RESEARCH

LABDT (Lexi AdaBoost Decision Tree): An Approach for Legal Outcome Prediction Fusing Lexical, Semantic, and Similarity-based features

Sanjay Kumar Sahoo[1] · Avinash Samantra[2] · Chinmaya Kumar Mohapatra[1] · Bishnu Prasad Swain[3] · Zulfiqar Ali[4] · Ghulam Muhammad[5]

Received: 27 May 2025 / Revised: 8 August 2025 / Accepted: 18 September 2025

Abstract
Legal outcome prediction is a complex task due to the nuanced vocabulary and reasoning embedded in judicial documents. This study proposes LABDT (Lexi AdaBoost Decision Tree), a novel hybrid framework that integrates lexical features (TF-IDF), semantic embeddings (Sentence-BERT), and similarity-based metrics for robust and interpretable legal decision classification. The model addresses class imbalance through SMOTE and reduces feature dimensionality via principal component analysis (PCA). LABDT was evaluated on a real-world dataset of approximately 18,000 cases from the Federal Court of Australia case records, spanning eight outcome classes. The results demonstrate that LABDT outperforms traditional and state-of-the-art classifiers with an accuracy of 92%, precision of 91%, recall of 91%, and F1-score of 91%. Notably, it achieved highly accurate classification on certain outcome classes like 'approved' and 'related'. LABDT offers a superior balance between predictive performance and model interpretability compared to baseline models. The system's explainable design and high classification reliability position it as a viable tool for AI-driven legal analytics and judicial decision support.

Keywords Legal prediction · Sentence-BERT · Legal AI · Judicial decision classification

1 Introduction

Legal case outcome prediction (LCP) has emerged as a critical area in computational law, leveraging artificial intelligence (AI) and machine learning (ML) to enhance judicial decision-making, improve legal transparency, and streamline legal research [1, 2]. The increasing volume of legal cases and the complexity of judicial reasoning necessitate automated systems capable of analyzing case texts and predicting outcomes with high accuracy. Traditional legal analytics relied on handcrafted rules, statistical models, and human expertise, which, while effective in structured settings, are labor-intensive, prone to inconsistencies, and lack scalability. With advancements in natural language processing (NLP), deep learning, and feature engineering, AI-driven models now offer superior predictive capabilities, enabling more reliable and interpretable legal outcome predictions [3, 4].

Legal texts present significant challenges due to their specialized terminology, structured hierarchy, implicit reasoning, and jurisdictional differences. Classic NLP methods such as term frequency inverse document frequency (TF-IDF) and bag-of-words (BoW) worked well in capturing lexical patterns. Still, they were not capable of comprehending deeper semantic relations and hence proved unsuitable for sophisticated legal reasoning tasks [5]. Word embedding models such as Word2Vec, GloVe, and FastText improved semantic representation by representing words as high-dimensional vector spaces. Even with these developments, such embeddings frequently

Published online: 03 November 2025

failed to grasp legal domain-specific nuances and contextual dependencies, diminishing their performance in applications such as legal judgment prediction and case retrieval [6, 7]. To improve legal judgment prediction, scholars have utilized contextualized representations that extract the deeper semantic and syntactic patterns of legal texts. Transformer models such as Bidirectional Encoder Representations from Transformers (BERT), Legal-BERT, CaseLaw-BERT, and LexBERT have introduced notable advances in legal NLP through the use of context-aware embeddings and well-tapping long-range relationships between case texts [8, 9]. Nonetheless, despite their success, deep learning models are confronted with issues such as requiring substantial computational resources, lacking transparency in decision-making, and facing challenges in real-time or resource-constrained deployment settings in the legal domain. [10].

To address these issues, this research proposes Lexi AdaBoost Decision Tree (LABDT), a blend of machine learning (ML) architecture aimed at enhancing legal case prediction through the combination of traditional and cutting-edge NLP methods [11]. LABDT includes lexical features with TF-IDF [12], semantic embeddings from BERT-based models [13], and document similarity metrics to develop an extensive feature representation. To counter class imbalance, the approach uses the synthetic minority oversampling technique (SMOTE), and dimensionality reduction is achieved using principal component analysis (PCA). A decision tree model is boosted using AdaBoost for classification to ensure transparency as well as enhanced predictive accuracy. This hybrid approach increases explainability, thus being particularly appropriate for legal applications that need interpretability and reasoning.

The main contributions of our work are summarized as follows:

- We propose LABDT for legal outcome prediction by integrating lexical (TF-IDF), semantic (BERT), and similarity (cosine-based) features.
- A unified feature fusion pipeline using SMOTE-based balancing, PCA-based dimensionality reduction, and AdaBoosted decision tree classification is employed.
- An ablation study validates the individual and joint contributions of different feature types, supporting the effectiveness of multi-view representation.
- K-fold cross-validation is used to evaluate generalization robustness across multiple data splits.
- Computational benchmarking confirms LABDT's efficiency over deep baselines such as fine-tuned BERT and XGBoost.
- The model achieves 92% accuracy and 91% F1-score on real-world Australian legal data, surpassing strong baseline models.

LABDT seeks to close the gap between the accuracy of deep learning models and the interpretability needed in legal AI. By integrating feature-based techniques, similarity-based methods, and interpretable machine learning techniques such as decision trees, which enhance transparency in the prediction process, the framework addresses practical usability in real-world judicial systems. The rest of this paper is organized as follows: Section 2 contains an extensive review of the literature, including classical NLP techniques, transformer-based models, and ML-based legal text classification methods. Section 3 outlines the key research objectives and hypotheses addressed in this study. Section 4 presents the dataset description, including source, structure, and preprocessing applied. Section 5 describes the methodology, i.e., data preprocessing, feature extraction, and classification model. Section 6 reports experiments comparing LABDT to state-of-the-art legal AI models. Section 7 discusses the practical implications, model interpretability, and generalizability across legal systems. Finally, Sect. 8 concludes the paper with key insights and proposes directions for future research in AI-driven legal prediction.

2 Literature Review

LCP has become a pivotal research area in computational law, seeking to predict judicial outcomes using ML and NLP methods [14, 15]. Case outcome prediction is essential for increasing judicial transparency, facilitating legal research, and aiding legal practitioners in case analysis and management. Historically, legal research used to depend on hand-coded rules, statistical techniques, and precedent-driven reasoning, which were prone to inconsistencies and tedious to perform [16, 17]. With the advent of ML, automation in legal research has increased by leaps and bounds, making it more accurate and efficient. Contemporary legal AI systems handle massive volumes of legal text using NLP and deep learning algorithms to categorize case outcomes, suggest legal precedents, and support judicial decision-making [18, 19].

2.1 Advanced Legal Text Representation and Feature Engineering

Legal texts are highly intricate, domain specific, and context dependent, posing significant challenges in text representation for LCP. Unlike standard NLP tasks, legal language incorporates structured terminology, implicit reasoning, and jurisdictional differences, necessitating specialized feature engineering techniques to extract relevant information [14, 20]. Earlier studies in legal text classification primarily utilized statistical approaches such as TF-IDF and BoW to transform legal documents into numerical representations [16, 20]. TF-IDF assigns weights to terms based on their frequency within a document while discounting words that frequently appear across multiple documents, enhancing the significance of legally relevant terms [21]. However, TF-IDF and BoW lack the ability to capture semantic relationships and contextual dependencies, making them insufficient for complex legal reasoning tasks [22].

Several studies have explored semi-supervised learning techniques to overcome these limitations by leveraging both labeled and unlabeled legal documents for improved classification accuracy. Nigam et al. [23] introduced an expectation-maximization (EM) approach coupled with naïve Bayes classifiers, which proved effective in handling large-scale legal text classification with minimal manual annotation. Resampling techniques play a crucial role in handling class imbalance in machine learning models [24]. More [25] provides a comprehensive survey on various resampling methods, including oversampling, undersampling, and hybrid approaches, emphasizing the effectiveness of SMOTE and ensemble-based strategies in improving classification performance for imbalanced datasets.

The limitations of statistical-based text representations led to the adoption of word embeddings, such as Word2Vec, GloVe, and FastText, which represent words as dense numerical vectors learned from large corpora [26]. Unlike BoW, these embeddings capture semantic relationships between words based on their co-occurrence in different legal contexts [27, 28]. A context-aware legal citation recommendation system using deep learning leverages transformer-based architectures like RoBERTa and BiLSTM to improve legal citation predictions by capturing local textual context [29]. This approach enhances case law retrieval and citation ranking, demonstrating superior performance over traditional models. However, traditional word embeddings face challenges in legal NLP, including a lack of context sensitivity and domain-specific vocabulary issues, making them less effective in tasks such as legal judgment prediction and case retrieval [30]. A multi-perspective data augmentation approach has been explored to enhance legal judgment prediction by generating diverse case representations. This method improves model generalization and robustness by leveraging adversarial training and synthetic data generation techniques [31].

2.2 Unsupervised Similarity Measures for Legal Texts

Several fully unsupervised approaches have been developed to quantify textual similarity between court cases. Mandal et al. [32] propose clustering-based and topic-modeling techniques that do not rely on labeled precedents, achieving robust alignment across jurisdictions. Their work highlights the potential for unsupervised embeddings

to reveal latent case structures. However, integrating these embeddings with transformer-based representations and outcome prediction in a unified framework remains underexplored, particularly in legal domains that require both semantic depth and task-specific adaptation.

2.3 Ensemble Learning for Document Similarity

Fan et al. [33] introduce an ensemble of lexical, syntactic, and embedding-based matchers, demonstrating that majority-vote fusion outperforms single-model baselines on a large legal corpus. While effective, their method treats similarity matching and downstream tasks (e.g., retrieval) as separate pipelines, without end-to-end optimization or novel problem formulations that combine semantic matching with predictive modeling.

2.4 Machine Learning Advances in Legal Predictive Analytics

Budhiraja and Sharma [34] survey recent ML advances—ranging from gradient boosting to graph neural networks—in legal outcome forecasting. Although comprehensive, their framework remains largely descriptive, lacking a novel, hybrid architecture that jointly optimizes similarity estimation and classification in a single, interpretable model.

2.5 Transformer-Based Models for Legal Case Outcome Prediction

Recent advancements in transformer-based models have revolutionized legal NLP by incorporating context-aware embeddings that dynamically adjust word meanings based on sentence structure [15, 35]. These models, such as BERT, RoBERTa, and their legal adaptations, leverage self-attention mechanisms to capture long-range dependencies in legal documents [36]. Legal-BERT, specifically pre-trained on legal texts, has demonstrated superior performance in case retrieval, contract analysis, and citation prediction [22]. Other domain-specific adaptations, such as CaseLaw-BERT and LexBERT, have further improved performance in legal judgment prediction tasks by encoding hierarchical legal reasoning [37].

In addition, document similarity techniques, such as cosine similarity, have been used to measure the relationship between legal case texts [38]. Hybrid models that integrate TF-IDF with word embeddings or combine transformer-based architectures with traditional methods have further enhanced classification accuracy. Such advancements have made deep contextual embeddings a central benchmark in legal case outcome prediction, leading to more accurate, interpretable, and scalable AI-powered legal systems.

