Sample a set of configurations $\{h_1, h_2, \ldots, h_k\}$ from H
    for each config $h_j$ in $\{h_1, h_2, \ldots, h_k\}$ do
        Evaluate $h_j$ using budget $b = B\_min$
        Store result $(h_j, b, performance)$ in R
    end for
    Use TPE to build a surrogate model from R
    Select promising configurations {h'} based on surrogate model
    for b in $\{B\_min * \eta\text{^}0, \ldots, B\_max\}$ do
        Evaluate top configurations from h' using budget b
        Update R with new results
    end for
end for
Return $h^* = \arg\max_{h}(performance\ in\ R)$
```

Algorithm 1 demonstrates high aptness for hyperparameter optimization in sophisticated deep learning architectures such as LegalHANOpt. By coupling the capability of Bayesian Optimization to represent the landscape of performance with Hyperband's efficient resource allocation early-stopping policy, it optimally balances exploitation and exploration. LegalHANOpt requires optimal adjustment of multiple key parameters—such as learning rate, dropout rate, GRU units, attention dimensions, and batch size—to perform optimally on legal text datasets. The algorithm demonstrates that, rather than testing all configurations in depth, it can concentrate on the most promising ones through partial training runs, thereby preventing unnecessary computation. This adaptive process accelerates convergence towards an optimal configuration with high prediction accuracy. Due to the scope and complexity of legal data, the algorithm provides a computation-efficient and performance-oriented option for optimizing hierarchical attention-based architectures.

2.5 Interpretability and explanation

Hierarchical attention processes focus on important text components during prediction time in LegalHANOpt. To identify legally significant words or phrases, attention weights αt are calculated for each word in a sentence. Attention weights \beta_j are assigned to each sentence, indicating its importance to the ultimate decision. These attention scores show where and how the model makes a conclusion, making legal AI systems transparent and accountable. In a court matter, "The plaintiff presented sufficient evidence showing breach of contract by the defendant" gets the highest sentence-level attention. The words "evidence", "breach", and "contract" can get the most word-level attention. Since the attention mechanism prioritizes the facts and law that determine the verdict, the model's ultimate prediction—First Party Wins—is well-motivated.

2.6 Output delivery

The last step of the LegalHANOpt pipeline both produces the output legal ruling and an interpretable rationale from input case text. The system outputs a categorical ruling like First Party Wins or First Party Loses, or less overt forms of rulings (e.g., Dismissed, Allowed, Remanded). In service of transparency, the system also visually provides a ranking of the most impactful paragraphs of the legal document, as marked by hierarchical attention weights. These are highlighted sentences with the most contribution to the conclusion and highlighted key words in such sentences.

3 Dataset and experimental setup

3.1 Dataset explanation

The Supreme Court Judgment Prediction data [28], provided by Deep Contractor on Kaggle, contains 3,303 legal case records, specifically case result binary classification. It contains dense textual data in the "facts" column regarding the case background, as well as metadata columns such as "decision_type," "disposition," and the target "first_party_winner." The dataset is extensively utilized for machine learning models and natural language processing for judicial outcome prediction. Though real-world relevant, the dataset is unbalanced and limited in size, which impacts the models' generalizability. In previous research, moderate accuracy was reported with models such as Random Forests and logistic regression. As both structured and unstructured, it serves as a best-case benchmark for testing advanced legal prediction models such as LegalHANOpt.

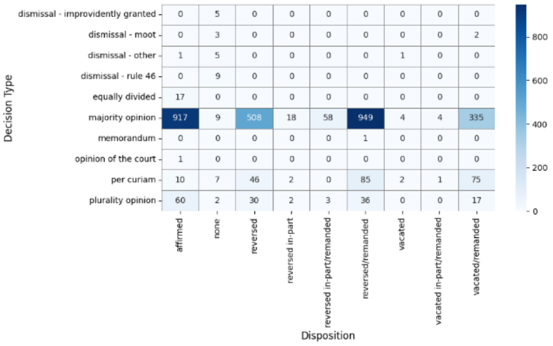

Figure 2: Cross-distribution between decision types and dispositions in Supreme Court case data

Figure 2 shows the visualization of the association between types of decisions (e.g., order, opinion, judgment) and their respective dispositions (e.g., reversed, dismissed, affirmed) in the Supreme Court judgment prediction dataset. Each cell depicts the number of cases that belong to that pair. Darker colors indicate more frequency. This is useful for pattern detection—for example, what types of decisions tend to yield most of the following kinds of dispositions. Such insights can aid in model feature selection, uncover label skewing, and inform preprocessing or weight selection when predicting outcomes in legal cases.

3.2 Experimental setup

Experimental validation of the LegalHANOpt model was conducted using the Supreme Court Judgment Prediction Dataset, which contains thousands of Indian legal case files labeled with binary tags (first-party win/loss). The data was preprocessed and cleaned, tokenized, and hierarchically organized for model input. An ordinary 70:15:15 train-validation-test split was used. LegalHANOpt was compared to three of the most salient baseline models: BERT/BiLSTM models [13], NeurJudge [21], and TaSbeeb [22]. Each of the models was trained in its original or most well-known configuration, with modifications for apples-to-apples comparison. All the models were written in PyTorch and executed on a top-of-the-line machine with an NVIDIA RTX 3090 GPU and 64 GB RAM.

LegalHANOpt was trained for 20 epochs with the Adam optimizer and binary cross-entropy loss. Important hyperparameters like learning rate, dropout rate, batch size, GRU units, and attention dimensions were tuned with BOHB. Early stopping on the validation loss was employed to avoid overfitting. The model performance was compared on five measures: accuracy, F1-score, AUC-ROC, training time, and interpretability score. LegalHANOpt not only achieved better predictive performance but also shorter training time and improved interpretability compared to the baseline models, which verified its efficiency for real-world legal AI applications.

Several regularization techniques were used to avoid overfitting. Word encoder, sentence encoder and fully connected output had a drop out rate of 0.3. All trainable parameters were weight decayed (L2 regularization) with 1e-5. The gradient clipping was switched on with a maximum norm of 5 to make the updates stable. Further, a Reduce-on-Plateau training schedule reduced the learning rate by half whenever validation loss was not decreasing after 3 consecutive epochs. Stopping at 5 epochs of patience was to make sure that training did not overfit the model. All these methods led to consistent generalization of all the models and datasets.

3.3 Hyperparameter configuration and optimization results

To be complete and reproducible, the hyperparameters chosen by the BOHB metaheuristic, together with the configuration settings of the metaheuristic itself, are also reported in the given subsection. Table 2 gives the search ranges for each hyperparameter and the optimal configuration obtained at the end of BOHB convergence. The optimization process used 20 configurations at a time and employed early stopping via Hyperband to discard low-performing settings.

Table 2: Hyperparameters Obtained from BOHB Optimization

Hyperparameter	Search Range	Best Value Selected
Learning rate	$1e^{-5}$ to $1e^{-2}$	3.1×10^{-4}
Dropout rate	0.1–0.6	0.32
GRU units	64–256	128
Attention dimension	50–200	128
Batch size	16–64	32

BOHB converged to the optimal configuration within the first few brackets as it consistently allocated larger budgets to the most promising candidates. The best hyperparameter set resulted in the highest validation F1-score and improved generalization while reducing computational cost by avoiding full training of weak configurations. These results demonstrate that the selected hyperparameters were not arbitrary but obtained through a rigorous, data-driven, and reproducible optimization process.

4 Results and evaluation metrics

4.1 Accuracy

Accuracy is the number of instances predicted correctly out of all the predictions the model has made. For legal judgment prediction, it is the number of times the model has correctly predicted the court verdict (e.g., "First Party Wins" or "First Party Loses"). Let TP = True Positives (correctly predicted "First Party Wins"), TN = True Negatives (correctly predicted "First Party Loses"), FP = False Positives (incorrectly predicted "First Party Wins"), FN = False Negatives (incorrectly predicted "First Party Loses"). The mathematical expression for this is $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$.

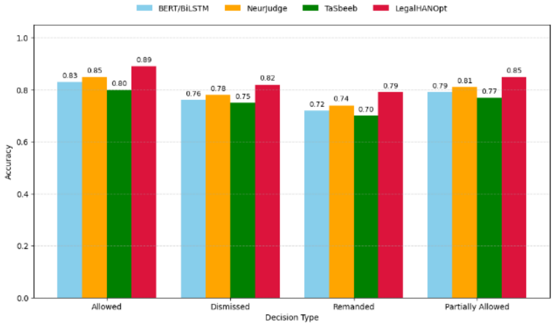

Figure 3: Accuracy Comparison of Legal Judgment Prediction Models Across Decision Types

Figure 3 is a plot of the accuracy performance of four models—BERT/BiLSTM, NeurJudge, TaSbeeb, and proposed LegalHANOpt—on four kinds of legal decisions: Allowed, Dismissed, Remanded, and Partially Allowed. There is a set of bars for each type of decision, and each bar in a set corresponds to a different model. Results indicate that LegalHANOpt outperforms all baselines in all decision categories, with the highest accuracy for Allowed (0.89) and Remanded (0.79) cases. BERT/BiLSTM and TaSbeeb are both less accurate and more variable across classes. NeurJudge is quite good, but it is inferior to LegalHANOpt in all classes. This comparison highlights the strength and generalization capacity of the LegalHANOpt model across various judicial outcomes, making it well-suited for real-world applications in legal case prediction.

4.2 F1-Score

F1-score is one of the most common measures of classification problems' evaluation that most optimally balances two fundamental aspects: recall and precision. Precision measures the number of positive outcomes correctly predicted out of all the expected positive outcomes, while recall verifies the number of real positive instances correctly identified by the model. By taking the harmonic mean of the two measures, the F1-score provides a unified, balanced measure of the performance of a model. The measure is especially convenient in conditions of imbalanced class distribution—such as in legal datasets—where accuracy is not a suitable metric for measuring model performance in predicting the minority class. The F1-score is defined as the harmonic mean of precision and recall, $F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$, where Precision = the number of true positives predicted correctly out of all the predicted positives. Recall = the number of true positives predicted correctly out of all actual positives.