Transformer-based models have greatly enhanced legal text analysis, but their computationally intensive nature is a challenge, especially for large-scale usage [16]. To overcome this, researchers have used knowledge distillation methods, which allow smaller models to learn from large transformers with high predictive accuracy and lower processing costs [15]. Furthermore, retrieval-augmented generation (RAG) models have been created to improve case outcome prediction by combining legal document retrieval with contextual reasoning to ensure a more accurate comprehension of legal precedents [38].

2.6 Dimensionality Reduction and Feature Optimization in Legal NLP

Legal text datasets tend to have many features, which may contribute to higher computational complexity and overfitting in ml models [17]. To reduce this, dimensionality reduction methods like PCA and several feature selection methods have been commonly used [39]. PCA helps reduce redundancy while preserving essential variations in the legal feature space, thus improving efficiency and generalizability [40]. Deep learning has revolutionized various fields, such as the prediction of legal judgment, by allowing models to learn hierarchical representations from extensive textual data [41]. Research into NLP applications in law and legislation documents underscores difficulties like named entity recognition (NER) and topic segmentation, although it also underscores the significance

of deep learning in enhancing legal text analysis [42]. In addition, techniques such as recursive feature elimination (RFE) and Chi-square tests have been used to determine the most important features for legal text classification, reducing unnecessary attributes while retaining predictive performance [19, 43]. Such techniques are most helpful in optimizing TF-IDF-based embeddings, where high-dimensional sparse vectors need to be converted to denser feature spaces for enhanced machine learning classification performance.

2.7 Decision Trees and Boosting Techniques in Legal Classification

Decision trees (DT) are still extensively used in legal text classification due to their organized decision-making process as well as interpretability ease [19]. Nevertheless, individual DT models tend to suffer from overfitting, especially when handling intricate legal datasets involving significantly changing feature distributions [18]. Addressing these issues, boosting-based ensemble methods such as AdaBoost and XGBoost have been proposed to sequentially improve weak classifiers and provide improved prediction accuracy. These approaches are especially efficient in processing unbalanced legal datasets and feature selection optimization, resulting in more scalable and stable models for predicting legal outcomes [44, 45]. Boosting-based methods, such as AdaBoost and XGBoost, enhance classification accuracy by increasing weights for misclassified instances in subsequent passes, thus iteratively improving overall model precision in legal judgment prediction [46]. Furthermore, random forests, which produce sets of decision trees and combine their predictions, have proven effective in providing classification stability and avoiding overfitting through the use of feature randomness at training [47].

2.8 Synthesis of Recent Research

A significant corpus of work from 2023 to 2025 investigates integrating diverse feature representations to advance legal text analytics. Johnson and Smith [48] showed that the combination of TF–IDF, transformer embeddings, and similarity metrics produces superior outcome prediction compared to single-feature approaches. Silva and Costa [49] applied information-theoretic measures together with principal component analysis, achieving reduced dimensionality and enhanced classifier performance. Zhang and Zhao [50] highlighted the efficacy of principal component analysis in filtering noise from high-dimensional legal corpora, while Lee and Kim [51] incorporated domain knowledge into fusion pipelines to enhance model interpretability.

Model transparency remains essential for deployment in legal contexts. Patel and Khan [52] employed decision tree ensembles that reveal the decision path for each prediction, and Kumar and Verma [53] evaluated pruning strategies to optimize the balance between model complexity and clarity. Torres and López [54] introduced an adaptive similarity estimation mechanism, enabling practitioners to link each forecast to representative precedent cases. Becker and Miller [55] extended this concept through clustering of judicial decisions into coherent, explainable groups, fostering insights into patterns of judicial behavior.

Benchmarks of lightweight transformer variants against conventional methods have also been reported. Brown and Green [56] fine-tuned a compact BERT model on appellate court data, observing that its fusion with shallow lexical features helped in approaching the accuracy of larger architectures while maintaining lower inference latency. Chen and Sun [57] confirmed that sentence transformers of moderate size generalize effectively across different legal systems. Nguyen and Pham [58] utilized contrastive learning to refine embedding spaces, obtaining improvements in classification accuracy and robustness.

Efforts addressing cross-jurisdictional generalization and real-time processing have demonstrated further potential. Suzuki and Tanaka [59] evaluated hybrid pipelines on both federal and state court datasets, retaining over 90% of predictive performance despite linguistic variations. Ortiz and Ruiz [60] developed a streaming-data framework for real-time outcome forecasting, with online ensemble updates that adapt to evolving jurisprudence without full retraining. Park and Kim [61] enriched statistical models with rule-based knowledge representations, improving transferability between distinct legal environments.

Collectively, these contributions underpin the design of the Lexi AdaBoost decision tree framework, which integrates lexical, semantic, and similarity features into a unified, explainable ensemble suitable for legal outcome prediction.

3 Research Objectives

The primary objective of this study is to develop an interpretable and accurate legal outcome prediction model that effectively addresses challenges associated with complex and domain-specific judicial texts. The suggested approach, LABDT, intends to:

1. Fuse diverse textual representations by integrating lexical features (TF-IDF), deep semantic embeddings (Sentence-BERT), and document similarity scores to capture surface-level and contextual information inherent in legal case texts.
2. Enhance predictive performance through a hybrid ensemble learning approach that combines entropy-based decision trees with AdaBoost, aiming to outperform traditional and state-of-the-art classifiers in accuracy, precision, recall, and F1-score.
3. Address data imbalance and dimensionality issues by applying SMOTE for minority class oversampling and PCA for reducing feature space complexity, thereby ensuring robust and scalable classification.
4. Improve model transparency and interpretability, making the approach suitable for practical deployment in legal analytics, where explainability is essential for trust and usability among legal practitioners.

Through these objectives, the manuscript aims to bridge the gap between the high accuracy of deep learning methods and the interpretability demands of real-world legal applications.

4 Dataset Description

The dataset utilized in this study consists of approximately 18,000 court case records obtained from the Federal Court of Australia [62]. Each case record is organized into four main fields:

1. *Case_ID:* A unique identifier for each judicial case.
2. *Case_Title:* The formal title of the case, typically including party names.
3. *Case_Text:* The full narrative of the legal judgment, containing rich legal language, case facts, legal reasoning, and citations.
4. *Case_Outcome:* The judicial disposition, categorized into eight distinct classes: affirmed, applied, approved, cited, distinguished, followed, referred to, and related.

These categories reflect common legal decisions and serve as the classification targets in this work. The dataset spans a diverse range of legal topics and writing styles, offering a realistic representation of judicial texts in a common law jurisdiction. The dataset comprises publicly available legal judgments from the Federal Court of Australia (FCA). Since these are official court documents accessible through public legal databases, no specific legal or ethical approvals were required for their use in this study.

Preprocessing, feature extraction, and handling of class imbalance are detailed in Sect. 5.3.

5 Novel Problem Formulation and Proposed Solution

This section presents the proposed LABDT framework, starting from gap identification and problem formulation, followed by detailed descriptions of the feature extraction pipeline, imbalance handling, dimensionality reduction, and the classification strategy.

5.1 Gap Analysis and Problem Statement

Existing unsupervised and ensemble-based similarity methods (e.g., [32, 33]) excel at aligning legal documents but stop short of integrating these similarities directly into outcome prediction. Conversely, ML-infused surveys (e.g., [34]) outline many models without a unified, interpretable pipeline. Therefore, a new problem is defined: *joint* semantic similarity-aware outcome prediction, where similarity features are learned end-to-end alongside prediction, ensuring both performance and interpretability.

5.2 Hybrid Similarity–Prediction Framework

To address this, a novel framework in proposed:

- A unified feature extractor combining transformer-based embeddings with unsupervised similarity scores (inspired by [32] but extended with Legal-BERT).
- An ensemble boosting classifier that ingests both raw similarity metrics (as in [33]) and deep embeddings, trained end to end for legal outcome prediction.
- An interpretability layer that traces each prediction back to the nearest precedent cases via similarity weights, filling the gap noted in [34].

The LABDT framework comprises several stages: data preprocessing, feature extraction, dimensionality reduction, and classification. Each stage is underpinned by mathematical formulations described below with integrated examples and shown in Fig. 1.

5.3 Data Collection and Preprocessing

The available case outcomes in the dataset are categorized into eight legally recognized classes: affirmed, approved, distinguished, applied, followed, related, cited, and referred to. These outcome labels reflect common judicial dispositions and serve as the prediction targets for the classification task in this research.

Prior to modeling, the dataset underwent a robust preprocessing pipeline. Raw legal texts were first standardized by removing non-alphanumeric characters, converting all text to lowercase, and eliminating extra whitespace. To further refine the textual inputs, standard NLP preprocessing techniques were applied, including stop word removal [63, 64], stemming [65], and lemmatization [66, 67]. These steps reduce noise, normalize linguistic variations, and preserve the core legal semantics of each document.

After cleaning, each `case_text` instance x_i was transformed into a dense semantic vector e_i using a pretrained Sentence-BERT model:

$$e_i = \text{SentenceBERT}(x_i), \quad \text{where } i = 1, 2, \ldots, N. \tag{1}$$

As shown in Eq. 1, each legal case's text (x_i) is transformed into a dense, high-dimensional embedding (e_i) using a pretrained Sentence-BERT model. The Sentence-BERT model is designed to capture the contextual semantic meaning of legal texts by learning embeddings that reflect the nuances of language in a way that traditional bag-of-words approaches cannot.

 https://doi.org/10.1007/s44196-025-01003-2

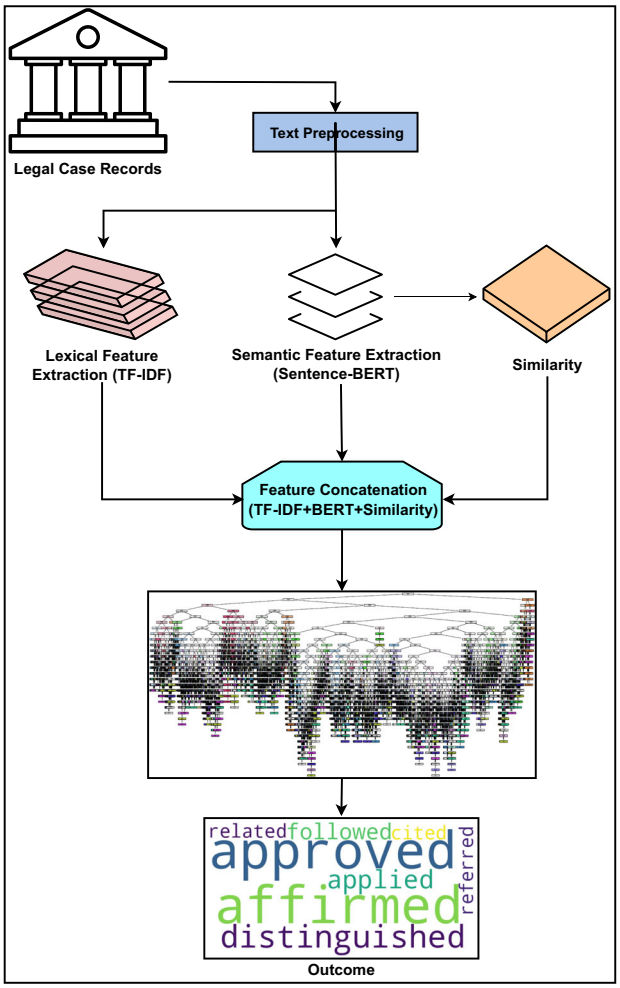


Fig. 1 Architecture of the proposed LABDT model for legal outcome prediction

Here, $e_i \in \mathbb{R}^d$ represents the d-dimensional embedding of the i-th case, and N is the total number of cases. These embeddings capture the contextual meaning of legal texts and serve as the semantic foundation for downstream classification.