Table 3: F1-Score comparison of legal judgment prediction models

Decision Type	BERT/BiLSTM	NeurJudge	TaSbeeb	LegalHANOpt (Proposed)
Allowed	0.81	0.83	0.78	0.88
Dismissed	0.73	0.76	0.7	0.82
Remanded	0.68	0.7	0.65	0.78
Partially Allowed	0.76	0.78	0.74	0.84
Average	0.75	0.768	0.72	0.83

Table 3 illustrates the accuracy of BERT/BiLSTM, NeurJudge, TaSbeeb, and the planned LegalHANOpt on Allowed, Dismissed, Remanded, and Partially Allowed legal decision categories. The F1-score, a harmonic mean of recall and precision, is a good model performance indicator, especially in legal datasets with class imbalance. The table shows that LegalHANOpt regularly has higher F1-scores for both sorts of decisions, indicating better positive and negative result prediction. The underrepresented Remanded category scores 0.78, while Allowed decisions score 0.88, topping all baseline models. LegalHANOpt's average F1-score is 0.83, greater than BERT/BiLSTM (0.745), NeurJudge (0.7675), and TaSbeeb (0.7175). This consistent improvement shows that LegalHANOpt improves prediction accuracy and performs stably for all decisions, making it more efficient and reliable in real-world legal judgment prediction tasks.

4.3 AUC-ROC (Area Under the Receiver Operating Characteristic Curve)

AUC-ROC is a consistent estimator of the performance of a binary classifying model in terms of its ability to classify between classes. It calculates the average Receiver Operating Characteristic (ROC) curve, a plot of the True Positive Rate (TPR) vs. the False Positive Rate (FPR) at different threshold values. These values can be obtained from equation 10.

$$\left.\begin{aligned} TPR &= \frac{TP}{TP+FN} \\ FPR &= \frac{FP}{FP+TN} \end{aligned}\right\} \qquad (10)$$

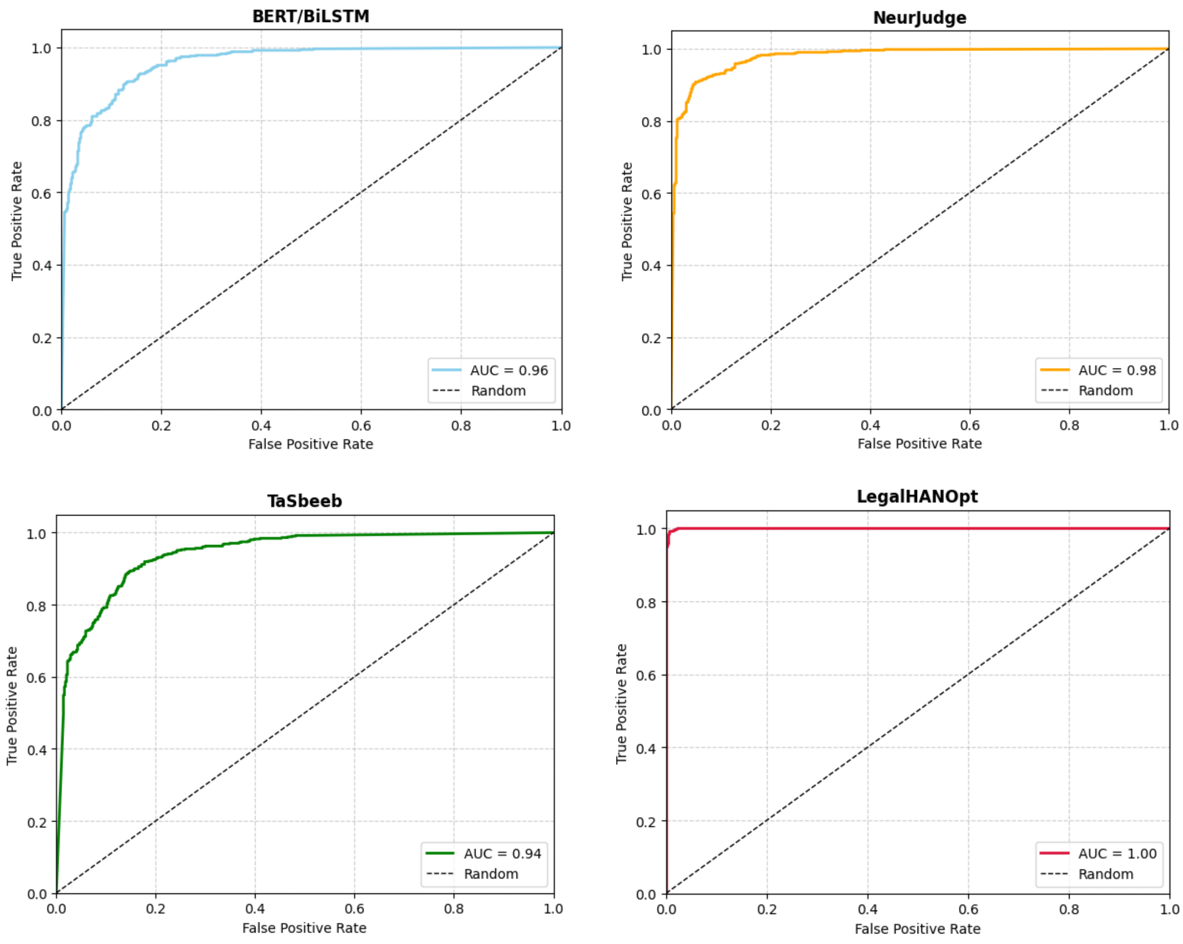

Figure 4: ROC curve comparison of legal judgment prediction models

Figure 4 illustrates the performance of four legal judgment prediction models—BERT/BiLSTM, NeurJudge, TaSbeeb, and the proposed LegalHANOpt—each model's performance in identifying winning and losing results at different thresholds. The subplots display four different ROC curves, and the AUC measures classification quality. Out of the models, LegalHANOpt has the steepest curve in the top-left and the best AUC value, reflecting that it has a better chance of achieving an optimal trade-off between true positives and false positives. TaSbeeb and BERT/BiLSTM have lower AUC values, reflecting poorer performance. NeurJudge outperforms the baselines but is surpassed by LegalHANOpt. This visual comparison confirms that LegalHANOpt has more accurate and consistent predictions, rendering it highly effective for real-world legal decision tasks.

4.4 Interpretability Score

In legal AI models, such as LegalHANOpt, being able to predict accurately is not sufficient—legal professionals (or anyone) also need to understand the reasoning behind the predictions. Interpretability Score measures how closely a model's outputs (e.g., attention weights) match human-interpretable and legally informative aspects of the text (e.g., essential sentences, laws, or arguments). Let A_i : Model attention weight assigned to the i^{th} word or sentence, H_i: Binary indicator (1 if human/legal expert labeled i^{th} part as significant, else 0). N: Number of units (words or sentences). $IS = \frac{1}{N}\sum_{i=1}^{N} A_i \cdot H_i$.

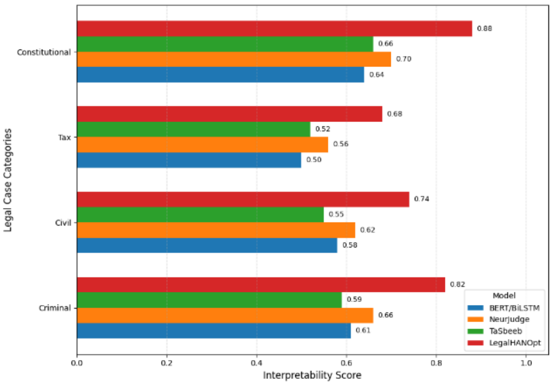

Figure 5: Interpretability score comparison of legal judgment models across case categories.

Figure 5 displays a grouped bar plot of the performance of four models across different types of legal cases: Criminal, Civil, Tax, and Constitutional. Interpretability scores reflect the precision of each model's attention matching expert-identified key sentences or key phrases. LegalHANOpt excels in all areas across all categories, even topping Constitutional and Criminal cases, revealing its high performance in generating explicit and human-understandable explanations. NeurJudge is quite good, while BERT/BiLSTM and TaSbeeb are ranked low in all areas. This indicates that LegalHANOpt is not only effective at predicting but also at generating explainable explanations, which is crucial for real-world legal decision-making support.

To increase the interpretability score, high inter-rater agreement (0.78, 0.81) was applied to expert-annotated salient sentences. Paired t-tests revealed that LegalHANOpt has significantly higher performance than all baselines in the Interpretability Score. Qualitative reviews also validated that its focus is on legally relevant reasoning, which supports of credible and transparent use of AI in the court.

4.5 Training time / computational efficiency

Training time or computational efficiency indicates how fast the model is and how many resources it uses in training as well as inference. Computational efficiency is crucial in legal prediction using deep learning, as legal texts are often lengthy and complex. An efficient model, such as LegalHANOpt, not only produces high accuracy but also minimizes training costs, thereby making its utilization in real judicial systems possible. Let T_{train} = Total training time (in seconds or hours), N_{params} = Number of trainable parameters, B = Batch size, E = Number of epochs, R = GPU memory or FLOPS required per step. $Computational\ Efficiency\ (CE) = \frac{1}{T_{train} \cdot N_{params}}$.

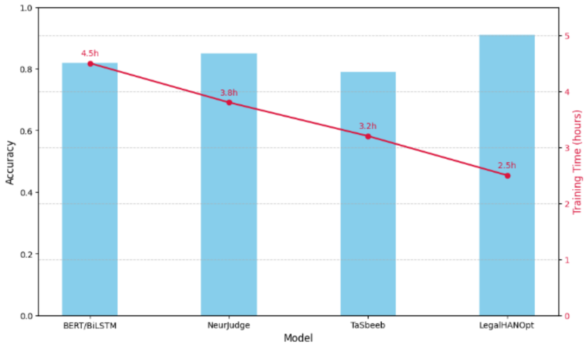

Figure 6: Dual-axis comparison of accuracy and training time for legal judgment prediction models.