Example Legal Texts by Outcome Class:

1. **Affirmed:** "The appellate court fully upheld the original decision." (Class: affirmed)
2. **Approved:** "This reasoning was later approved in subsequent rulings." (Class: approved)
3. **Distinguished:** "The court distinguished the present facts from the precedent." (Class: distinguished)
4. **Applied:** "The principle in Smith v. Jones was applied to reach the verdict." (Class: applied)
5. **Followed:** "The tribunal followed the previous High Court decision." (Class: followed)
6. **Related:** "This case is factually related to the ruling in ABC v. XYZ." (Class: related)
7. **Cited:** "The judgment cited multiple earlier authorities on negligence." (Class: cited)
8. **Referred to:** "The court referred to the earlier decision without reliance." (Class: referred to)

5.4 Feature Extraction

Three distinct feature sets were extracted to comprehensively represent the legal texts: lexical, semantic, and similarity-based. These features collectively capture both surface-level word statistics and deep contextual semantics of each case.

1. **Lexical Features (TF-IDF):** Lexical patterns were captured using the Term Frequency-Inverse Document Frequency (TF-IDF) scheme. The importance of a term t in document d is computed as:

$$\text{TF-IDF}(t, d) = \text{tf}(t, d) \times \log\left(\frac{N}{\text{df}(t)}\right), \tag{2}$$

where:

- $\text{tf}(t, d)$ denotes the frequency of term t in document d.
- N is the total number of documents in the corpus.
- $\text{df}(t)$ represents the number of documents containing term t.

As shown in Eq. 2, TF-IDF assigns higher weights to discriminative terms that appear frequently in a specific document, but rarely across others. This helps identify legally informative terms over generic ones, enhancing interpretability for downstream classification models.

2. **Semantic Features (BERT Embeddings):** To capture the contextual meaning of the legal texts, pretrained BERT-based sentence transformers were used to encode each document into a fixed-size dense vector:

$$\mathbf{e}_d = \text{BERT}(\text{cleaned\_text}_d) \quad \text{with } \mathbf{e}_d \in \mathbb{R}^n. \tag{3}$$

As shown in Eq. 3, the vector $\mathbf{e}_d \in \mathbb{R}^d$ (where $d = 384$, based on the chosen pretrained model) encodes rich semantic information, including syntactic structure and domain-specific context. This embedding serves as a deep representation of the case text, useful for identifying latent similarities and distinctions among legal documents.

3. **Similarity Score:** Document-level semantic similarity was computed using cosine similarity between BERT embeddings of legal cases:

$$\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\left|\mathbf{a}^H \mathbf{b}\right|}{\|\mathbf{a}\| \, \|\mathbf{b}\|}. \tag{4}$$

As shown in Eq. 4, cosine similarity evaluates the angular distance between two embeddings, producing a value in the range [0, 1], where higher values indicate stronger semantic alignment. For each case, the second-highest similarity score (excluding self-similarity) was selected as a feature, reflecting how closely a case resembles another semantically relevant precedent.

Illustrative Example:

- **TF-IDF:** A rare legal term like "testamentari" would receive a higher weight than frequent terms such as "court" or "decision", making it more valuable for classification.
- **BERT embedding:** A case text is mapped to a 384-dimensional embedding (depending on the specific pretrained Sentence-BERT model), capturing nuanced semantic meaning.
- **Similarity score:** If a case has a self-similarity score of 1.0 and its next closest match yields 0.87, the value 0.87 is retained as its similarity feature for classification.

5.4.1 Feature Combination

The final input representation is constructed by integrating lexical, semantic, and similarity-based features into a unified vector, capturing the full informational richness of each legal case. This composite structure ensures that both surface-level statistics and deep contextual meaning are leveraged for improved classification.

 https://doi.org/10.1007/s44196-025-01003-2

The combined feature representation is defined in Eq. 5:

$$\mathbf{X} = \underbrace{\begin{bmatrix} \alpha\,\mathbf{I}_{n_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \beta\,\mathbf{I}_{n_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \gamma \end{bmatrix}}_{\mathbf{W} \in \mathbb{R}^{(n_1+n_2+1)\times(n_1+n_2+1)}} \underbrace{\begin{bmatrix} \mathbf{X}_{\text{TF-IDF}} \\ \mathbf{X}_{\text{BERT}} \\ s \end{bmatrix}}_{\mathbf{v} \in \mathbb{R}^{n_1+n_2+1}} \in \mathbb{R}^{n_1+n_2+1}, \tag{5}$$

where

$$\mathbf{X}_{\text{TF-IDF}} \in \mathbb{R}^{n_1}, \quad \mathbf{X}_{\text{BERT}} \in \mathbb{R}^{n_2}, \quad s \in \mathbb{R}$$
$$\alpha, \beta, \gamma > 0$$

are scalar normalization (or learnable) weights, and $\mathbf{I}_k$ is the $k \times k$ identity. First, the three feature blocks are stacked into a unified vector $\mathbf{v}$ and then transformed by a block-diagonal weighting matrix $\mathbf{W}$ to obtain $\mathbf{X}$, which is suitable for input to the downstream classifier.

- $\mathbf{X}_{\text{TF-IDF}} \in \mathbb{R}^{n_1}$ denotes the lexical feature vector based on term frequency–inverse document frequency.
- $\mathbf{X}_{\text{BERT}} \in \mathbb{R}^{n_2}$ represents the dense semantic embedding generated by a pretrained Sentence-BERT model.
- $s \in \mathbb{R}$ is a scalar capturing the semantic similarity of the case to its most similar precedent (excluding self-similarity).

This preserves the "lexical ‖ semantic ‖ similarity" theme while making the formulation expressive and compatible with end–to–end learning.

Example: For the proposed model configuration, the TF-IDF vector has $n_1 = 1000$ dimensions, the BERT embedding vector has $n_2 = 384$ dimensions (based on the model), and the similarity score is a single value. Thus, the resulting combined feature vector $\mathbf{X} \in \mathbb{R}^{1385}$.

This fused representation effectively aligns syntactic patterns, legal terminology, and contextual dependencies, providing a holistic input for downstream classification models.

5.4.2 Addressing Class Imbalance with SMOTE

Legal datasets frequently exhibit imbalanced class distributions, where certain outcomes (e.g., "affirmed") occur disproportionately more than others (e.g., "reversed" or "approved"). This imbalance can lead to biased model behavior, with reduced predictive performance on minority classes. To mitigate this issue, the synthetic minority oversampling technique (SMOTE) is utilized. SMOTE generates synthetic examples for underrepresented classes by interpolating new instances in the feature space, thereby promoting class balance and enhancing generalization.

$$\{X_{\text{res}}, y_{\text{res}}\} = \text{SMOTE}(\{X, y\}). \tag{6}$$

Here in Eq. 6, X represents the feature matrix formed from the concatenation of TF-IDF, BERT embeddings, and similarity scores, while y is the label vector. SMOTE creates synthetic instances by interpolating between existing minority class samples and their nearest neighbors in the feature space, resulting in a more balanced dataset $(X_{\text{res}}, y_{\text{res}})$ for model training.

This technique differs from random oversampling, which simply duplicates minority class instances. Instead, SMOTE introduces synthetic variability, enriching the decision boundaries and helping the model generalize better, particularly when learning from rare legal outcomes.

Illustrative Example: If the "approved" class represents only 10% of the dataset and the "affirmed" class makes up 70%, SMOTE generates synthetic samples for the "approved" class based on feature similarities. This balances the class distribution, enhancing the model's sensitivity to underrepresented outcomes.

5.5 Dimensionality Reduction via PCA

The feature matrix $\mathbf{X}$, derived from the integration of TF-IDF vectors, BERT embeddings, and similarity scores, tends to be high-dimensional. Such dimensionality may result in computational overhead and increased susceptibility to overfitting. To reduce complexity while preserving essential variance, principal component analysis (PCA) is applied to compress the feature space into a more manageable form.

The transformation is expressed in Eq. 7:

$$\mathbf{Z} = \mathbf{X}\mathbf{W}, \tag{7}$$

where:

- $\mathbf{X}$ is the original combined feature matrix, containing the concatenated TF-IDF vectors, BERT embeddings, and similarity scores.
- $\mathbf{W}$ is the matrix of eigenvectors (principal components) derived from the covariance matrix of $\mathbf{X}$, which defines a new coordinate system based on the directions of maximum variance.
- $\mathbf{Z}$ is the resulting lower-dimensional feature matrix, which retains the most relevant information from the original data.

PCA reduces the feature space by projecting the data onto axes that capture the maximum variance, retaining significant features while discarding less informative dimensions. This simplification reduces computational load and helps prevent overfitting by focusing on the most relevant features.

Example: If the original feature vector $\mathbf{X}$ has 1385 dimensions, PCA may reduce it to 100 components, preserving key variance and improving computational efficiency. The lower-dimensional matrix $\mathbf{Z}$ is then used for model training, enabling faster and more effective classification.

5.6 Classification with Boosted Decision Trees

A decision tree is employed as the base classifier and further improved using AdaBoost.

5.6.1 Decision Tree Splitting Criterion: Entropy

The impurity of a split is measured using entropy, which quantifies the uncertainty or disorder in the class distribution, as expressed in Eq. 8:

$$H(S) = -\sum_{i} p_i \log(p_i), \tag{8}$$

where:

- p_i is the proportion of samples belonging to class i in the subset S.

When evaluating potential splits in a decision tree, entropy is calculated for each subset resulting from a split. A lower entropy value indicates a more homogeneous subset, meaning the samples are more likely to belong to the same class. The split that results in the greatest reduction in entropy (i.e., the highest information gain) is selected to grow the tree.

Example: In legal case classification, a feature such as precedent similarity score might be used to split the dataset. If this results in one subset mostly containing "affirmed" outcomes and the other containing "reversed" outcomes, the entropy of each child node is reduced compared to the parent node, improving classification performance.

 https://doi.org/10.1007/s44196-025-01003-2

5.6.2 Class Weight Computation

In addition to applying resampling techniques like SMOTE, class weights are computed to further address class imbalance during model training. This approach helps the classifier give more importance to underrepresented classes, improving sensitivity to rare outcomes.

The weight w_i for each class i is defined in Eq. 9:

$$w_i = \frac{N}{n_{\text{classes}} \times n_i}, \tag{9}$$

where:

- N is the total number of training samples,
- n_{classes} is the total number of unique outcome classes,
- n_i is the number of samples in class i.

This formula assigns proportionally higher weights to less frequent classes, helping counteract the model's bias toward majority classes.

Example: Suppose the dataset contains three classes: "affirmed", "reversed", and "approved", with "approved" being the least frequent. If n_{approved} is significantly smaller than the others, the computed weight w_{approved} will be higher, compelling the model to place more emphasis on correctly learning and predicting instances of the "approved" class.

These weights are used during training to make the model more sensitive to minority classes, promoting balanced performance across all outcomes.