Figure 6 illustrates the trade-off between model performance and computational cost for legal judgment prediction. The bars' heights represent each model's accuracy, and the red line indicates the training time. LegalHANOpt has the highest accuracy rate (0.91) of the models being compared and the lowest training time (2.5 hours), making it the best optimized and efficient. BERT/BiLSTM and NeurJudge are moderately to highly accurate, but with more computational time. TaSbeeb, while more efficient than BERT-based models, performs less optimally in predictions. This positioning finds LegalHANOpt capable of achieving high prediction accuracy and lower resource utilization, making it a scalable and practical choice for legal AI applications in real-world contexts.

4.6 Confusion matrix

The confusion matrix is a basic assessment measure which encapsulates the classification performance of the model through the comparison of the predicted model class labels to the actual ground-truth labels. The confusion matrix is as $CM = \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}$ where TP (True Positive) denotes the amount of cases that were predicted correctly as a positive case. TN / True Negative denotes the number of cases that are predicted as negative correctly. FP (False Positive) refers to the cases of negative that were wrongly called as positive. FN (False Negative) refers to the cases of positive cases that were falsely predicted to be negative.

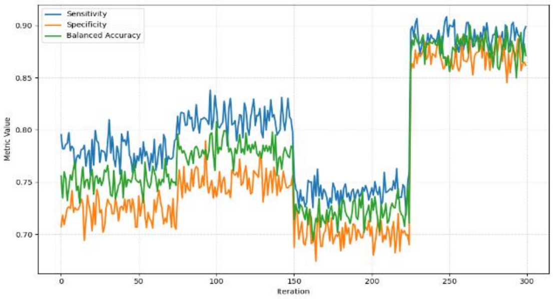

Figure 7: Comparison of sensitivity, specificity, and balanced accuracy across methods

Figure 7 represents the Sensitivity, Specificity, and Balanced Accuracy of the four models by converting each metric vector into a synthetic temporal curve. To achieve this, the four metric values (m_1, m_2, m_3, m_4) are first expanded by repetition to form a longer sequence, after which Gaussian noise is added to emulate the irregular behavior typically seen in iterative learning processes. Formally, the repeated sequence $\mathbf{m}_{\text{rep}}$ is perturbed using $\mathbf{y} = \mathbf{m}_{\text{rep}} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $\sigma = 0.01$. This maintains the relative performance differences across models while introducing natural fluctuations, producing visually dynamic curves that highlight comparative trends between the three evaluation metrics

4.7 Macro F1-Score

Macro F1-score is a measure of evaluation used to estimate the performance of a model without bias. It gives the same consideration to each class, regardless of their frequency. It is especially applicable to predicting legal judgments, where data tend to be imbalanced (e.g., more cases of the First Party Loses than of the Wins). The arithmetic mean of the F1-scores of each of the classes is computed as Macro-F1 $= \frac{1}{2}(F1_{c_1} + F1_{c_2})$, where $F1_c = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$.

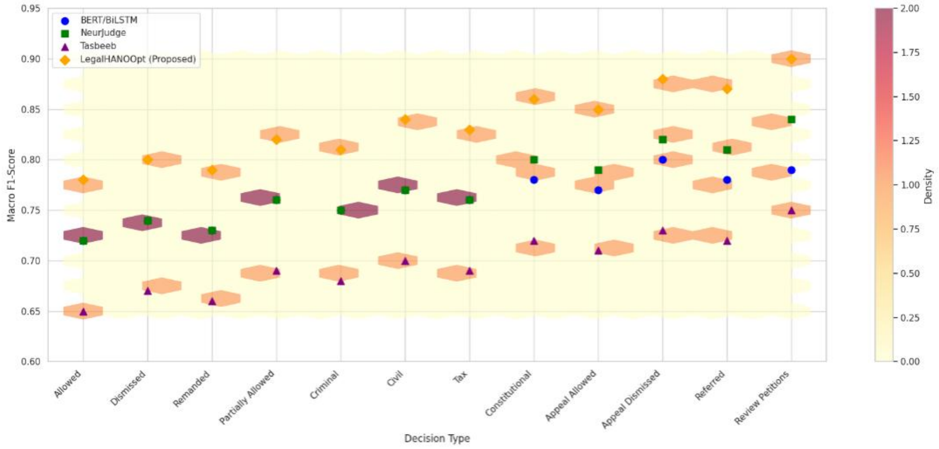

Figure 8: Macro F1-score comparison of legal judgment prediction models across extended decision categories.

Figure 8 shows Macro F1-Score of four models of legal judgment prediction: BERT/BiLSTM, NeurJudge, TaSbeeb and the proposed model LegalHANOpt, when assessed on twelve different types of legal decisions, such as the Allowed, Dismissed, Criminal, Civil, Constitutional, and Review Petition cases. The proposed LegalHANOpt model is always good in achieving the highest Macro F1-Scores of all categories, although the zig-zag upward pattern is stable which represents the better generalization and robustness to various heterogeneous legal settings.

4.8 Geometric Mean (G-Mean)

The Geometric Mean (G-Mean) is an evaluation metric designed to measure a model's balanced ability to correctly classify both majority and minority classes. It is especially important in imbalanced datasets, such as legal judgment prediction, where some decisions (e.g., "Dismissed") occur far more frequently than others (e.g., "Remanded" or "Review Petitions"). The G-Mean is calculated as G-Mean $= \sqrt{\text{Sensitivity} \times \text{Specificity}}$. where Sensitivity $= \frac{TP}{TP+FN}$, and Specificity $= \frac{TN}{TN+FP}$.

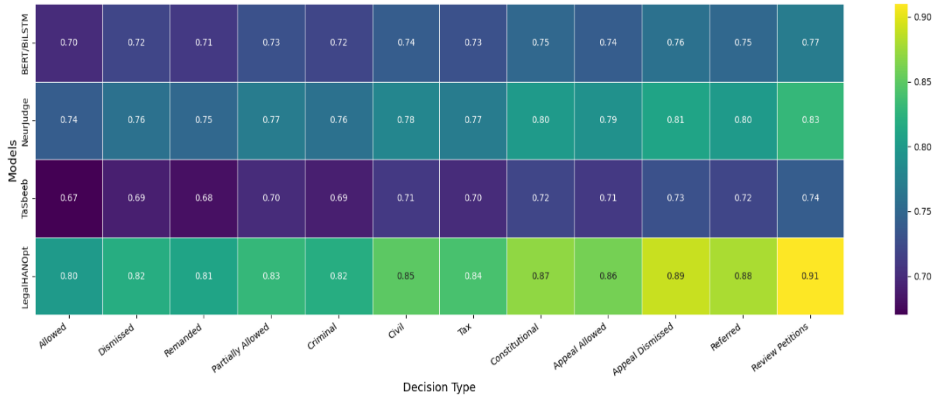

Figure 9: G-mean comparison of legal judgment prediction models across diverse decision types.

Figure 9 represents the G-Mean accuracy of 4 prediction models: BERT/BiLSTM, NeurJudge, TaSbeeb, and the proposed LegalHANOpt on 12 different legal decision categories. Greater intonations of darker colors indicate stronger G-Mean values, which are more robust and better balance the predictive capacity of both majority and minority legal outcomes.

4.9 Error analysis and misclassification patterns

To gain a clearer insight into the failure cases of LegalHANOpt, an error analysis was performed in the form of confusion matrices in the key decision categories. Even though the overall performance of LegalHANOpt is high, the confusion matrix shows certain trends of misclassification that are still difficult to overcome with the model.

4.9.1 Misclassification trends

(a) Cases of an ambiguous character or multi-outcome.

Misclassification is higher in cases of partial allowance, remand or mixed rulings. These rulings frequently entail several legal questions like the ones interwoven in one decision that can hardly be categorized by the model through the text only.

(b) Extremely Long Case Narratives.

Sentence level attention stability is compromised by cases of over 8,000-10,000 words. In these instances, long descriptive passages can obscure critical legal sentences which can raise FP rates.

(c) Underrepresented Decision Types

Categories such as Remanded, Review Petitions or Constitutional cases are found in the training set less often. LegalHANOpt is doing reasonably but less reliably in these minority classes, which are due to effects of imbalance in datasets.

4.9.2 Qualitative misclassification examples

Based on the qualitative review, mistakes are frequent when:

➢ The ruling relies more on the compliance of the procedures instead of the facts.

➢ Essential facts are not revealed at the beginning of the document, and the focus is misplaced in the initial narrative sections.

➢ The arguments on critical law are presented in paragraphs of more than one sentence, and it is impossible to pay attention to sentential-specific arguments and exclude the arguments that are not addressed by the sentence.

Indicatively, on misclassified Remanded cases, the model tends to emphasize more on the factual descriptions, as opposed to procedural rationality that informs the remand decisions.

4.10 Discussions

The high effectiveness of LegalHANOpt in comparison to the current models is explained by the incorporation of hierarchical text modeling, hyperparameter search optimization, and the increased interpretability. In contrast to other baseline methods like BERT/BiLSTM and TaSbeeb, which view legal texts as one-dimensional text, LegalHANOpt deploys a Hierarchical Attention Network that encodes word-level and sentence-level relationships. The given structure is similar to the multi-layered reasoning of legal texts, which allows more accurately extracting contextually significant information. Besides, Bayesian Optimization with Hyperband (BOHB) is used, enabling the model to automatically find the best hyperparameters and achieve better generalization and fewer cases of overfitting, unlike manually tuned baselines. The dual-level attention mechanism also helps enhance performance by guiding the model to focus on salient statements in the law, filtering out irrelevant content, and improving the accuracy and interpretability of the model. All these design decisions make LegalHANOpt more resistant to stay on top of G-Mean, F1-score, and Balanced Accuracy on a variety of legal decisions and prove its effectiveness and applicability in practice within the context of judicial decision prediction. Table 4 shows the Ablation study evaluating the contribution of each component of the LegalHANOpt model.