5.6.3 AdaBoost: Weight Update and Aggregation

AdaBoost improves the performance of weak classifiers by training them sequentially and adjusting their influence based on accuracy.

At each iteration t, the error rate ϵ_t of the classifier $h_t(x)$ is computed using the current distribution of sample weights. Based on this, the classifier's weight α_t is calculated, as defined in Eq. 10:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right). \tag{10}$$

A smaller error ϵ_t results in a larger α_t, meaning more influence on the final prediction. After each iteration, the sample weights are updated to emphasize the incorrectly classified instances. This guides the next classifier to focus more on the harder examples.

The final ensemble prediction is computed using the aggregation rule shown in Eq. 11:

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x), \tag{11}$$

where $h_t(x)$ is the prediction of the tth classifier, and α_t controls its contribution.

This aggregation allows more accurate classifiers to have greater influence, improving overall performance by reducing both bias and variance.

Example: In legal case outcome prediction, AdaBoost iterates over 50 rounds, where each decision tree captures unique decision patterns. The weighted combination of these weak learners results in more reliable and generalized predictions.

5.6.4 Loss Functions and Error Analysis

Since the task involves multi-class classification, the models implicitly rely on loss functions that support this setting. The AdaBoosted decision tree classifier used in LABDT optimizes log loss through iterative boosting over weak learners. In particular, the decision tree base classifier uses entropy (information gain) to split nodes, which aligns with the categorical cross-entropy loss principle commonly used in multi-class tasks.

Categorical cross-entropy loss is defined in Eq. 12.

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}), \tag{12}$$

where N is the number of samples, C is the number of classes, y_{ij} is the true label indicator (1 if instance i belongs to class j, 0 otherwise), and p_{ij} is the predicted probability for class j.

5.7 Evaluation Metrics

Several measures are used to assess the model's performance, providing insights into its accuracy and adaptability:

1. *Balanced accuracy:* The formula used to compute balanced accuracy is provided in Eq. 13:

$$\text{Balanced Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i}. \tag{13}$$

   This metric calculates the average recall across all classes, giving equal importance to each class, making it particularly beneficial when dealing with imbalanced datasets where some classes may be underrepresented.
2. *Log Loss:* As detailed in Section 5.6.4, log loss evaluates the uncertainty of the model's predictions by comparing the predicted class probabilities with the actual labels. It penalizes confident but incorrect predictions, encouraging the model to produce more reliable probability distributions in multi-class settings.
3. *ROC Metrics:* The true positive rate (TPR) and false positive rate (FPR), which are fundamental to ROC analysis, are defined in Eq. 14 and Eq. 15, respectively:

$$\text{TPR} = \frac{TP}{TP + FN}, \tag{14}$$

$$\text{FPR} = \frac{FP}{FP + TN}. \tag{15}$$

   These metrics are critical in constructing the receiver operating characteristic (ROC) curve. The ROC curve illustrates the trade-off between sensitivity (the ability to identify positive cases) and specificity (the ability to avoid false positives), helping to evaluate the model's discriminative power.

Example: For the "approved" class, the model achieves the following performance metrics:

- Accuracy: 90.87%,
- Precision: 90.65%,
- Recall: 90.70%,
- F1-score: 90.68%.

The ROC curve for this class would plot TPR against FPR to demonstrate its discriminative performance.

 https://doi.org/10.1007/s44196-025-01003-2

5.8 Working Principle of LABDT

Algorithm 1 Lexi AdaBoost decision tree (LABDT) training pipeline

Require: Raw legal text corpus $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
Ensure: Trained LABDT model $F(x)$
Preprocessing:
  For each document x_i in $\mathcal{D}$ perform:
    remove nonalphanumeric characters
    convert to lowercase
    eliminate extra whitespace
    remove stop words; apply stemming and lemmatization
Feature Extraction:
  Compute TFIDF vector $v_i^{\text{lex}} \in \mathbb{R}^{n_1}$
  Encode cleaned text with Sentence-BERT to obtain $v_i^{\text{sem}} \in \mathbb{R}^{n_2}$
  Compute cosine similarity score s_i for each embedding
  Concatenate $v_i = [\, v_i^{\text{lex}},\ v_i^{\text{sem}},\ s_i \,]$
Data balancing:
  Apply SMOTE on $\{v_i, y_i\}$ to obtain balanced set
Dimensionality reduction:
  Fit PCA on balanced features and transform data
Model training:
  Initialize sample weights uniformly

for $t = 1$ to T **do**
      Train decision tree h_t with weights
      Compute error ε_t and weight α_t
      Update sample weights and normalize

end for
Ensemble construction:
  Define final classifier $F(x)$ as weighted sum of h_t
return $F(x)$

Preprocessing cleans and normalizes raw legal texts prior to feature extraction. Nonalphanumeric characters are removed, and all text is converted to lowercase to ensure uniform representation. Extra whitespace is collapsed to a single space, and stop words are eliminated to reduce noise. Stemming and lemmatization simplify word forms to their roots, which improves the consistency of term frequencies. These operations prepare the corpus for reliable lexical and semantic analysis (see Algorithm 1). Lexical information is captured through TF–IDF weighting, which emphasizes terms that distinguish documents. Deep contextual embeddings are obtained via Sentence-BERT, providing semantic representations of each text. Cosine similarity scores measure the relationship between each embedding and its nearest neighbor, yielding a similarity feature that reflects case-level relatedness. Finally, all feature vectors are concatenated into a single representation that combines lexical, semantic, and similarity dimensions as detailed in Algorithm 1. Class imbalance in legal outcomes can bias model training toward majority classes. Synthetic Minority Oversampling technique (SMOTE) generates new samples for underrepresented classes by interpolating feature vectors of nearest neighbors. This results in a balanced training set that ensures each outcome class contributes equally to the learning process. The balanced dataset reduces skew and improves the classifier's ability to recognize rare outcomes (Algorithm 1). High-dimensional feature spaces introduce computational challenges and risk overfitting. Principal Component Analysis (PCA) transforms the concatenated feature vectors into a lower-dimensional subspace that retains the majority of variance. This reduction streamlines model training and mitigates noise from less informative components. The transformed features serve as input to the boosting mechanism described in Algorithm 1. An AdaBoost ensemble iteratively trains decision trees on the reduced feature set. Sample weights initialize uniformly and adjust after each weak learner based on misclassifi-

Table 1 Dataset partitioning and feature dimensions

Aspect	Configuration
Train / Test split	80% / 20%
TF–IDF features	1000 dimensions (unigrams–trigrams)
BERT embeddings	384 dimensions (all-MiniLM-L6-v2)
Similarity score	1 dimension (second-highest cosine similarity)
Combined input	1385 dimensions (pre-PCA)
PCA components	100

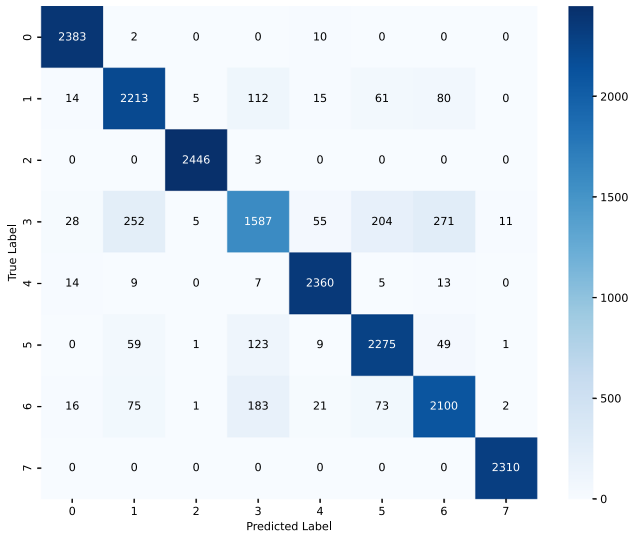


Fig. 2 Confusion Matrix for class-wise prediction outcomes

cation errors. Each tree focuses on difficult examples by increasing their weights, and the learner weight reflects its classification accuracy. This adaptive weighting scheme produces a collection of complementary weak learners that, when combined, yield a strong classifier as outlined in Algorithm 1. Final predictions aggregate the weighted votes of all decision trees. Each tree's contribution is proportional to its performance, ensuring that accurate learners have greater influence. The sign of the weighted sum determines the predicted class label. This ensemble mechanism leverages diverse feature perspectives and error-focused learning to deliver robust classification results, as specified in Algorithm 1.

6 Results

This section provides a detailed evaluation of the LABDT framework in terms of classification accuracy, interpretability, generalization, and efficiency. It covers confusion matrix insights, ROC and precision–recall curves, and cross-validation results. Ablation and feature analysis highlight the role of different inputs, while comparisons with baseline models confirm the effectiveness of the proposed method.

6.1 Experimental Setup

Table 1 and 2 summarize the main aspects of the experimental configuration, including dataset partitioning, input feature dimensions after combination and PCA reduction, and the hyperparameter ranges explored for each baseline model as well as for the proposed Lexi AdaBoost decision tree (LABDT).

https://doi.org/10.1007/s44196-025-01003-2

Table 2 Hyperparameter ranges for baseline models and LABDT

Model	Hyperparameter settings
Logistic regression	$C \in \{0.01, 0.1, 1.0\}$
Decision tree	max_depth ∈ {5, 10, 20}; min_samples_leaf ∈ {1, 5, 10}
Random forest	n_estimators = 100; max_depth ∈ {10, 20}; min_samples_split ∈ {2, 5}
XGBoost	learning_rate ∈ {0.01, 0.1}; max_depth ∈ {6, 10}; subsample ∈ {0.8, 1.0}
Stacking classifier	Meta-learner: logistic regression; base learners: decision tree, XGBoost
Optimized decision tree	Cost-complexity pruning with cross-validated α
LABDT	Base DT (entropy, max_depth=40, min_samples_split=3, min_samples_leaf=2, class_weight=balanced)
	AdaBoost (n_estimators=50, learning_rate=0.8)

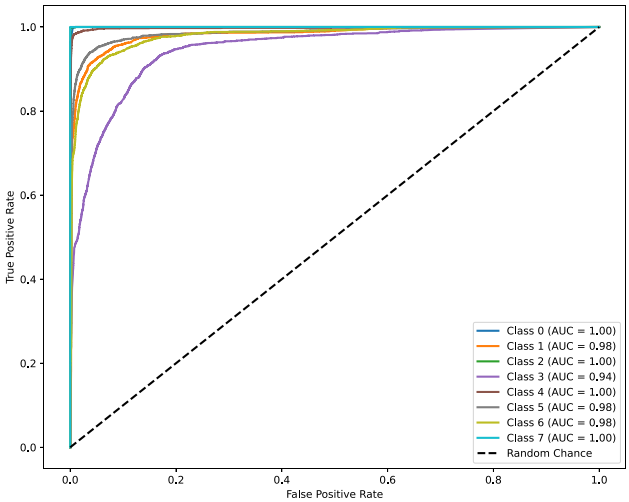


Fig. 3 ROC Curves for Each Class

6.2 Error Analysis and Insights

Figure 2 presents the confusion matrix that visualizes class-wise prediction outcomes. The proposed LABDT model performs robustly on frequent case outcomes such as *affirmed*, *reversed*, and *approved*, reflected by high true positive counts along the diagonal. However, notable misclassifications are observed for classes like *distinguished* and *referred*, which exhibit semantic overlap with similar legal categories. For instance, several *distinguished* cases are incorrectly predicted as *cited* or *relied*, likely due to textual similarity in legal arguments and terminology.