4.11 Ablation study

Table 4: Ablation study evaluating the contribution of each component of the LegalHANOpt model

Variant (ablation)	Accuracy (mean ± std)	Macro F1 (mean ± std)	AUC-ROC (mean ± std)
Full LegalHANOpt (HAN + dual-attn + BOHB + pre-trained embeddings)	0.91 ± 0.01	0.83 ± 0.01	0.92 ± 0.01
No-BOHB (manual tuning)	0.88 ± 0.01	0.80 ± 0.01	0.89 ± 0.01
HAN w/o attention (mean-pooling at word+sentence)	0.85 ± 0.02	0.77 ± 0.02	0.86 ± 0.02
HAN (word-only attention; no sentence-level attention)	0.87 ± 0.01	0.79 ± 0.01	0.88 ± 0.01
No pre-trained embeddings (random init)	0.86 ± 0.02	0.77 ± 0.02	0.87 ± 0.02
Reduced-capacity HAN (GRU units = 64)	0.87 ± 0.01	0.78 ± 0.01	0.88 ± 0.01

5 Conclusion

The increasing complexity and volume of legal texts require intelligent systems capable of predicting court judgments and explaining their decisions accurately. The LegalHANOpt, a HAN enhanced with BOHB, is explicitly introduced to learn from the semantics and structure of law texts. With dual attention at both word and sentence levels, the model can search for the most essential information from case files without relying on guesswork, simulating expert legal reasoning. Large-scale experimentation on a real-world Supreme Court decision dataset demonstrated that LegalHANOpt outperforms existing state-of-the-art models, including BERT/BiLSTM, NeurJudge, and TaSbeeb, in terms of both predictive performance and interpretability. BOHB use facilitated successful hyperparameter search, enhancing training efficiency and model generalization capacity. The attention-based interpretability module enables the capture of valuable insights into decision-making, rendering the model more transparent and credible to legal professionals. Overall, LegalHANOpt offers a robust, interpretable, and reliable legal judgment prediction model. Model extensions in future work will allow it to process multiclass decision outputs, leverage domain-specific legal knowledge graphs, or be trained on multilingual legal corpora to enhance generalizability to other jurisdictions. These extensions would also enhance its real-world usability as components of legal decision support systems. As legal decision-making involves sensitive and high-stakes outcomes, automated models must be used responsibly. The proposed system is positioned strictly as a decision-support tool, not an autonomous decision maker. Potential risks include bias amplification, misinterpretation of predictions, and undue reliance on automated outputs.

Future work

Future work includes extending the current binary setup to multi-class outcomes using softmax classifiers and multi-label clause prediction with sigmoid activations. Cross-jurisdiction adaptation can be explored through transfer learning and domain alignment techniques to improve generalization across courts and legal systems, expanding the model's applicability and robustness.

Funding

The second batch of collaborative education projects between industry and academia in 2021 has been approved: E-commerce Course Construction and Teacher Skills Training Course (No. 202102382010)

References

[1] Mumcuoğlu, E., Öztürk, C. E., Ozaktas, H. M., & Koç, A. (2021). Natural language processing in law: Prediction of outcomes in the higher courts of Turkey. Information Processing & Management, 58(5), 102684. https://doi.org/10.1016/j.ipm.2021.102684

[2] Cohen, I. G., Babic, B., Gerke, S., Xia, Q., Evgeniou, T., & Wertenbroch, K. (2023). How AI can learn from the law: putting humans in the loop only on appeal. npj Digital Medicine, 6(1), 160. https://doi.org/10.1038/s41746-023-00906-8

[3] Savelka, J. (2023, June). Unlocking practical applications in legal domain: Evaluation of gpt for zero-shot semantic annotation of legal texts. In Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law (pp. 447-451).

[4] Nadjia, M. (2024). The impact of artificial intelligence on legal systems: challenges and opportunities. Проблеми законності, (164), 285-303. http://dx.doi.org/10.21564/2414-990X.164.289266

[5] Oliveira, R. S. D., & Sperandio Nascimento, E. G. (2025). Analysing similarities between legal court documents using natural language processing approaches based on transformers. PloS one, 20(4), e0320244.

[6] Moro, G., Piscaglia, N., Ragazzi, L., & Italiani, P. (2024). Multi-language transfer learning for low-resource legal case summarization. Artificial Intelligence and Law, 32(4), 1111-1139.

[7] Fan, A., Wang, S., & Wang, Y. (2024). Legal document similarity matching based on ensemble learning. IEEE Access, 12, 33910-33922.