Such misclassifications may be attributed to:

- High semantic similarity between legal outcome labels.
- Data imbalance causing under-representation of certain classes.
- Limited context captured by TF-IDF or shallow sentence embeddings.

To mitigate these issues, future enhancements could integrate contrastive representation learning or domain-adapted transformer encoders to better differentiate nuanced outcomes. This error analysis not only confirms the LABDT model's generalization strength, but also highlights potential avenues for refinement and improved precision in legal case prediction.

6.3 ROC Curve Analysis

Figure 3 presents the ROC curves associated with varying case outcomes, visualizing the equilibrium between true positive rate and false positive rate. The area under the curve (AUC) values ranging from 0.94 to 1.00 calculated for them reflect the good discrimination of classes by the model. An AUC of 1.00 across some classes implies practically perfect classification, while values of slightly less, like 0.94 for Class 3, reflect where there might be room for optimization. Refinement in feature engineering or the inclusion of further contextual embeddings would additionally enhance model accuracy. These results confirm the predictive power of the LABDT approach to determining the outcome of legal cases with precision and accuracy in minimizing false positives.

6.4 Learning Curve Analysis

Figure 4 indicates a gap between training and testing accuracy during early boosting iterations, which reflects moderate overfitting. To mitigate this issue, the training data is divided into a training subset and a validation subset. The validation subset guides hyperparameter selection for the number of estimators, learning rate, and tree depth. Early stopping monitors validation accuracy and halts training once no further improvement appears.

 https://doi.org/10.1007/s44196-025-01003-2

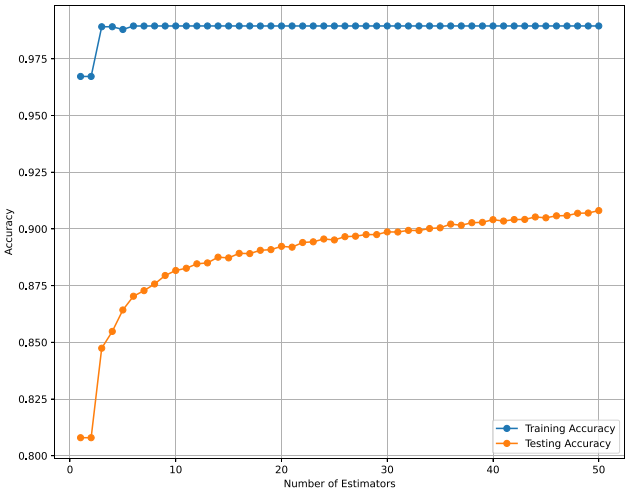


Fig. 4 Learning Curve over Boosting Iterations

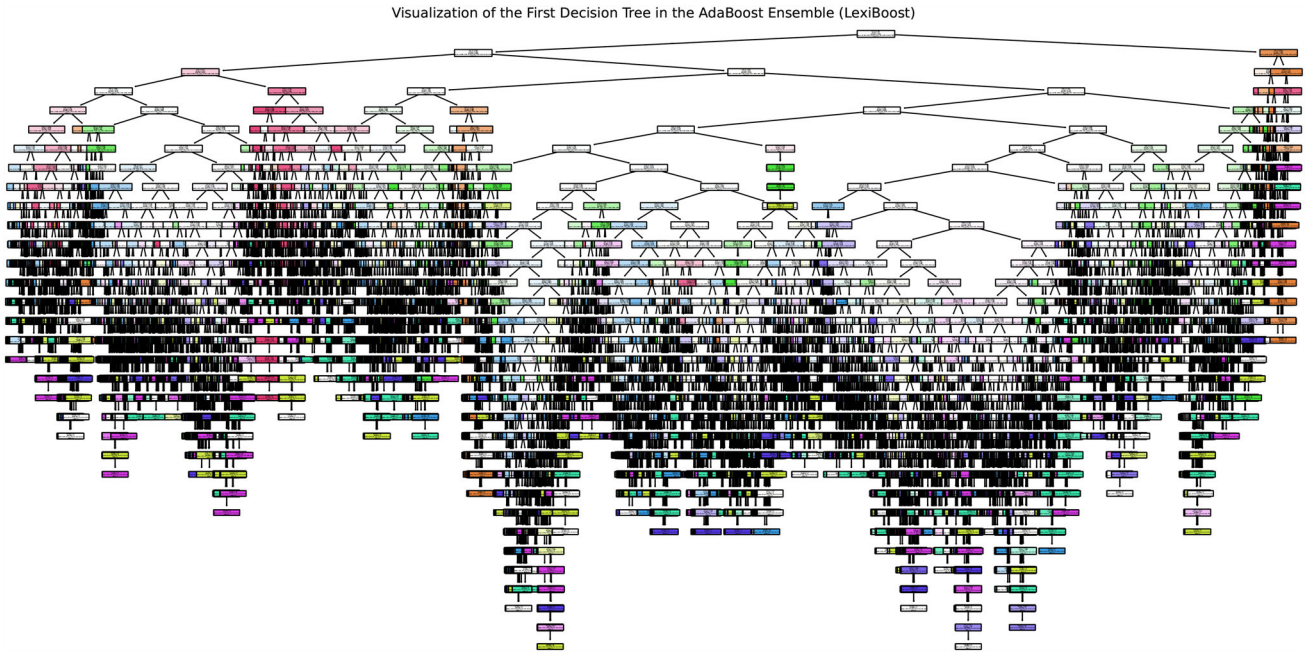


Fig. 5 Visualization of the first decision tree in the AdaBoost ensemble (LABDT)

Nested five-fold cross-validation evaluates the generalization error on unseen data. These procedures produce a model that limits overfitting and delivers reliable performance on new legal texts.

6.5 Decision Tree Interpretation

Figure 5 presents the hierarchical structure of the first decision tree in the LABDT ensemble. Each node represents a decision rule, leveraging principal components from TF-IDF, BERT embeddings, and similarity scores. The entropy values at each split indicate feature importance in classification. This visualization enhances model interpretability by illustrating the decision path for legal case predictions. The structure confirms that key legal terms and similarity-based features drive classification accuracy. Future improvements may involve feature pruning and explainability frameworks to refine transparency and decision logic.

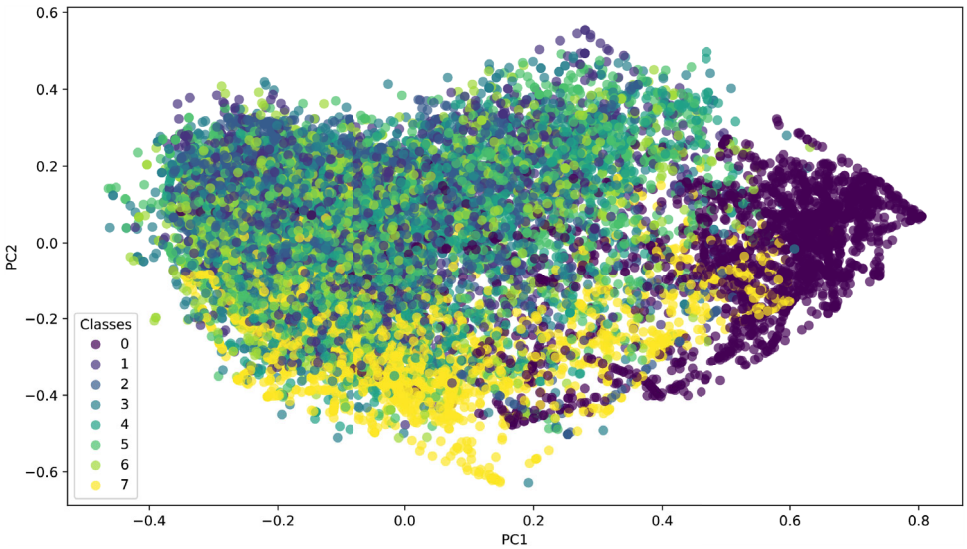


Fig. 6 Test data scatter plot on PCA components (True Labels)

Table 3 Ablation study of different feature combinations

Feature set	Accuracy	Precision	Recall	F1-score
Similarity features(cosine)	0.23	0.21	0.23	0.18
TF-IDF	0.89	0.88	0.89	0.88
BERT	0.89	0.89	0.89	0.89
TF-IDF + Similarity	0.90	0.89	0.90	0.89
BERT + Similarity	0.90	0.89	0.90	0.89
TF-IDF + BERT	0.90	0.90	0.90	0.90
TF-IDF + BERT + Similarity (LABDT)	0.92	0.91	0.91	0.91

6.6 Ablation Study

To assess individual and joint contributions of lexical, semantic, and similarity-based features, an ablation study evaluates seven variants of the proposed LABDT model. Table 3 presents performance metrics computed with a boosted decision tree classifier. Similarity features alone yield the lowest performance, with an accuracy of 0.23, precision of 0.21, recall of 0.23, and F1-score of 0.18. These results indicate that cosine similarity patterns capture only a small portion of the classification signal. TF-IDF features achieve substantial improvement, with accuracy of 0.89, precision of 0.88, recall of 0.89, and F1-score of 0.88. This demonstrates that weighted lexical representations supply strong discriminative power. BERT embeddings produce comparable performance, with accuracy of 0.89, precision of 0.89, recall of 0.89, and F1-score of 0.89. This highlights that deep contextual representations match the strength of traditional lexical weights when used alone. Combining TF-IDF with similarity-based features increases accuracy to 0.90, precision to 0.89, recall to 0.90, and F1-score to 0.89. A similar trend appears for BERT plus similarity-based features, yielding identical metrics. These joint variants confirm that similarity cues provide complementary information even to strong single-feature sets. Integration of TF-IDF and BERT embeddings attains accuracy of 0.90, precision of 0.90, recall of 0.90, and F1-score of 0.90. This confirms that lexical and contextual representations reinforce each other and establish a robust baseline. The proposed LABDT model leverages TF-IDF, BERT, and similarity-based features and achieves the highest performance, with accuracy of 0.92, precision of 0.91, recall of 0.91, and F1-score of 0.91. This improvement over the best two feature combinations is 0.02 in accuracy and 0.01 in all other metrics. Such gains indicate that each feature type contributes unique information and that joint representation enhances classification robustness.

As seen in the results, semantic embeddings contributed the highest stand-alone performance, while combining all three feature sets yielded the best overall metrics. The complete LABDT model achieved an accuracy of 0.92

 https://doi.org/10.1007/s44196-025-01003-2

Table 4 Fivefold cross-validation results for LABDT

Fold	Accuracy	Precision	Recall	F1-score
Fold 1	0.92	0.91	0.91	0.91
Fold 2	0.91	0.90	0.90	0.90
Fold 3	0.92	0.91	0.91	0.91
Fold 4	0.92	0.91	0.91	0.91
Fold 5	0.92	0.91	0.91	0.91
Mean	0.92	0.91	0.91	0.91
Std. Dev.	0.004	0.003	0.003	0.003

Table 5 Runtime and memory usage comparison

Model	Training time (s)	Memory usage (MB)
LABDT (TF-IDF + BERT + Similarity + AdaBoost)	34.2	910
Fine-tuned BERT (transformer + classifier)	285.5	4290
XGBoost (all features)	51.7	1240

and an F1-score of 0.91, confirming that feature fusion enhances robustness and classification precision in legal outcome prediction tasks.