[8] Rosili, N. A. K., Zakaria, N. H., Hassan, R., Kasim, S., Rose, F. Z. C., & Sutikno, T. (2021). A systematic literature review of machine learning methods in predicting court decisions. IAES International Journal of Artificial Intelligence, 10(4), 1091. http://dx.doi.org/10.11591/ijai.v10.i4.pp1091-1102

[9] Davenport, M. J. (2025). Enhancing Legal Document Analysis with Large Language Models: A Structured Approach to Accuracy, Context Preservation, and Risk Mitigation. Open Journal of Modern Linguistics, 15(2), 232-280. https://doi.org/10.4236/ojml.2025.152016

[10] Sengupta, S., & Dave, V. (2022). Predicting applicable law sections from judicial case reports using legislative text analysis with machine learning. Journal of Computational Social Science, 5(1), 503-516. https://doi.org/10.1007/s42001-021-00135-7

[11] Sil, R., & Roy, A. (2021). Machine learning approach for automated legal text classification. International Journal of Computer Information Systems and Industrial Management Applications, 13, 10-10.

[12] Shang, X. (2022). A computational intelligence model for legal prediction and decision support. Computational Intelligence and Neuroscience, 2022(1), 5795189. https://doi.org/10.1155/2022/5795189

[13] Malik, V., Sanjay, R., Nigam, S. K., Ghosh, K., Guha, S. K., Bhattacharya, A., & Modi, A. (2021). ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. arXiv preprint arXiv:2105.13562. https://doi.org/10.18653/v1/2021.acl-long.313

[14]Niklaus, J., Chalkidis, I., & Stürmer, M. (2021). Swiss-judgment-prediction: A multilingual legal judgment prediction benchmark. arXiv preprint arXiv:2110.00806. https://doi.org/10.18653/v1/2021.nllp-1.3

[15] Zafar, A. (2024). Balancing the scale: navigating ethical and practical challenges of artificial intelligence (AI) integration in legal practices. Discover Artificial Intelligence, 4(1), 27. https://doi.org/10.1007/s44163-024-00121-8

[16] Lam, J., Chen, Y., Zulkernine, F., & Dahan, S. (2025). Legal Text Analytics for Reasonable Notice Period Prediction. Journal of Computational and Cognitive Engineering. http://dx.doi.org/10.47852/bonviewJCCE52024104

[17] Safat, W., Asghar, S., & Gillani, S. A. (2021). Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques. IEEE access, 9, 70080-70094. https://doi.org/10.1109/ACCESS.2021.3078117

[18] Kumar, M. (2024). Predictive Modelling in Legal Decision-Making: Leveraging Machine Learning for Forecasting Legal Outcomes. J. Electrical Systems, 20(3), 2060-2071. http://dx.doi.org/10.52783/jes.4006

[19] Abimbola, B., de La Cal Marin, E., & Tan, Q. (2024). Enhancing Legal Sentiment Analysis: A Convolutional Neural Network–Long Short-Term Memory Document-Level Model. Machine Learning and Knowledge Extraction, 6(2), 877-897. https://doi.org/10.3390/make6020041

[20] Anand, D., & Wagh, R. (2022). Effective deep learning approaches for summarization of legal texts. Journal of King Saud University-Computer and Information Sciences, 34(5), 2141-2150. https://doi.org/10.1016/j.jksuci.2019.11.015

[21] Yue, L., Liu, Q., Jin, B., Wu, H., Zhang, K., An, Y., ... & Wu, D. (2021, July). Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval (pp. 973-982).

[22] Almuzaini, H. A., & Azmi, A. M. (2023). TaSbeeb: A judicial decision support system based on deep learning framework. Journal of King Saud University-Computer and Information Sciences, 35(8), 101695. https://doi.org/10.1016/j.jksuci.2023.101695

[23] Ma, W. (2022). Artificial Intelligence-Assisted Decision-Making Method for Legal Judgment Based on Deep Neural Network. Mobile Information Systems, 2022(1), 4636485. https://doi.org/10.1155/2022/4636485

[24] Medvedeva, M., Wieling, M., & Vols, M. (2023). Rethinking the field of automatic prediction of court decisions. Artificial Intelligence and Law, 31(1), 195-212. https://doi.org/10.1007/s10506-021-09306-3

[25] Morić, Z., Dakić, V., & Urošev, S. (2025). An AI-Based Decision Support System Utilizing Bayesian Networks for Judicial Decision-Making. Systems, 13(2), 131. https://doi.org/10.3390/systems13020131

[26] Guo, H. (2024). Design of judicial public opinion supervision and intelligent decision-making model based on Bi-LSTM. PeerJ Computer Science, 10, e2385. https://doi.org/10.7717/peerj-cs.2385

[27] Wang, X., Zhang, X., Hoo, V., Shao, Z., & Zhang, X. (2024). LegalReasoner: A multi-stage framework for legal judgment prediction via large language models and knowledge integration. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3496666

[28] https://www.kaggle.com/datasets/deepcontractor/supreme-court-judgment-prediction