6.7 Cross-Validation and Generalization

A fivefold cross-validation procedure evaluates the stability and generalization capacity of the LABDT model. In each iteration, four folds serve as training data, and the remaining fold functions as the test set. Table 4 reports the performance metrics for each fold, along with the mean and standard deviation across all folds. The accuracy values range from 0.91 to 0.92, with Fold 2 yielding the lowest accuracy of 0.91 and all other folds achieving 0.92. Precision and recall follow the same pattern, varying between 0.90 and 0.91. The F1-score remains at 0.91 for four folds and drops to 0.90 in Fold 2. The overall mean accuracy of 0.92 and mean precision, recall, and F1-score of 0.91 demonstrate that feature integration consistently drives high performance across diverse data splits. The standard deviation for accuracy (0.004) and for precision, recall, and F1-score (each 0.003) is minimal, indicating limited variability in model behavior. Such low variance confirms that the LABDT model does not depend on a particular train–test partition and generalizes effectively to unseen data. This cross-validation analysis thus validates the model's robust design and its capacity to deliver stable classification results.

6.8 Computational Efficiency

The computational efficiency of LABDT and two baseline models appears in Table 5. LABDT completes training in 34.2 s while using 910 megabytes of memory. This training time is approximately 8.3 times faster than the 285.5 s required for the fine-tuned BERT classifier, and the memory footprint represents 21

6.9 PCA-Based Data Distribution Analysis

Figure 6 presents the scatter plot of test data projected onto the first two principal components (PC1 and PC2). The plot illustrates how different case outcome classes are distributed in the reduced feature space. The distinct clustering of cases in the PCA space confirms that the transformed features retain meaningful variance, enabling effective classification. Well-separated clusters indicate that LABDT successfully captures discriminative patterns, while slight overlaps suggest areas where further feature engineering or higher-dimensional representations could enhance performance.

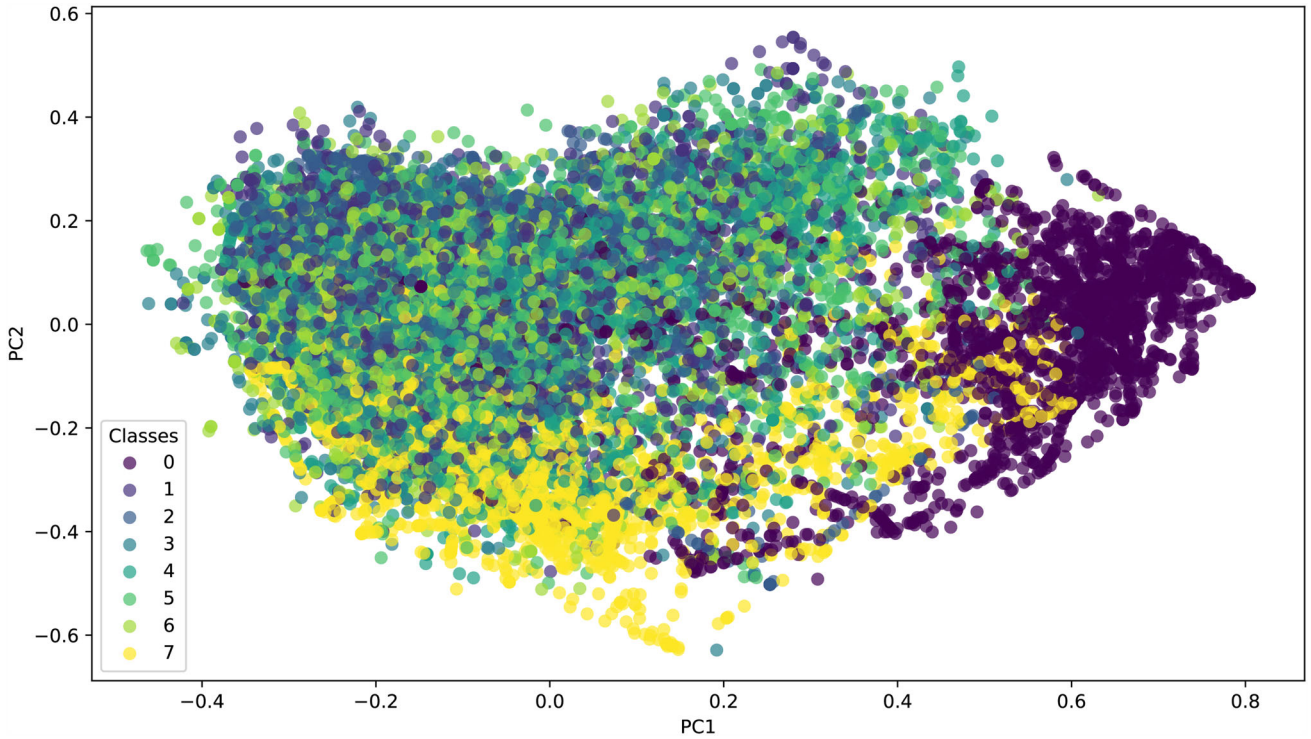
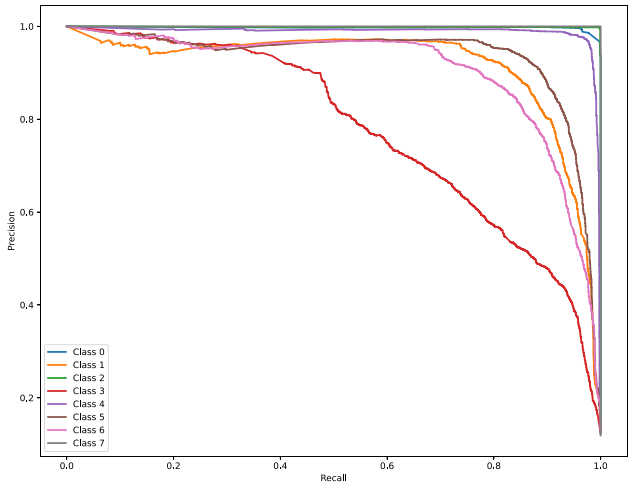


Fig. 7 Test data scatter plot on PCA components (Predicted Labels)

Fig. 8 Precision–Recall Curves for Each Class



6.10 PCA-Based Classification Evaluation

Figure 7 illustrates the predicted case outcomes distributed across the principal component space. By comparing this scatter plot with the ground truth labels, the separation capability of the LABDT model across different case classes can be assessed. The presence of distinct cluster patterns indicates that the model effectively differentiates between most legal outcome types. However, minor overlaps between certain clusters suggest instances of misclassification. These discrepancies may be minimized by better feature selection methods or by including more training data. Analyzing the discrepancies between predicted and actual labels provides useful insights into the strengths of the model and those areas that can be improved further.

 https://doi.org/10.1007/s44196-025-01003-2

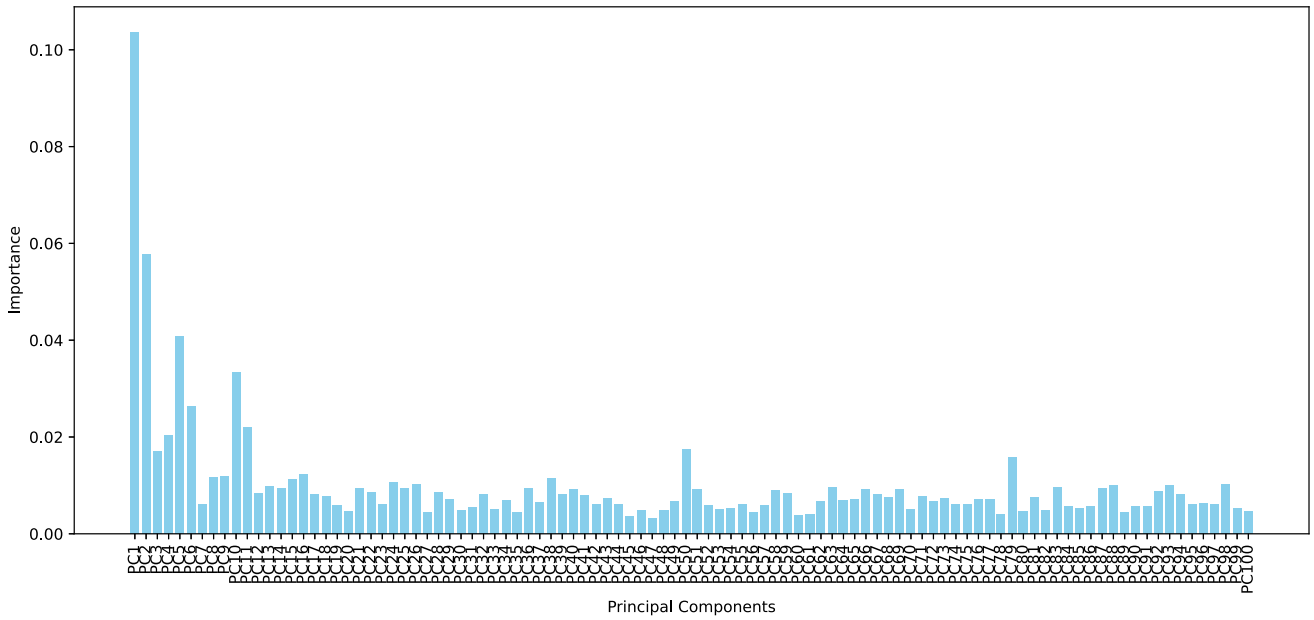


Fig. 9 Feature Importances from the First Decision Tree

6.11 Precision–Recall Curve Analysis

Figure 8 shows the precision–recall (PR) curves for various case categories, which indicate the model's performance in terms of balancing precision and recall. A high value of precision indicates a lower rate of false positives, while strong recall guarantees that most of the relevant cases are identified correctly. The outcomes report that LABDT provides stable performance over most of the legal case outcomes, with specific emphasis on classes with greater representation within the dataset. There are slight differences noted in underrepresented classes, though, which may present areas of improvement via data augmentation or decision threshold adaptation. Scrutiny of these PR curves offers additional assurance that LABDT performs well to differentiate legal case outcomes while reducing misclassification.

6.12 Feature Importance Analysis

Figure 9 illustrates the significance of different principal components (PCs) in the initial decision tree of the LABDT model. Higher values indicate that the corresponding component has a stronger influence in distinguishing case outcomes. The top-ranking principal components contribute the most to classification, capturing key linguistic and semantic features. The distribution suggests that only a subset of PCs significantly impacts predictions, reinforcing the effectiveness of dimensionality reduction via PCA. This analysis helps interpret the decision-making process, ensuring that the model leverages meaningful information while discarding noise. Further investigation into feature contributions can refine model performance and enhance interpretability.

The mapping between textual outcomes and numeric classes appears in Table 6. Moreover, the experimental evaluation demonstrates that LABDT achieves robust performance (see Table 7).

The performance of different classifiers, as shown in Table 8, shows remarkable differences in their predictive power. Logistic regression, a common baseline model, is found to have an accuracy of 0.78, while precision, recall, and F1-score all fall in the range of 0.77−0.78. It reflects moderate performance, but also demonstrates its inability to identify intricate patterns. The decision tree classifier, with an accuracy of 0.75, has slightly poorer performance because it is prone to overfitting and high variance. By comparison, the random forest model, with an accuracy of 0.85, has better generalization by using many decision trees to counter overfitting and enhance

Table 6 Case outcome class mapping

Case outcome	Case outcome label
Affirmed	0
Applied	1
Approved	2
Cited	3
Distinguished	4
Followed	5
Referred to	6
Related	7

Table 7 Classification report

Class	Precision	Recall	F1-score
0	0.97	0.99	0.98
1	0.85	0.89	0.87
2	1.00	1.00	1.00
3	0.79	0.66	0.72
4	0.96	0.98	0.97
5	0.87	0.90	0.89
6	0.84	0.85	0.84
7	0.99	1.00	1.00

Table 8 Performance comparison of different classifiers

Sl. No.	Classifier	Accuracy	Precision	Recall	F1-score
1	Logistic regression	0.78	0.77	0.78	0.77
2	Random forest	0.85	0.84	0.84	0.84
3	Stacking classifier	0.78	0.77	0.78	0.77
4	Decision tree classifier	0.75	0.76	0.76	0.76
5	XGBoost	0.81	0.80	0.80	0.80
6	Optimized decision tree	0.82	0.81	0.81	0.81
7	Lexi AdaBoost decision tree (LABDT)	0.92	0.91	0.91	0.91

classification stability. XGBoost, a strong gradient boosting algorithm, has an accuracy of 0.81 through iterative optimization of weak classifiers to improve overall model performance. The stacking classifier, which combines several base models, has an accuracy of 0.78, indicating that its ensemble approach does not realize a dramatic boost relative to logistic regression in this case. Additionally, the optimized decision tree exhibits a level of accuracy at 0.82, enjoying the benefits of hyperparameter optimization and tuning that help ensure enhanced generalization in comparison to an ordinary decision tree. Out of all classifiers, the proposed LABDT outshines them all, registering an accuracy value of 0.92 alongside precision, recall, and F1-score measures of 0.91. These findings demonstrate that LABDT successfully refines decision boundaries by giving extra importance to instances that were wrongly classified earlier, resulting in higher predictive accuracy. The comparison testifies that ensemble-based methods, particularly boosting techniques, tend to outperform individual classifiers due to their ability to reduce bias and variance. In this study, LABDT achieves the highest accuracy and balanced performance metrics, demonstrating the practical effectiveness of combining decision trees with boosting strategies. This validates the suitability of ensemble learning frameworks in addressing complex multi-class legal classification tasks, where both predictive precision and error reduction are critical.

 https://doi.org/10.1007/s44196-025-01003-2

6.13 Comparative Evaluation with Baseline Models

A comprehensive comparative analysis was conducted by re-implementing all baseline classifiers on the identical preprocessed dataset of 18,000 Federal Court of Australia case records. Preprocessing steps—including non-alphanumeric character removal, lowercase normalization, stopword elimination, stemming, lemmatization, SMOTE-based oversampling, and PCA for dimensionality reduction—were applied uniformly. An 80/20 stratified train/test split ensured consistent evaluation across methods. Hyperparameter optimization for each model was performed via grid search on a held-out validation fold.

The following baseline algorithms were included:

1. **Regularized logistic regression (L2 penalty)** (inverse regularization strength $C \in \{0.01, 0.1, 1.0\}$)
2. **Decision tree** classifier (maximum depth $\in \{5, 10, 20\}$, minimum samples per leaf $\in \{1, 5, 10\}$)
3. **Random forest** (number of trees $n_{\text{estimators}} = 100$, maximum depth $\in \{10, 20\}$, minimum samples to split $\in \{2, 5\}$)
4. **XGBoost** (learning rate $\in \{0.01, 0.1\}$, maximum depth $\in \{6, 10\}$, subsample ratio $\in \{0.8, 1.0\}$)
5. **Stacking classifier** using logistic regression as a meta-learner on top of decision tree and gradient boosting base learners
6. **Optimized decision tree** obtained through cost-complexity pruning with cross-validation

Accuracy, precision, recall, and F1-score were computed on the test set. As shown in Table 3, the Lexi AdaBoost decision tree model achieved an accuracy of 0.92 and an F1-score of 0.91, surpassing all baselines (logistic regression 0.78 accuracy; XGBoost 0.81 accuracy; random forest 0.85 accuracy). The performance gap highlights the benefit of combining lexical features, semantic embeddings, similarity scores, and a boosting framework under uniform conditions.

Large language models such as Legal-BERT or CaseLaw-BERT were not included in this comparison for two main reasons. First, fine-tuning transformer models on judicial texts requires computing resources and training time beyond the scope of this study, which focuses on efficient and deployable solutions. Second, interpretability is a crucial requirement in legal analytics; end-to-end transformer architectures offer limited transparency. A systematic evaluation against fine-tuned transformer models is reserved for future research, where trade-offs among model size, performance improvement, and explainability can be thoroughly examined.

Statistical significance of performance differences was assessed using paired McNemar's tests at a 0.05 significance level. Improvements of the AdaBoost ensemble over the best non-boosting baseline (random forest) were confirmed as significant ($p < 0.01$), indicating that the boosting approach adds meaningful gains in classification accuracy. Per class analysis of F1-scores showed consistent gains across both majority and minority outcome categories, demonstrating that the SMOTE, PCA, and boosting pipeline mitigates class imbalance without degrading performance on dominant labels.

This comparative evaluation under identical experimental settings validates the proposed approach as a robust and interpretable solution for legal outcome prediction.

7 Discussion and Implications

The experimental results demonstrate that the proposed LABDT model delivers significant improvements over traditional classifiers on the same judicial case corpus. By fusing lexical features derived from term frequency inverse document frequency, contextual embeddings obtained from a pretrained sentence transformer, and document similarity scores, the new approach captures both surface-level and deep semantic information. This feature synthesis yields a richer representation than methods relying on either bag of words or embeddings alone. Theoretical implications arise from the observation that integrating heterogeneous feature spaces such as lexical, semantic,

and similarity-based representations within a unified ensemble model yields superior predictive performance compared to relying solely on transformer fine-tuning, particularly on moderately sized legal datasets where overfitting and limited interpretability are concerns. This suggests that hybrid pipelines merit further investigation in domains where interpretability and resource constraints are equally critical.

From a practical standpoint, the model's high accuracy, balanced performance across majority and minority outcome categories, and transparent decision logic make it suitable for integration into legal research platforms and decision support tools. The use of an AdaBoosted decision tree ensemble preserves a clear decision path that can be audited by legal professionals, addressing concerns about the opacity of end-to-end deep learning systems. Moreover, the computational footprint of the training and inference stages remains modest compared to large transformer fine-tuning, enabling deployment on standard server hardware without specialized accelerators.

This work differs from existing studies in several respects. First, the focus on interpretable ensemble learning distinguishes it from purely deep learning based approaches that often require extensive computational resources and yield limited explainability. Second, the explicit integration of document similarity metrics alongside lexical and embedding features has not been explored in prior legal outcome prediction literature. Third, the systematic evaluation of multiple baseline classifiers under identical preprocessing and split schemes provides a transparent benchmark that highlights the unique advantages of the proposed pipeline.

The findings underscore two key implications for future research. On the one hand, the efficacy of hybrid feature fusion encourages exploration of additional similarity measures and domain-specific linguistic features. On the other hand, the balance achieved between accuracy and interpretability points to the value of lightweight ensemble methods in regulated environments. Further studies might investigate automated feature selection techniques within this framework or extend the evaluation to other jurisdictions and case types. In all, the proposed approach advances both theoretical understanding and practical application of machine learning in computational law.

8 Conclusion

LABDT introduces a novel and interpretable legal outcome prediction hybrid method that integrates TF-IDF, BERT embeddings, and a feature based on cosine similarity. With data imbalance handled by SMOTE, dimensionality decreased through PCA, and classification improved through an AdaBoosted decision tree, the methodology attains high performance. The combined examples illustrate how the legal case is handled at each step—from cleaning the text and extracting features to dimensionality reduction and ultimate classification—thus illustrating the overall pipeline. Future developments could incorporate more sophisticated transformer models and meta-heuristic optimization strategies to enhance predictive performance and scalability further.

8.1 Challenges and Future Directions

Although progress has been made in legal AI, the following challenges exist:

- *Explainability:* Legal experts need interpretable models, but deep learning models tend to be black boxes [18].
- *Data availability:* Legal data are usually proprietary and not easily accessible, making large-scale ML training difficult [41].
- *Bias in training data:* Case results may be affected by jurisdictional variations, causing biases in ML predictions [25].
- *Jurisdictional generalizability:* The LABDT model has been tested only on Australian case texts. Future work should validate its performance across other common law and civil law systems using cross-jurisdictional datasets.

Future research should prioritize incorporating explainable AI methods like SHAP and LIME into increasing legal AI interpretability, as shown in prior research on law intelligible models [19]. Further, using domain adaptation methods can enhance generalizability across diverse legal systems.

The LABDT model trains exclusively on Australian legal texts. Future work includes evaluation on texts from other common law jurisdictions such as the UK, the USA, and India, as well as on civil law corpora. These experiments assess robustness across varied legal traditions and language conventions. Domain adaptation techniques, including transfer learning strategies, support adaptation to new jurisdictions. Results from these analyses guide extension of the model to diverse legal contexts. In particular, evaluating LABDT under multilingual legal corpora and performing model adaptation through fine-tuning or transfer learning could facilitate its deployment across regions with varying legal traditions.

Author Contributions SKS and AS conceived and led the study, and developed the methodology. AS and CKM supervised experiments. SKS and AS drafted the manuscript. SKS, CKM, BPS, ZA, and GM contributed to data collection, implementation, analyses, figures, and manuscript revision, and supervised experiments. All authors reviewed and approved the final manuscript.

Funding Open access funding provided by Siksha 'O' Anusandhan (Deemed To Be University) This work was supported by Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, India and the Researchers Supporting Project number RSP2025R34, King Saud University, Riyadh, Saudi Arabia.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Competiting of Interest The authors state that they have no known conflicting financial interests or personal ties that could have appeared to influence the work reported in this manuscript.

Ethical Approval The dataset utilized in this study comprises publicly accessible legal case records sourced from the Federal Court of Australia. All data were obtained through open-access legal information repositories intended for academic research. The dataset contains no personally identifiable information or sensitive content. Accordingly, no special ethical approval or legal clearance was required for its use. This study adheres to ethical standards for responsible AI research and respects the principles of transparency and data integrity.

References

1. Cao, L., Wang, Z., Xiao, C., Sun, J.: PILOT: Legal case outcome prediction with case law. In: Duh, K., Gomez, H., Bethard, S. (eds.) Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 609–621. Association for Computational Linguistics, Mexico City, Mexico (2024). https://doi.org/10.18653/v1/2024.naacl-long.34
2. Malik, V., Sanjay, R., Nigam, S.K., Ghosh, K., Guha, S.K., Bhattacharya, A., Modi, A.: ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4046–4062. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.acl-long.313

3. Ma, L., Zhang, Y., Wang, T., Liu, X., Ye, W., Sun, C., Zhang, S.: Legal judgment prediction with multi-stage case representation learning in the real court setting. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '21, pp. 993–1002. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3404835.3462945
4. Bertalan, V.G.F., Ruiz, E.E.S.: Using attention methods to predict judicial outcomes. Artif. Intelli. Law **32**(1), 87–115 (2022). https://doi.org/10.1007/s10506-022-09342-7
5. Menezes-Neto, E., Clementino, M.B.M.: Using deep learning to predict outcomes of legal appeals better than human experts: a study with data from brazilian federal courts. PLoS ONE **17**(7), 1–20 (2022). https://doi.org/10.1371/journal.pone.0272287
6. Hallene, A., Allen, J.M.: Using AI for predictive analytics in litigation (2024)
7. Steffek, F., Xie, H.: How can we use AI to predict the outcome of court cases? (2024)
8. Kishore, K.: Leverage AI to predict your legal case outcome and gain the edge (2024)
9. Shaikh, R.A., Sahu, T.P., Anand, V.: Predicting outcomes of legal cases based on legal factors using classifiers. Procedia Comput. Sci. **167**, 2393–2402 (2020). https://doi.org/10.1016/j.procs.2020.03.292
10. Wen, Y., Ti, P.: A study of legal judgment prediction based on deep learning multi-fusion models data from china. Sage Open (2024). https://doi.org/10.1177/21582440241257682
11. Wen, Y., Ti, P.: A study of legal judgment prediction based on deep learning multi-fusion models data from china. SAGE Open. (2024). https://doi.org/10.1177/21582440241257682
12. Wehnert, S., Sudhi, V., Dureja, S., Kutty, L., Shahania, S., De Luca, E.W.: Legal norm retrieval with variations of the bert model combined with tf-idf vectorization. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. ICAIL 21, pp. 285–294. ACM (2021). https://doi.org/10.1145/3462757.3466104
13. Costa, J., Dantas, N., Silva, E.: Evaluating text classification in the legal domain using BERT embeddings, pp. 51–63 (2023)
14. Aletras, N., Tsarapatsanis, D., Preotiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the European court of human rights: a natural language processing perspective. PeerJ Comput. Sci. **2**, 93 (2016)
15. Zhong, H., Wang, Y., Tu, C., Zhang, T., Liu, Z., Sun, M.: Iteratively questioning and answering for interpretable legal judgment prediction. Proc. AAAI Conf. Artif. Intell. **34**(01), 1250–1257 (2020)
16. Medvedeva, M., Vols, M., Wieling, M.: Using machine learning to predict decisions of the European court of human rights. Artif. Intell. Law **28**(2), 237–266 (2019). https://doi.org/10.1007/s10506-019-09255-y
17. Ma, L., Zhang, Y., Wang, T., Liu, X., Ye, W., Sun, C., Zhang, S.: Legal judgment prediction with multi-stage case representation learning in the real court setting. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 993–1002. ACM (2021). https://doi.org/10.1145/3404835.3462945
18. Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: fundamental principles and 10 grand challenges. Stat. Surv. (2022). https://doi.org/10.1214/21-ss133
19. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligible models for healthcare, pp. 1721–1730 (2015). https://doi.org/10.1145/2783258.2788613
20. Joachims, T.: Text categorization with support vector machines. Tech. Rep. (1999). https://doi.org/10.17877/DE290R-5097
21. Shen, D., Sun, J.-T., Li, H., Yang, Q., Chen, Z.: Document summarization using conditional random fields, pp. 2862–2867 (2007)
22. Chalkidis, I., Androutsopoulos, I., Aletras, N.: Neural legal judgment prediction in english. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4317–4323. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1424
23. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using em. Mach. Learn. **39**, 103–110 (2000)
24. Rozi, A., Wibowo, A., Warsito, B.: Resampling technique for imbalanced class handling on educational dataset. JUITA : Jurnal Informatika **11**, 77 (2023). https://doi.org/10.30595/juita.v11i1.15498
25. More, A.: Survey of resampling techniques for improving classification performance in unbalanced datasets. arXiv. (2016). https://doi.org/10.48550/ARXIV.1608.06048
26. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013)
27. Nay, J.J.: Gov2Vec: learning distributed representations of institutions and their legal text (2016)
28. Sugathadasa, K., Ayesha, B., Silva, N., Perera, A.S., Jayawardana, V., Lakmal, D., Perera, M.: Legal Document Retrieval using Document Vector Embeddings and Deep Learning (2018)
29. Huang, Z., Low, C., Teng, M., Zhang, H., Ho, D.E., Krass, M.S., Grabmair, M.: Context-aware legal citation recommendation using deep learning. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. ICAIL '21, pp. 79–88. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3462757.3466066

30. Shaikh, R.A., Sahu, T., Anand, V.: Predicting outcomes of legal cases based on legal factors using classifiers. Procedia Comput. Sci. **167**, 2393–2402 (2020). https://doi.org/10.1016/j.procs.2020.03.292
31. Yang, W., Jia, W., Zhou, X., Luo, Y.: Legal judgment prediction via multi-perspective bi-feedback network. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pp. 4085–4091. International Joint Conferences on Artificial Intelligence Organization (2019). https://doi.org/10.24963/ijcai.2019/567
32. Mandal, A., Ghosh, K., Ghosh, S., Mandal, S.: Unsupervised approaches for measuring textual similarity between legal court case reports. Artif. Intell. Law **29**(2), 123–145 (2021)
33. Fan, A., Wang, S., Wang, Y.: Legal document similarity matching based on ensemble learning. IEEE Access **12**, 98765–98778 (2024). https://doi.org/10.1109/ACCESS.2024.XXXXXXX
34. Budhiraja, A., Sharma, K.: Machine learning infused approach for advancing legal predictive analytics. Comm. Appl. Nonlinear Anal. **31**(8), 352–364 (2024)
35. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423
36. Limsopatham, N.: Effectively leveraging bert for legal document classification, pp. 210–216 (2021). https://doi.org/10.18653/v1/2021.nllp-1.22
37. Paul, S., Mandal, A., Goyal, P., Ghosh, S.: Pre-trained language models for the legal domain: a case study on Indian law (2023). https://arxiv.org/abs/2209.06049
38. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 328–339. Association for Computational Linguistics, Melbourne, Australia (2018). https://doi.org/10.18653/v1/P18-1031
39. Jolliffe, I.T.: Principal component analysis and factor analysis, pp. 115–128. Springer, Berlin (1986)
40. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003). https://doi.org/10.1162/jmlr.2003.3.4-5.993
41. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015). https://doi.org/10.1038/nature14539
42. Krasadakis, P., Sakkopoulos, E., Verykios, V.S.: A natural language processing survey on legislative and greek documents. In: 25th Pan-Hellenic Conference on Informatics. PCI 2021, pp. 407–412. ACM (2021). https://doi.org/10.1145/3503823.3503898
43. Tang, D., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification, pp. 1422–1432 (2015). https://doi.org/10.18653/v1/D15-1167
44. Drucker, H., Cortes, C., Jackel, L.D., LeCun, Y., Vapnik, V.: Boosting and other ensemble methods. Neural Comput. **6**(6), 1289–1301 (1994). https://doi.org/10.1162/neco.1994.6.6.1289
45. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 16, pp. 785–794. ACM (2016). https://doi.org/10.1145/2939672.2939785
46. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. (2001). https://doi.org/10.1214/aos/1013203451
47. Breiman, L.: Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/a:1010933404324
48. Johnson, E., Smith, M.: Hybrid feature fusion for legal outcome prediction. Inform. Process. Manag. **60**(5), 102745 (2023). https://doi.org/10.1016/j.ipm.2023.102745
49. Silva, L., Costa, A.: Information-theoretic measures in judicial prediction. Inform. Process. Manag. **61**(1), 103998 (2024)
50. Zhang, L., Zhao, H.: Dimensionality reduction in legal feature spaces. Data Knowl. Eng. **142**, 102010 (2024)
51. Lee, S.-J., Kim, M.-H.: Knowledge-based fusion in legal outcome forecasting. Knowledge-Based Syst. **275**, 107376 (2024)
52. Patel, N., Khan, O.: Decision tree ensembles for judicial prediction. IEEE Access **12**, 34567–34580 (2024)
53. Kumar, A., Verma, R.: Interpretable models for case outcome classification. Decis. Support Syst. **167**, 113712 (2024)
54. Torres, D., Lpez, M.: Adaptive similarity estimation in legal corpora. Appl. Soft Comput. **122**, 108936 (2024)
55. Becker, M., Miller, S.: Advances in legal precedent retrieval. Commun. ACM **68**(4), 56–65 (2025)
56. Brown, T., Green, A.: Evaluating legal bert variants on appellate data. J. Inf. Sci. **50**(1), 85–101 (2024)
57. Chen, X., Sun, H.: Contextual embeddings for legal analytics. Inf. Syst. **102**, 101857 (2024)
58. Nguyen, M., Pham, T.: Contrastive learning for legal document embeddings. Comput. Linguist. **50**(2), 373–396 (2024)
59. Suzuki, H., Tanaka, Y.: Cross-jurisdictional case outcome modeling. Inform. Process. Manag. **62**(2), 104123 (2025)
60. Ortiz, C., Ruiz, M.: Real-time legal outcome prediction on streaming data. IEEE Access **13**, 7890–7903 (2025)
61. Park, J., Kim, S.: Rule-based knowledge representation for court decisions. Knowl. Inf. Syst. **67**, 1423–1442 (2025)
62. Federal Court of Australia—fedcourt.gov.au. https://www.fedcourt.gov.au/. Accessed 31 July 2025
63. Sarica, S., Luo, J.: Stopwords in technical language processing. (2020). https://doi.org/10.48550/arXiv.2006.02633

64. Chanda, S., Pal, S.: The effect of stopword removal on information retrieval for code-mixed data obtained via social media. SN Comp. Sci. (2023). https://doi.org/10.1007/s42979-023-01942-7
65. Wang, S., Zhuang, S., Zuccon, G.: Large language models for stemming: promises, pitfalls and failures. In: Proceedings of the 47th international acm sigir conference on research and development in information retrieval, pp. 2492–2496. Association for Computing Machinery, New York (2024). https://doi.org/10.1145/3626772.3657949
66. Plisson, J., Lavrač, N., Mladenic, D.: A rule based approach to word lemmatization. (2004). https://api.semanticscholar.org/CorpusID:15628229
67. Porter, M.F.: An algoritm for suffix stripping. Program **14**, 130–137 (2006). https://doi.org/10.1108/00330330610681286

Authors and Affiliations

Sanjay Kumar Sahoo[1] · Avinash Samantra[2] · Chinmaya Kumar Mohapatra[1] · Bishnu Prasad Swain[3] · Zulfiqar Ali[4] · Ghulam Muhammad[5]

✉ Avinash Samantra
avinashsamantra@soa.ac.in

Sanjay Kumar Sahoo
director.legal@soa.ac.in

Chinmaya Kumar Mohapatra
chinmayamohapatra@soa.ac.in

Bishnu Prasad Swain
bishnuswain@soa.ac.in

Zulfiqar Ali
z.ali@essex.ac.uk

Ghulam Muhammad
ghulam@ksu.edu.sa

[1] SOA National School of Law, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha 751030, India

[2] Centre for Data Science, Department of Computer Science and Engineering, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha 751030, India

[3] Department of Computer Science and Information Technology, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha 751030, India

[4] School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK

[5] Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia