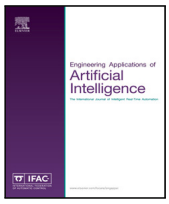
Engineering Applications of Artificial Intelligence 160 (2025) 112002

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Research paper

Integrated dual-level dependency analysis framework for multi-task judicial decision prognosis in murder cases

Prameela Madambakam, Himangshu Sarma *

Human Computer Interaction Lab, Department of Computer Science and Engineering, Indian Institute of Information Technology Sri City, Sri City, India

ARTICLE INFO

Keywords:
Multi-task dataset
Dependency analysis
Threshold modulation
Deep learning
Evidence and exceptions

ABSTRACT

Judicial decision prognosis involves predicting judicial outcomes based on the case facts and evidence, a critical yet challenging task in the legal domain. Despite its importance, there is a scarcity of multi-task datasets and frameworks specifically tailored to the Indian judicial system. To address this gap, this work introduces the Indian Judicial Multi-task (IJMT) dataset, developed to resolve murder cases by incorporating facts, evidence, and exceptions. Furthermore, this work proposes a multi-task decision-making framework that integrates a lexical-vector dependency analysis engine designed using deep learning techniques and applies it to the IJMT dataset. To address the issue of label imbalance prevalent in legal datasets, a novel context-aware threshold modulation strategy is developed that enables controlled adjustment and dynamic fine-tuning of thresholds. The framework's superior performance on the IJMT dataset compared to similar baselines leveraging dependency analysis between tasks validates its robustness and the dataset's applicability for real-world legal experiments. By bridging the gap in multi-task datasets for the Indian legal context and presenting a robust decision-making framework, this work provides a significant contribution to advance research in legal judgment prognosis.

1. Introduction

Judicial Decision Prognosis (JDP), also known as Legal Judgment Prediction (LJP), refers to the process of predicting judicial outcomes by analyzing case facts and evidence. It is a crucial task in the legal domain, aiding lawyers, judges, and policymakers in understanding potential case outcomes and improving decision-making efficiency. With advancements in computational models and datasets, this task has gained momentum as research strives toward creating systems capable of handling complex legal scenarios (Madambakam and Rajmohan, 2022). The main motivation is to leverage advanced technologies such as Deep Learning (DL) to provide a systematic solution for the backlog of pending court cases each year and expedite the judgment delivery process.

In the context of Indian law, the Indian Penal Code (IPC) plays a pivotal role in defining the legal framework to address criminal cases, including murder. The IPC provides comprehensive sections and clauses that specify offenses and penalties, such as prison and fine terms, forming the backbone of the judicial process in criminal justice. This work focuses on resolving criminal cases, particularly murder cases, by adhering to the procedures specified under the IPC, the Criminal Procedure Code (CrPC), and the Indian Evidence Act. Unlike criminal cases where laws are more structured, judgments in civil cases tend to vary significantly due to multiple pleas and heterogeneous outcomes (Madambakam et al., 2023).

For murder cases, the IPC outlines the conditions constituting murder while also accounting for exceptions like self-defense, provocation, or accidents. The detailed structure of the IPC helps in distinguishing between culpable homicide under sections 299 and 304 and murder under sections 300, 302, 304 A, and 304B, assigning appropriate sections to cases, and determining punishments such as imprisonment, fines, or both. Accurate application of these legal provisions requires an in-depth understanding of the facts, evidence, and exceptions presented in each case.

By integrating the IPC's structured legal principles with advancements in data-driven approaches, predictive frameworks can assist in resolving murder cases more effectively. These frameworks analyze facts, evidence, and legal exceptions to recommend applicable sections and predict outcomes, thereby supporting the judicial systems in handling complex cases with consistency and fairness.

Legal judgment prognosis for murder cases involves interdependent tasks such as identifying applicable law sections, determining prison and fine terms, etc. These tasks are inherently connected, requiring a multi-task learning approach to model their dependencies effectively.

As per the understanding, existing works lack a comprehensive analysis of the Indian legal judgment outcome format and the various

* Corresponding author.
E-mail addresses: prameela.m@iiits.in (P. Madambakam), himangshu.sarma@iiits.in, himangshu.tezu@gmail.com (H. Sarma).

https://doi.org/10.1016/j.engappai.2025.112002
Received 8 March 2025; Received in revised form 10 July 2025; Accepted 9 August 2025
Available online 19 August 2025

components involved in it. Unlike other international and domestic datasets, Indian judgments exhibit a complex structure comprising multiple components such as multiple applicable sections, diverse types of penalties: varying imprisonment terms including prison and default terms, along with their implementation nature, multiple fine terms, and lastly prison_type. These components are determined based on diverse case information, including facts, proven evidence, and exceptions raised.

In contrast, other multi-task datasets typically consist of single values for components like section, charge, prison, and fine terms, which are determined solely based on facts. There is a notable lack of sufficient research on accurately predicting fine terms. As far as we know, this is the first work conducting a detailed analysis of the complex Indian judgment outcome format encompassing various components and the dependencies between them.

Based on the above analysis, the objectives of this work are twofold: 1. To create a multi-task dataset tailored to the Indian legal context, incorporating evidence and exceptions along with the case facts. 2. To develop a multi-task model that predicts various judgment components based on the Indian judicial outcome format by analyzing the dependencies between the tasks. To the best of our knowledge, this is the first work addressing these objectives for the Indian legal system.

To achieve the first objective, a multi-task dataset is proposed in the English language, focusing primarily on murder cases and related sub-crime sections contributing to the crime. Existing datasets lacked a comprehensive coverage of all necessary judgment components in the Indian judicial format. This gap is addressed by the proposed Indian Judicial Multi-task (IJMT) dataset. This dataset extends case facts with gathered evidence and exceptions that influence the judgment, enabling recommendations for multi-task decision components.

These critical aspects for determining various judgment components are often overlooked in existing datasets in India and abroad. The IJMT dataset is the first of its kind in India, enabling multi-task predictions of legal judgment components in line with the Indian judicial format. It incorporates evidence and exceptions, such as mental illness, insanity, or considerations for sole family providers, to ensure a holistic analysis.

To satisfy the second objective, a multi-task decision-making sequential framework integrating a lexical-vector dependency analysis engine designed using DL techniques is introduced and applied to the IJMT dataset for judgment prediction. It is a sequential framework, as the prediction of each task relies on the outcomes of its preceding tasks in a sequential order. The model predicts four sets of judgment outcomes: applicable sections and their corresponding dependent tasks, including prison terms, fine terms, default terms, and prison_type. This is achieved by analyzing dual-level dependencies at the lexical and vector levels among the tasks, factoring in case facts, evidence, and exceptions.

The highlights of the proposed framework are,

Effectiveness: The framework establishes a new state-of-the-art in predicting key subtasks, including applicable legal sections, prison terms, fine terms, default terms, and prison_type, demonstrating its capability in handling complex legal decision-making tasks.

Robustness: The framework consistently achieves high performance across datasets of varying sizes and languages, highlighting its robustness and adaptability to diverse legal contexts.

The model's robustness and the dataset's real-time applicability for legal experiments are validated through the superior performance resulting from the evaluation of the IJMT dataset and the proposed sequential model compared to various datasets and baselines.

Contributions:

1. Conducted an extensive analysis on the thousands of judgment copies from various district courts solved in real time to derive a standardized Indian judgment outcome format for multi-task legal research, incorporating all essential components.
2. Developed an IJMT multi-task dataset, integrating case facts, evidence, and exceptions that influence judgments and facilitate multi-task decision-making recommendations.
3. Proposed a sequential multi-task decision-making framework that incorporates a lexical-vector dependency analysis engine. This framework models dual-level dependencies among the tasks by leveraging case facts, evidence, and exceptions for accurate judgment prediction.
4. Designed a context-aware threshold modulation strategy to address the prevalent data imbalance problem in legal judgment prognosis.
5. Evaluated the performance of the IJMT dataset and the proposed sequential model against various datasets and baselines to demonstrate the model's robustness and the dataset's real-time applicability for legal domain experiments.

The structure of this article is organized as follows: Section 2 reviews the related work, Section 3 details the creation of the IJMT multi-task dataset, Section 4 describes the proposed sequential dual-level dependency analysis framework, Section 5 presents the experimental results of the framework on the IJMT dataset, Section 6 provides an evaluative analysis of the dataset and the proposed framework by comparing with various datasets and baseline models, and Section 7 concludes the work with a discussion on future research directions.

2. Related work

Judicial decision prognosis is a complex and challenging task that has attracted significant attention over the years, with gradual advancements observed over time. The evaluation of deep learning across various fields has led to its adaptation into the legal domain. Consequently, JDP is converted into a text classification problem, with a primary focus on improving the prediction accuracy of the outcomes (Yao et al., 2020).

2.1. Judicial decision prognosis

Luo et al. (2017) proposed a unified framework for charge prediction and relevant article extraction using a two-stack attention-based neural network implemented through document and article encoders. Wang et al. (2018) addressed two aspects of crime classification named label dynamics and label imbalance using DPAM. This method improved the performance by leveraging multi-task learning and a dynamic threshold predictor. Wang et al. (2019) introduced HMN which combines hierarchical structures and label semantics into a unified framework. This approach transformed the classification task into a matching problem by decomposing article definitions into residual and alignment substructures, resulting in improved prediction outcomes. Xu et al. (2020) developed LADAN to resolve confusion in law articles during legal judgment prediction by utilizing a graph distillation operator to extract distinguishing features.

Chalkidis et al. (2019) introduced a new English LJP dataset using European Court of Human Rights (ECtHR) cases and created a hierarchical version of BERT that overcomes BERT's length limitation, yielding optimal performance. Chalkidis et al. (2020) extended BERT for the legal domain by systematically investigating its use, further pre-training BERT from scratch on specialized domains. They released LEGAL-BERT, LEGAL-BERT-SC, and LEGAL-BERT-SMALL, which outperformed larger models in efficiency and performance. Hao et al. (2021) designed DEAL for inductive link prediction between nodes with only attribute information by incorporating two node embedding encoders and an alignment mechanism. Paul et al. (2021) proposed LeSICiN, a heterogeneous graph model that explores both text and legal citation networks for legal statute identification. Yang et al. (2016) proposed a hierarchical attention network for document classification, featuring a structure that mirrors document hierarchy and dual attention mechanisms at the word and sentence levels to focus on essential content.

2.2. Encoders and transformers

Mikolov et al. (2013) proposed two novel architectures for generating continuous word vector representations from large datasets. These models achieved significant accuracy improvements in word similarity tasks with lower computational costs, learning high-quality word vectors from 1.6 billion words in under a day. Le and Mikolov (2014) introduced an unsupervised method known as the paragraph vector, which generates fixed-length feature representations for variable-length text inputs, such as paragraphs, documents, and sentences. The resulting dense document vector is trained to predict the words within the document. Pennington et al. (2014) combined global matrix factorization with local context window techniques to develop an unsupervised log-bilinear regression model by producing a meaningful substructure in the vector space. Kim (2014) demonstrated that CNNs with pre-trained word vectors perform well for sentence classification, with fine-tuning task-specific vectors boosting performance. They also proposed a modification to combine static and task-specific vectors for better results.

Devlin et al. (2019) introduced BERT, a pre-trained deep bidirectional representation model, by leveraging masked language modeling and next sentence prediction tasks. BERT was the first fine-tuning model requiring only a single added output layer. Yang et al. (2019a) developed XLNet by combining transformer-XL and an autoregressive model. XLNet addresses BERT's pretrain-finetune discrepancy using a segment recurrence mechanism and a relative encoding scheme.

State-of-the-art transformer models, such as InLegalBERT (Paul et al., 2023) and Longformer (Beltagy et al., 2020), addressed judicial judgment decisions effectively (Paul et al., 2024). Paul et al. (2023) explored pre-training in the Indian legal domain by continuing the pre-training of popular legal PLMs, LegalBERT and CaseLawBERT, on Indian legal data. They also trained a model from scratch using a vocabulary derived from Indian legal texts. Their findings revealed that this approach improved performance not only in the Indian legal domain but also in the original European and UK contexts. Transformers typically struggle with long sequences due to the quadratic scaling of self-attention. The Longformer (Beltagy et al., 2020) overcomes this limitation with a linear-scaling attention mechanism that combines local windowed attention with task-specific global attention, allowing efficient processing of lengthy documents.

Among various encoders and transformers, Doc2Vec and XLNet have demonstrated superior performance on unstructured textual data such as case facts (Madambakam et al., 2023) and are therefore adopted in this work for encoding.

2.3. Multi-task dependency learning

Unlike traditional methods that treat subtasks independently and ignore their interdependencies, the topjudge (Zhong et al., 2018) is a topological multi-task learning framework that represents these relationships using a Directed Acyclic Graph (DAG). This approach leveraged the subtasks and their DAG-based dependencies to enhance judgment prediction. Yang et al. (2019b) highlight that existing methods inefficiently utilize result dependencies among subtasks and struggle to predict accurately for cases with similar descriptions but different penalties due to ignored word collocation information. They proposed a multi-perspective Bi-feedback network with a word collocation attention mechanism by leveraging subtask dependencies through a forward prediction and backward verification framework while incorporating word collocation features to improve accuracy.

Yao et al. (2020) developed the GHEDAP method, a multi-tasking model designed to dynamically identify subtasks in fact descriptions and their dependencies while extracting deep semantic information. Later, Yao et al. (2021) extended this work with CSDNET, which enhances GHEDAP by capturing commonalities, specificities, and dependencies among subtasks using three modules: learning, denoising, and reinforcing. Chen et al. (2022) proposed a method for predicting article citations in law documents, addressing data imbalance via transfer learning with weight sharing and handling missing values through nonstatic word embeddings. It also used fact-label projection for few-shot learning. Lyu et al. (2022) identified two major challenges in LJP tasks: indistinguishable fact descriptions and misleading law articles with similar TF–IDF representations. To address these, they proposed CEEN, a reinforcement learning-based framework that extracts four key criminal elements: criminal, target, intentionality, and behavior. CEEN used an RL-based extractor for precise element identification and constructs distinct representations to improve law article predictions that enable accurate multi-task judgment predictions.

2.4. Existing datasets

Joshi et al. proposed IL-TUR (Joshi et al., 2024), Indian benchmark datasets comprising monolingual tasks (in English and Hindi) and multilingual tasks in nine Indian languages, focusing on domain-specific challenges that cover various aspects of the legal system, particularly in understanding and reasoning over Indian legal documents. Paul et al. (2021, 2024) compiled a large novel dataset named Indian Legal Statute Identification (ILSI) for the legal statute identification task featuring case facts from several major Indian courts and statutes from the Indian penal code where statutes are identified based on the provided facts. Malik et al. (2021) introduced the Indian Legal Documents Corpus (ILDC) dataset comprising 35,000 cases from the Indian Supreme Court, each annotated with the original court decisions. It assists the judges in predicting case outcomes with an explanation to accelerate the judicial process. Nigam et al. (2024) introduced Prediction with Explanation (PredEx), the largest expert-annotated dataset tailored for legal judgment prediction and explanation in the Indian context, featuring over 15,000 annotations. This innovative corpus significantly advances the training and assessment of AI models in legal analysis. Paul et al. (2020) showed that using less than 3% of documents with sentence-level crime labels in a multi-task learning setup significantly enhances Automatic Charge Identification (ACI) performance. They employed a unique weighting scheme to distinguish crime-indicative sentences, improving document representations, and created a dataset from real legal cases from the Supreme Court of India.

Chalkidis et al. (2022) introduced the Legal General Language Understanding Evaluation (LexGLUE) benchmark, a comprehensive collection of datasets in the English language listed as ECtHR (Chalkidis et al., 2019), SCOTUS (Fang et al., 2023), EUR-LEX (Aumiller et al., 2022), LEDGAR (Tuggener et al., 2020), UNFAIR-ToS (Lippi et al., 2019), and CaseHOLD (Zheng et al., 2021) designed to assess model performance across a diverse range of legal NLU tasks in a standardized manner. But it does not contain a multi-task dataset. Xiao et al. (2018) introduced the CAIL 2018 dataset, the first large-scale Chinese legal dataset for judgment prediction, containing over 2.6 million criminal cases from the Supreme People's Court of China. It includes detailed annotations such as law articles, charges, and prison terms. Despite using conventional text classification baselines, predicting judgment results, especially prison terms, remains a challenge for current models. Chalkidis et al. (2019) introduced the ECtHR-B dataset, a legal resource derived from the European Court of Human Rights (ECtHR) case corpus designed to predict court decisions based on case facts. This dataset contains over 10,000 cases, each with detailed case facts and corresponding violated legal sections, making it suitable for multi-label classification tasks. It is available in English and marked as an important benchmark for evaluating Natural Language Processing (NLP) and machine learning models in the legal domain, thereby contributing to the advancements in legal judgment prediction.

Based on the study, it was found that only legal statute identification datasets and frameworks focused on determining applicable statutes based on case facts are currently available for the Indian legal system. However, there is a lack of a multi-task dataset that encompasses all

Table 1
A sample judgment for an Indian murder case.

Facts and evidence	The brief facts of the case of the prosecution, as stated in column No. 7 of the charge sheet, are that Smt. Lourd Mary, aged 82 years, the mother of CW.9 Bernard Balarj, CW.11 Igneshiesh @ Vignesh, and CW.12 Uday Kumar, was running a chakana shop at No. 35/6, Sarahi Road, Agrahara Dasarahalli, Bangalore City. A-2 Srinivas is the brother-in-law of CW.11. He was assisting Smt. Lourd Mary in her shop...
Sections applied:	302, 120B, 201, 404
Section1: 302	Prison1: Life Prison1_imprison_type: Rigorous Fine1: Rs. 25,000 Default1: 5 years Default1_imprison_type: Simple
Section2: 120B	Prison2: Life Prison2_imprison_type: Rigorous Fine2: Rs. 25,000 Default2: 5 years Default2_imprison_type: Simple
Section3: 201	Prison3: 3 years, 6 months Prison3_imprison_type: Simple Fine3: Rs. 5,000 Default3: 2 years Default3_imprison_type: Simple
Section4: 404	Prison4: 2 years Prison4_imprison_type: Simple Fine4: Rs. 5,000 Default4: 6 months Default4_imprison_type: Simple
Prison_type	Concurrent
Other_judgment	If fine is deposited and if no appeal is preferred within the period of limitation, then compensation of Rs. 50,000/- will be given to the injured Jagpal out of the total fine.

essential legal decision components such as applicable law sections, prison terms, fine amounts, default terms, and prison_type following the Indian judgment outcome format. Additionally, there is a gap in multi-task models aimed at recommending these comprehensive decision components. It was observed that existing datasets, both domestic and international, do not adequately address the recommendation of these multi-task components. They fail to consider aspects such as proven evidence, applied exceptions, and section descriptions that define the scope of punishment terms (e.g., prison and fines) and their collective impact on the final judgment in multi-task scenarios, as seen in the dataset like CAIL.

3. Dataset creation

Based on the study, it is concluded that there is a lack of comprehensive multi-task datasets including all necessary judgment components according to the Indian judgment format. This gap is addressed by the Indian Judicial Multi-task (IJMT) dataset that offers recommendations for multi-task decision components by incorporating the gathered evidence, the facts of the case, and the consideration of exceptions influencing the judgment. These aspects are proposed by the advocate during discussions on the key elements to include in the facts of the murder case. These crucial aspects required for determining various judgment components are often overlooked in existing datasets, both in India and abroad. The IJMT dataset is the first of its kind in India to enable multi-task predictions of legal judgment components based on the Indian judicial format by accounting presented evidence and applied exceptions such as mental illness, insanity, or being the sole provider for the family, etc.

3.1. Preliminaries

This work focuses on criminal cases that can be addressed using relevant IPC sections, whereas systematically solving civil cases is challenging due to their heterogeneous nature (Madambakam and Rajmohan, 2022). The Indian judicial multi-task dataset focuses primarily on murder cases and the related sub-crime sections that contribute to the occurrence of the crime in the English language. The Indian Penal Code is essential for effective justice delivery in solving criminal cases by providing a standardized legal framework for defining offenses and prescribing punishments. Section 53 of the Indian Penal Code defines the types of punishments for crime as death, imprisonment for life, imprisonment, forfeiture of property, and fine. This work grouped death and imprisonment for life as classes under imprisonment. Forfeiture of property is grouped as a class under the fine task. Imprisonment is a common punishment in India involving the confinement of an offender to prison for a specific period to rehabilitate them. According to this, imprisonment is categorized into two types:

- *Rigorous imprisonment*: This involves confinement with hard labor such as agricultural work, carpentry, or drawing water. For example, life imprisonment and death are considered as rigorous by default.
- *Simple imprisonment*: In this form, the offender is confined without hard labor. It is imposed when a fine is insufficient and aims to isolate the offender from habitual criminals.

Another punishment fine is a monetary penalty used mainly for minor offenses such as traffic violations and revenue breaches. It is a common form of punishment across penal systems and is often imposed as a fine, compensation, or cost.[1]

Significance of incorporating IPC section descriptions in the dataset and model: The Indian penal code includes a comprehensive list of sections that outline various criminal offenses and their corresponding descriptions detailing the punishable acts along with the minimum and maximum terms of imprisonment and fines. For example,[2] section

[1] https://www.legalserviceindia.com/legal/article-8098-punishment-under-the-indian-code.html

[2] https://www.indiacode.nic.in/bitstream/123456789/15289/1/ipc_act.pdf

313 in IPC describes "Causing miscarriage without woman's consent—Whoever commits the offense defined in the last preceding section without the consent of the woman, whether the woman is quick with child or not, shall be punished with 2*[imprisonment for life], or with imprisonment of either description for a term which may extend to ten years, and shall also be liable to fine". After determining the applicable sections, the corresponding punishments are decided based on the IPC's section guidelines.

Indian judgment outcome format: Unlike the CAIL dataset, which includes single values for sections, charges, and penalties, the Indian judgment outcome consists of multiple values for sections, prison terms, fines, default terms, optionally including nature of imprisonment, which might be simple or rigorous for prison and default terms, and lastly prison_type.

Sections are initially determined based on the count of crime events proven by evidence, as described in the case facts. For each imposed section, the prison and fine terms are decided according to the IPC section guidelines based on the severity of the crime and exceptions requested in the case. If the offender fails to pay the fine, they are required to undergo an additional period of imprisonment known as the *Default* term as compensation. The type of imprisonment (simple or rigorous) is specified optionally for prison and default terms, indicating the nature of the imposed sentence. Lastly, the term *prison_type* specifies the total imprisonment period to be undergone by the criminal. It is applied only when multiple prison terms are imposed under different sections. It is implemented in either a consecutive or concurrent manner as follows:

Concurrent sentences: Multiple crimes are tried together, and a single sentence is served for all crimes, with sentences running simultaneously. For example, if an offender receives a 20-year sentence under Section 1 and a 10-year sentence under Section 2, they will serve both simultaneously and be released after 20 years.

Consecutive sentences: Sentences are served back-to-back, meaning the convict completes one sentence before starting the next. Using the same example, a 20-year and a 10-year sentence would result in a total of 30 years before release.

Concurrent sentences are applied for less serious offenses, whereas consecutive sentences are for more severe crimes. Judges determine the sentence, i.e, imprisonment and prison_type by evaluating factors such as the severity of the crimes, the defendant's criminal history, and potential threat to society.[3]

A sample judgment for an Indian case is illustrated in Table 1. It is worth noting that judgments for a case can vary among judges even when adhering to the same IPC guidelines, as they account for the specific severity of each crime. All such judgments are valid within the legal framework.

3.2. IJMT dataset

This work developed a new approach to create a multi-task dataset specifically for murder cases. This involved discussions with multiple lawyers and devising a methodology to extract murder cases and their corresponding judgment components from the Indian legal information database.

The IJMT dataset is in the English language and built by concentrating primarily on seven IPC sections related to murder as outlined in the IPC Act of 1860[2] as follows:

- Section 299 - Culpable homicide
- Section 300 - Murder
- Section 302 - Punishment for murder
- Section 304 - Punishment for culpable homicide not amounting to murder
- Section 304 A - Causing death by negligence
- Section 304B - Dowry death
- Section 307 - Attempt to murder

To compile the dataset, supporting crime sections that contribute to criminal acts such as kidnapping, harassment, and severe injury, etc., were also accumulated, resulting in a dataset encompassing 57 IPC sections finally. To collect relevant data for dataset creation, multiple interviews were conducted with multiple lawyers focusing on murder-related sections, essential details to outline the facts of the murder case, group conspiracy, case-solving procedure, evidence collection, judgment delivery processes, etc.

It was enlightened that the fine is determined based on the damage caused during the crime. The prison and fine terms are decided based on the type of court handling the case. For instance, magistrate courts oversee cases where the prison term is less than 5 years, while district courts can handle cases with prison terms of up to 7 years. Similarly, understand that murder cases are primarily adjudicated in district courts. Consequently, the case documents were collected from multiple district courts available in the Indian Kanoon legal information database[4] to create the multi-task dataset legal corpus indicated by the Indian Kanoon ID number in the IJMT dataset. It was observed that these documents contain judgment components in the last few paragraphs. Indian Kanoon includes cases from two district courts named Delhi and Bangalore, for multiple years. For the Delhi district, case records span from 2007 to 2024 years and a total of 459,470 documents are available. For the Bangalore district, case records span from 2015 to 2024 years and a total of 184,579 documents are available. As these case documents are in a lengthy and unstructured format, this work manually reviewed a total of 459,470 + 184,579 = 644,049 documents to extract the required judgment components to create the IJMT dataset. This entire dataset creation process took four months from the data-gathering inquiry process to completion. Since not all these documents met the criteria for the multi-task dataset creation, the following restrictions were applied to filter relevant murder case documents comprising the required judgment components:

- The document must pertain to a murder case.
- The accused must be convicted, excluding acquitted cases.
- The accused must be punishable under one of the seven murder sections (299, 300, 302, 304, 304A, 304B, 307) primarily, along with the optional supporting crime sections if proven.
- The case should not involve intentional group murder under section 34, specifying group criminal conspiracy punishable under section 120B. Although section 34 is applied, only one person is convicted while others are acquitted.
- The case must not be a reappeal requesting a review of a prior judgment.
- The case must not involve a bail request.

These restrictions were summarized in the query: "murder AND convicted AND (section 299 OR section 300 OR section 302 OR section 304 OR section 307 OR section 304 A OR section 304B) NOT section 34 NOT reappeal NOT bail NOT acquitted".

This query was used to extract murder case documents from Indian Kanoon's Delhi and Bangalore district court records for all available years. The focus was on single-person murders by individual offenders, excluding cases involving group intentional conspiracies under section 34. If a group of people is involved in a crime specified under section 34, then each person is prosecuted separately and will be punished under multiple crimes where their involvement is noticed and proved with evidence. This makes the judgment logic more complex. While section 34 cases involve complex judgments with multiple charges for each accused, the IJMT dataset primarily centers on single-offender murder

[3] https://sociallawstoday.com

[4] https://indiankanoon.org/

cases involving additional proven crime events. After filtering, 424 eligible murder documents were obtained out of 644,049 documents containing the required judgment components for building the dataset. This work has excluded the cases falling under the Motor Vehicle Act as they pertain to non-IPC sections.

The IJMT dataset was initially developed without including section descriptions. However, the prediction of prison, fine, and default terms for each section relies on the IPC description limits. To address this, a supplementary dataset named MT_Dataset_IPC_1860_Act_Des was created to provide the necessary section descriptions supporting the IJMT dataset for all 57 sections. The fields included in this supplementary dataset, along with their significance, are detailed in Table A.22 listed in Appendix. Subsequently, the fields from both datasets were merged into a single comprehensive dataset named IJMT_IPC_Des incorporating section descriptions for real-time adaptability as presented in Table A.23 in Appendix.

The *Id* field refers to the list of Indian Kanoon document IDs used to create the IJMT dataset legal corpus. The *District* field will have values of either Delhi or Bangalore, as these are the only two districts with data available in the Indian Kanoon legal database for obtaining murder cases. The *Year* field specifies the year in which the case was resolved. For certain years, no qualified documents were present meeting the defined restrictions for filtering murder cases from the Indian Kanoon legal database.

Dataset splitting: A *Split* column was introduced in the dataset to divide the final set of 424 filtered documents into training, validation, and testing sets in a 60:10:30 ratio, resulting in 249, 46, and 129 documents, respectively. Out of the total 57 sections, 7 correspond to murder-related sections, while the remaining 50 are supporting crime sections such as dacoity, kidnapping, riots, abduction, and others. To ensure an even distribution of the 57 section labels across the train, validation, and test sets, the following manual procedure was employed.

1. First, split the documents based on the list of 50 non-murder sections.
2. For each section N in this list, extract the set of records where section $S_i = N$, where $i = \{1, 2, 3, 4\}$ indicates up to four sections allowed per record.
3. These extracted records were then distributed among the training, validation, and testing sets in a 60:10:30 ratio.
4. Next, split the documents based on the list of 7 murder sections by following the same approach listed in steps 2 and 3.

In the raw IJMT dataset, very few cases contain section 5. Therefore, the dataset is restricted to a maximum of four different sections named *Section1, Section2, Section3*, and *Section4* based on the maximum values found. Section1 corresponds to one of the seven murder sections, while section2, section3, and section4 represent one of the fifty supporting crime sections. Each section is further extended with its corresponding IPC description by referring to the supplementary dataset MT_Dataset_IPC_1860_Act_Des, as detailed in Table A.22 in Appendix for all 57 sections. For each section imposed, associate the corresponding decision components, such as prison, prison_imprison_type, fine, default, and default_imprison_type.

Since the IJMT dataset considers a maximum of four sections, this results in four prison terms named *Prison1, Prison2, Prison3*, and *Prison4* with their respective implementation nature (either simple or rigorous) specified as *Prison1_imprison_type, Prison2_imprison_type, Prison3_imprison_type*, and *Prison4_imprison_type*. Similarly, the corresponding fine terms are listed as *Fine1, Fine2, Fine3*, and *Fine4*. The default terms are *Default1, Default2, Default3*, and *Default4* with their implementation nature (simple or rigorous) indicated by *Default1_imprison_type, Default2_imprison_type, Default3_imprison_type*, and *Default4_imprison_type*.

As multiple prison terms are involved, namely prison1, prison2, prison3, and prison4, the term *Prison_type* determines how these prison terms are to be implemented, either consecutively or concurrently. This impacts the total imprisonment duration the offender must undergo. In the *consecutive* type, the total imprisonment duration is the sum of all prison terms, i.e.,

$$\texttt{prison1} + \texttt{prison2} + \texttt{prison3} + \texttt{prison4}.$$

In contrast, the total imprisonment of *concurrent* type is calculated as the maximum value among the prison terms, i.e.,

$$\max(\texttt{prison1}, \texttt{prison2}, \texttt{prison3}, \texttt{prison4}).$$

The field *Total_prison* computes the total duration of imprisonment for the criminal based on the value of prison_type. Similarly, the field *total_fine* calculates the total fine amount imposed on the offender and computed as,

$$\texttt{Total\_fine} = \texttt{Fine1} + \texttt{Fine2} + \texttt{Fine3} + \texttt{Fine4}.$$

Additional components or directives included in the judgment were recorded in the *Other_judgment* field. For instance, "If the fine amount of Rs. 5000/- is deposited, the entire fine amount shall be paid to PW-1, Veeramani, father of the deceased, by way of compensation under section 357 of Cr.P.C". After processing all the above fields, the dataset was preprocessed through a renaming operation to standardize and make the labels unique. For example, all variations such as 304(a), 304 A, and 304-A are renamed to 304A.

Table 2 summarizes the statistics of the dataset.

Table 2
Statistics of the IJMT dataset.

Parameter	Value
No. of cases in the dataset	424
No. of train\validation\test set cases	249/46/129
Law articles (sections)	57
Murder\Non-murder sections	7/50
Prison terms	55
Fine terms	28
Default terms	53
Prison_types	2
Max. no. of labels per document	4
Avg. no. of labels per document	2.5

As per the law, imprisonment duration ranges from 24 h to life imprisonment. Accordingly, the various classes for prison terms are 1 to 29 days, 1 to 11 months, 1 to 13 years, life imprisonment, and death. This results in a total of 55 classes:

$$29 + 11 + 13 + 1 + 1 = 55.$$

The fine classes were determined by the distinct fine values present in the dataset, which were distributed among fine1, fine2, fine3, and fine4, along with an additional class of forfeiture of property. Since the default terms do not include life imprisonment or death, these two labels were excluded from the prison term classes, resulting in 53 classes used for default terms. The field prison_type contains two possible labels, listed as concurrent and consecutive.

High-quality legal datasets in English are scarce. The ECtHR dataset (Chalkidis et al., 2019) does not include the descriptions for the statutes, and another prominent dataset, CAIL (Xiao et al., 2018), is available only in Chinese. Additionally, the average number of statutes cited per document is notably low in these datasets, which is 0.71 for ECtHR and 1.09 for CAIL, indicating that most documents reference at most one statute or none in the case of ECtHR. This poses a challenge because the Legal Statute Identification (LSI) problem is fundamentally a multi-label classification problem, and these datasets do not adequately reflect this complexity (Paul et al., 2024). In contrast, the IJMT dataset captures the multi-label nature of the LSI problem more effectively with an average of 2.5 labels per document.

Courts are now publishing their judgments and orders online, which are publicly accessible in real time. Numerous copies of these court

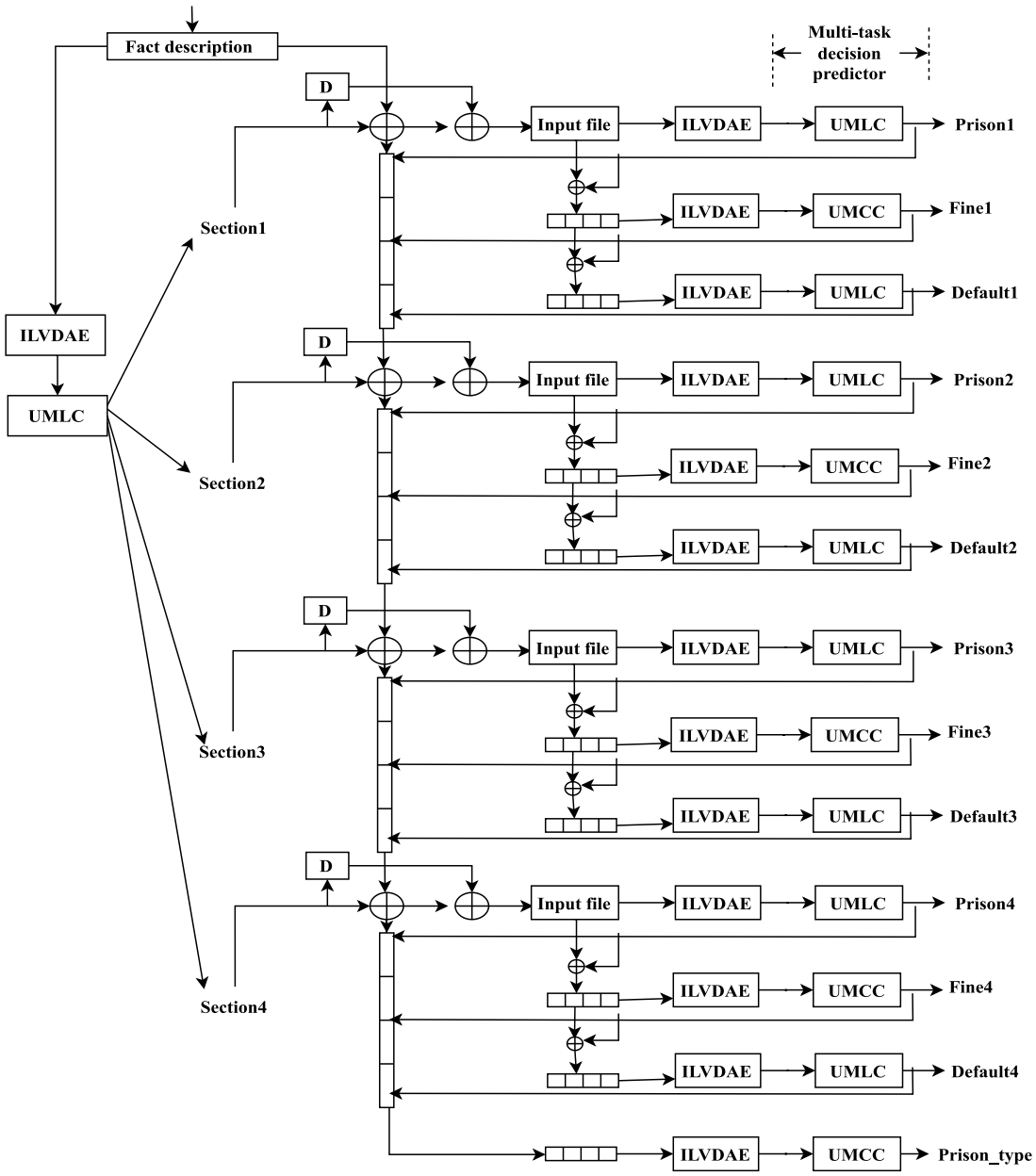


Fig. 1. Comprehensive dependency analysis framework for multi-task judicial decision system: A sequential approach.

decisions exist, and the Indian Kanoon legal information database hosts one such collection. As the IJMT dataset is derived from Indian Kanoon, which represents real-time solved court judgments, it is well-suited for real-world legal case analysis within the Indian context. The key advantage of this dataset is its inclusion of detailed facts and evidence, which encompass descriptions of the crime event, the evidence collected and validated, as well as any exceptions requested that influence the judgment process. Unlike existing domestic and international datasets, which lack this critical information, the IJMT dataset addresses these gaps. It facilitates the development of real-time adaptable frameworks that rely on comprehensive facts, evidence, and exceptions for accurate judgment delivery. One such framework described in the next section is the proposed sequential approach named the comprehensive dependency analysis framework for multi-task judicial decision systems. This framework is specifically designed to deliver judgments by incorporating all documented facts, validated evidence, and requested exceptions.

4. Proposed model

According to the study, no multi-task dataset or multi-task models currently exist that meet the requirements of the Indian judiciary system. Recognizing the necessity for multi-task models capable of predicting all essential judgment components, this work proposed a multi-task decision-making framework integrated with a Lexical-Vector Dependency Analysis Engine (ILVDAE) tailored specifically for the Indian judicial context. This approach mirrors the following methodology used by judges in real case analysis and verdict delivery, integrating an evidence-based and exception-aware decision-making process. The proposed IJMT dataset has been crafted to support the real-world methodology by including these critical details in the 'facts and evidence' and enabling accurate real-time judgment delivery.

- **Analysis of facts and evidence:** Initially, the judge reviews the facts and evidence of the case by examining the nature and severity of the crime events along with the sections under which

Table 3
Table listing the input file contents and the variant of the multi-task decision predictor for each predicted legal term.

S.No	Legal term	Input file	Prediction variant
1	Sections	{Facts and evidences}	UMLC
2	Prison1	{Facts and evidences, section1, sec1_desc}	UMLC
3	Fine1	{Facts and evidences, section1, sec1_desc, prison1}	UMCC
4	Default1	{Facts and evidences, section1, sec1_desc, prison1, fine1}	UMLC
5	Prison2	{Facts and evidences, section1, prison1, fine1, default1, section2, sec2_desc}	UMLC
6	Fine2	{Facts and evidences, section1, prison1, fine1, default1, section2, sec2_desc, prison2}	UMCC
7	Default2	{Facts and evidences, section1, prison1, fine1, default1, section2, sec2_desc, prison2, fine2}	UMLC
8	Prison3	{Facts and evidences, section1, prison1, fine1, default1, section2, prison2, fine2, default2, section3, sec3_desc}	UMLC
9	Fine3	{Facts and evidences, section1, prison1, fine1, default1, section2, prison2, fine2, default2, section3, sec3_desc, prison3}	UMCC
10	Default3	{Facts and evidences, section1, prison1, fine1, default1, section2, prison2, fine2, default2, section3, sec3_desc, prison3, fine3}	UMLC
11	Prison4	{Facts and evidences, section1, prison1, fine1, default1, section2, prison2, fine2, default2, section3, prison3, fine3, default3, section4, sec4_desc}	UMLC
12	Fine4	{Facts and evidences, section1, prison1, fine1, default1, section2, prison2, fine2, default2, section3, prison3, fine3, default3, section4, sec4_desc, prison4}	UMCC
13	Default4	{Facts and evidences, section1, prison1, fine1, default1, section2, prison2, fine2, default2, section3, prison3, fine3, default3, section4, sec4_desc, prison4, fine4}	UMLC
14	Prison_type	{Facts and evidences, section1, prison1, fine1, default1, section2, prison2, fine2, default2, section3, prison3, fine3, default3, section4, prison4, fine4, default4}	UMCC

these events are filed. Based on the validated evidence, the judge determines a list of applicable law sections corresponding to the offenses listed in the First Information Report (FIR).

- **Reference to IPC descriptions:** The judge refers to the Indian penal code descriptions to ascertain the permissible limits of prison and fine terms for each applicable offense.
- **Considering exceptional circumstances:** Lastly, the judge takes into account any exceptional circumstances pertinent to the case to decide the prison, fine, and default terms for each applicable section, as well as prison_type indicating the overall punishment to be penalized.

This work aims to predict the judgment by capturing the dependencies between various legal fields of inputs and predicted outputs in a sequential manner, which are derived from the above-outlined judicial methodology as follows:

Sections → Facts and evidence

Prison → Facts and evidence, section, sec_description

Fine → Facts and evidence, section, sec_description, prison

Default → Facts and evidence, section, sec_description, prison, fine

Prison_type → Facts and evidence, section, prison, fine, default

The proposed framework is designed based on the sequential process outlined in the above judgment methodology as depicted in Fig. 1. The inputs for each prediction task will vary depending on the outputs of the preceding tasks, combined with the fact description and the relevant section description, as described in the above dependencies and shown in Table 3. The function D(section$i$) retrieves the IPC description for the section$i$.

The proposed model aligns with the Indian judgment outcome format, which requires the prediction of multiple components, including applicable sections, multiple values of prison, fine, default terms, and prison_type. The model first identifies a list of applicable legal sections based on the number of crime events substantiated by the input facts and evidence. It considers up to four sections as constrained by the IJMT dataset, specifically labeled as section1, section2, section3, and section4. For each predicted section$i$, where $i = \{1, 2, 3, 4\}$, the model sequentially determines prison$i$, fine$i$, and default$i$ through a dynamically generated input file. This input file is created internally and includes facts and evidence, section$i$, section$i$ description, and dependent legal terms if applicable, such as section$j$, prison$j$, fine$j$, and default$j$, where $j = \{1, 2, \ldots, i-1\}$, as illustrated in Table 3. The file is then processed by the ILVDAE module, which analyzes the dependencies among the input components at both the lexical and vector levels to generate the enriched lexical features that are fed into the subsequent prediction module. The prediction module uses either a multi-label or multi-class classification approach, depending on the target component. Finally, the prison_type is determined using a similar process, but the input file includes only the facts and evidence and dependent legal fields such as section$i$, prison$i$, fine$i$, and default$i$ for $i = \{1, 2, 3, 4\}$ without including the section description.

The model predicts a total of 14 values as outcomes listed as applicable law sections, prison1, prison2, prison3, prison4, fine1, fine2, fine3, fine4, default1, default2, default3, default4, and prison_type. It is important to note that if multiple prison sentences are imposed, then prison_type is determined accordingly. The overall framework is constructed using 2 modules named Integrated Lexical-Vector Dependency Analysis Engine (ILVDAE) and a multi-task decision predictor module. The core functionality of the dual-level dependency analysis framework is handled by the ILVDAE module as depicted in Fig. 2.

4.1. Integrated lexical-vector dependency analysis engine (ILVDAE)

This module receives an internally generated input file as illustrated in Fig. 1 and processes the different types of data it contains, such as facts and evidence, IPC section description (sec_description), and dependent legal terms individually. It then analyzes the dependencies between these diverse input components related to the target term to be predicted at a dual level, i.e., lexical and vector, using a multi-phase approach depicted in Fig. 2. It generates the enriched lexical semantics as output, which is forwarded to the subsequent multi-task decision predictor module to accurately forecast the required legal term.

Due to the unstructured and complex nature of legal case documents, achieving high performance using standard RNN components like LSTM (Hochreiter and Schmidhuber, 1997), and GRU (Chung et al., 2014), popular transformer models such as BERT and XLNet, as well as other Pre-trained Language Models (PLMs) and Large Language Models (LLMs) tailored for text, is challenging. To address this issue, the ILVDAE module integrates various DL techniques designed to enhance the understanding of intricate legal text data for better outcomes. To prevent information loss often caused by deeper layers applied on raw input data, ILVDAE is built with only the essential layers necessary for its function. These layers are applied iteratively on the raw facts to extract rich features that aid in predicting multiple legal terms such as sections, prison terms, fines, default terms, and prison_type.

The ILVDAE module is composed of several distinct phases listed as encoder, lexical dependency analysis, vector fusing, and vector dependency analysis. Each phase is designed to work cohesively to address

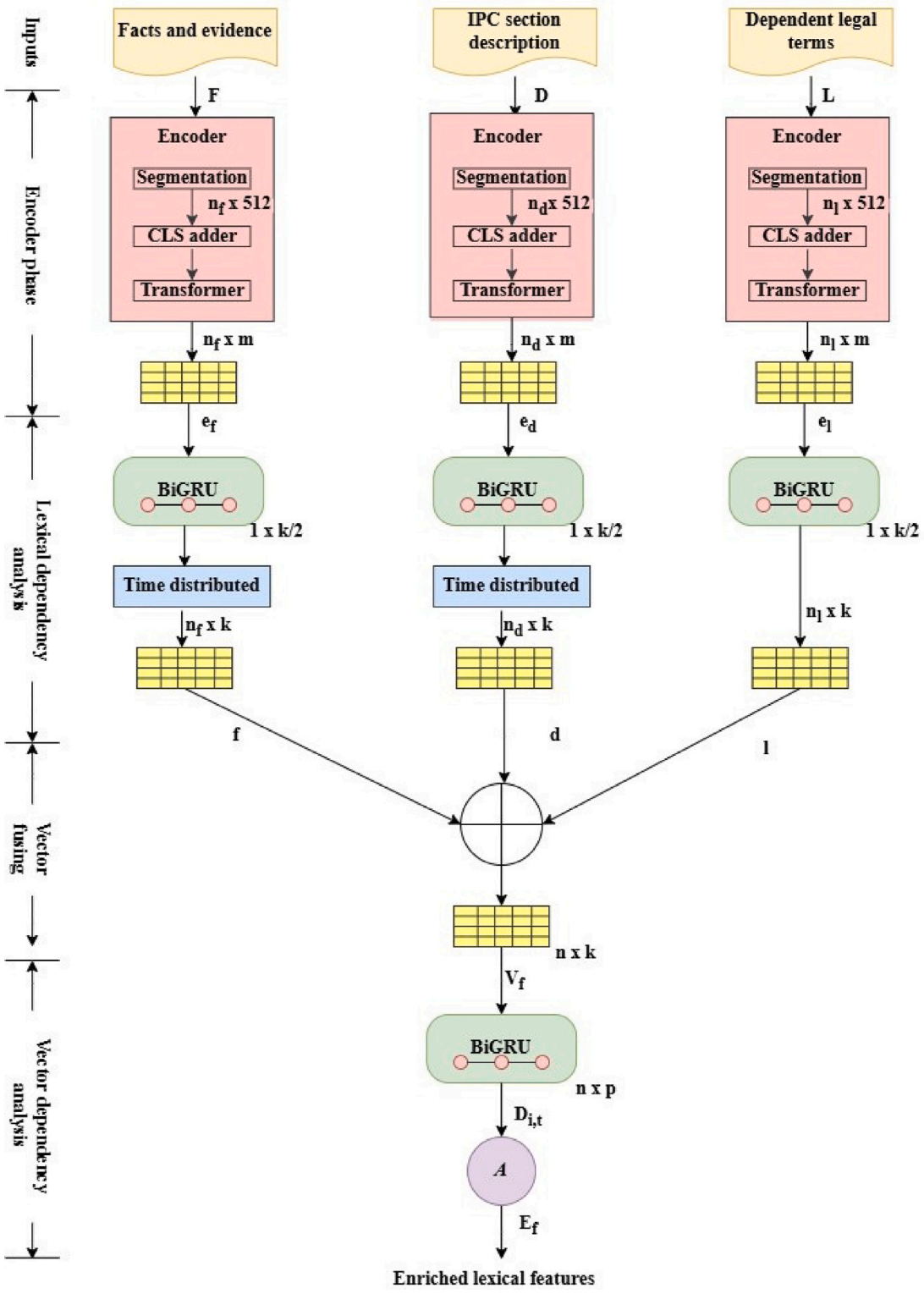


Fig. 2. Integrated Lexical-Vector Dependency Analysis Engine (ILVDAE) module.

the complexities of legal text data to enable robust and informed decision-making in the multi-task prediction process.

4.1.1. Encoder

This phase processes the input from the input file, which includes a combination of facts and evidence (F), IPC section description (D), and dependent legal terms (L), depending on the term being predicted. The encoder handles each type of input separately and generates embeddings e_f, e_d, and e_l. The input text is segmented according to the XLNet transformer's input limit of 512 tokens per partition. For instance, the facts and evidence input is divided into n_f segments, resulting in $n_f \times 512$ partitions, where n_f varies based on the length of

the text. A CLS token is appended to each segment, and these $n_f \times 512$ word partitions are transformed into numeric vectors using the XLNet transformer. The resulting output vectors have a size of $n_f \times m$, where m represents the encoded segment length.

Similarly, this phase is applied to the other two inputs: IPC section description (D) and dependent legal terms (L). The IPC section description is divided into n_d segments, resulting in $n_d \times 512$ partitions and producing an output vector of size $n_d \times m$. Likewise, the dependent legal terms are split into n_l segments, yielding $n_l \times 512$ partitions and generating an output vector of size $n_l \times m$.

4.1.2. Lexical dependency analysis

This phase takes the output vectors from the encoder and aggregates these embeddings using a Bidirectional GRU (BiGRU) (Kowsrihawat et al., 2018) in a time-distributed manner. Let the number of BiGRU units be $k/2$. First, consider the input facts and evidence for processing. The BiGRU processes each lexeme, i.e., each word embedding $wf_{i,t}$ in the 1D vector of size m, and captures the dependencies among the m lexemes to produce k aggregated embeddings as the outcome. The time-distributed layer applies BiGRU across all n_f vectors of size m and produces a condensed output of $n_f \times k$ aggregated vectors that capture vector-wise lexical dependencies denoted as f. In essence, the time-distributed layer is employed to identify long-term dependencies between lexemes within a vector.

This lexical dependency analysis phase is similarly applied to the other two inputs: the IPC section description and the dependent legal terms. It extracts the corresponding lexical dependencies between the text embedding vectors of sizes $n_d \times m$ and $n_l \times m$ that results in condensed vectors d of size $n_d \times k$ and l of size $n_l \times k$ respectively. Let the word embeddings for IPC section description and dependent legal terms be represented by $wd_{i,t}$ and $wl_{i,t}$. Note that for the dependent legal terms input, the time distribution layer is either optional or not required because the input length is $L \leq 512$ words, implying $n_l = 1$ segment. The following equations illustrate the process of drawing dependencies at the lexeme level i for each vector f_j using BiGRU and the time-distributed layer:

$$f_j = BiGRU(wf_{i,t}, h_{i,t-1}) \tag{1}$$

where $j = \{1, 2, 3, \ldots, n_f\}$ and $f = [f_1, f_2, \ldots, f_{n_f}]$

$$d_q = BiGRU(wd_{i,t}, h_{i,t-1}) \tag{2}$$

where $q = \{1, 2, 3, \ldots, n_d\}$ and $d = [d_1, d_2, \ldots, d_{n_d}]$

$$l_p = BiGRU(wl_{i,t}, h_{i,t-1}) \tag{3}$$

where $p = \{1, 2, 3, \ldots, n_1\}$ and $l = [l_1, l_2, \ldots, l_{n_l}]$

This process captures the long-term dependencies between lexemes and aggregates them into concise vector representations. Here, $h_{i,t-1}$ is the hidden state.

To be more specific, assume word embeddings received from the transformer encoder to calculate f_1 are $\{wf_1, wf_2, \ldots, wf_m\} \in \mathbb{R}^{n_f \times m}$ corresponding to the facts and evidence. Let $\overrightarrow{h_i^f}$ and $\overleftarrow{h_i^f}$ represent the forward and backward hidden layers of a word or lexical embedding wf_i, calculated by the following equations:

$$\overrightarrow{h_i^f} = \overrightarrow{GRU}(wf_i), \quad \overleftarrow{h_i^f} = \overleftarrow{GRU}(wf_i) \tag{4}$$

Final hidden layer vector representation of the lexical embedding wf_i is obtained by concatenating the forward and backward GRU hidden layer representations i.e $h_i^f = [\overrightarrow{h_i^f}; \overleftarrow{h_i^f}](h_i^f \in R^k)$ where k is twice to the count of hidden units in GRU. Similarly, lexical embeddings $\{wf_1, wf_2, \ldots, wf_m\}$ are given as input to the BiGRU gate, and the final representation of f_1 is returned by taking the last layer (F) hidden state of the GRU as follows (Yang et al., 2022):

$$\{h_1^F, h_2^F, \ldots, h_k^F\} = \text{BiGRU}\left(\{wf_1, wf_2, \ldots, wf_m\}\right) \tag{5}$$

Time distributed layer: The time distributed layer (Yang et al., 2022) applies a fully connected operation at each time step of the sequence $\{h_1^F, h_2^F, \ldots, h_k^F\} \in \mathbb{R}^{m \times k}$. Its input and output are both k vectors, where k indicates the output dimension, which is the number of units in the fully connected layer. Time distribution is applied to n_f segments of lexical embeddings, resulting in an output dimension of $n_f \times k$ lexically dependent vectors. Thus, time distributed enhances the ability to extract information from many-to-many vector sequences. To enrich this feature of the lexical encoder to extract long text information, the output generated by the BiGRU $\{h_1^F, h_2^F, \ldots, h_k^F\}$ is piped into the time distributed layer, yielding the following output vectors:

$$\{e_1^f, e_2^f, \ldots, e_k^f\} = TimeDistributed(\{h_1^F, h_2^F, \ldots, h_k^F\}) \tag{6}$$

4.1.3. Vector fusing

This phase receives input from the previous lexical dependency analysis phase, specifically the three condensed dependency vectors listed as f, d, and l. These vectors represent the dependencies at the lexeme level for facts and evidence, IPC section description, and dependent legal terms, respectively. The vectors are then horizontally fused or concatenated to form a unified vector V_f of size $n \times k$ where $n = n_f + n_d + n_l$. This fused vector V_f effectively captures the combined semantic information from all input sources to enable the comprehensive analysis in the subsequent stages of the model.

$$Vf = concat(f, d, l) \tag{7}$$

4.1.4. Vector dependency analysis

This phase takes the output vector V_f of size $n \times k$ from the previous vector fusing phase as input. It then applies BiGRU to capture long-term dependencies between the multiple vector data within V_f. BiGRU processes each vector $v_{i,t}$ within V_f and outputs $D_{i,t}$, which contains the dependencies between the vectors. To summarize, BiGRU operates at the whole vector level without a time-distributed layer.

$$D_{i,t} = BiGRU(Vf, h_{i,t-1}) \tag{8}$$

Where $h_{i,t-1}$ is the hidden state from the previous time step.

Following this, attention (A) (Vaswani et al., 2017) is applied to extract enriched lexical features that contain the essential semantics, which will influence the decisions made in the subsequent prediction module named multi-task decision predictor.

$$E_f = attention(D_{i,t}) \tag{9}$$

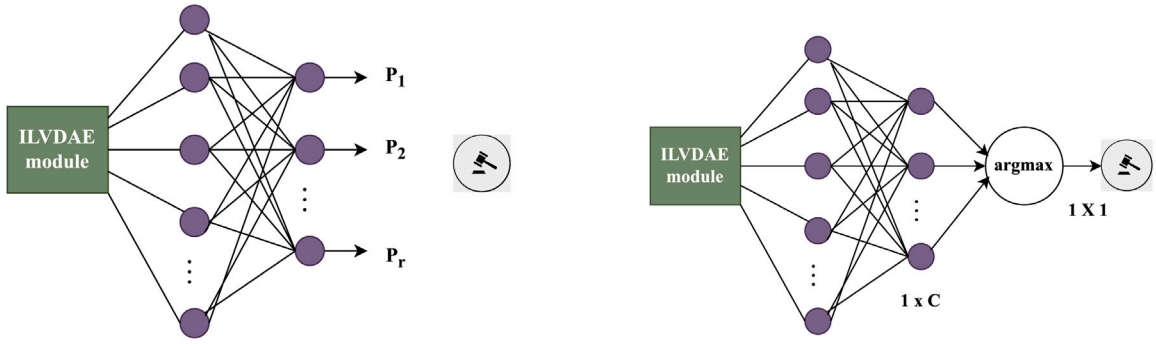
4.2. Multi-task decision predictor

The multi-task decision predictor module takes the enriched lexical features E_f extracted from the ILVDAE module as input and returns the predicted probabilities, ultimately leading to the final outcome prediction. Depending on the type of legal term to be predicted and the classification method applied, this prediction module has two variants to handle the diverse nature of legal term predictions.

4.2.1. Unified multi-label classifier (UMLC)

This variant utilizes multi-label classification to predict multiple labels that are applied simultaneously in the outcome, as shown in Fig. 3(a). It is specifically designed to predict applicable sections, prison terms, and default terms according to the Indian judgment format, where multiple labels for sections, defaults, and prison terms are predicted and applied simultaneously. A sample judgment could be,

- Section 302 (section), 7 years 6 months 15 days (prison), 3 months 10 days (default)
- Section 304 (section), 5 years (prison), 2 months (default)



(a) Unified multi-label classifier (UMLC). (b) Unified multi-class classifier (UMCC)

Fig. 3. Variants of the multi-task decision predictor.

UMLC predicts r labels as the outcome, where r indicates the label count and varies based on the type of legal term to be predicted.

$$r = \left\{ \begin{array}{ll} 58 & \text{for section,} \\ 56 & \text{for prison,} \\ 54 & \text{for default} \end{array} \right\}$$

including a null value, indicating not applicable or null.

UMLC uses softmax as an output function in the dense layer.

$$Y_r^p = softmax(W_{r,t} E_{tr} + b_{r,t}) \tag{10}$$

Here $W_{r,t}$ is the input weight; E_{tr} are the attention weights; $b_{r,t}$ is the bias; Y_r^p is the predicted distributions of the legal task r. Applied cross-entropy for the loss function given by,

$$L_r = -\lambda_r \sum_{i=1}^{|r|} Y_{r,i} log(Y_{r,i}^p) \tag{11}$$

where $|r|$ indicates the number of labels of the legal term r to be predicted, $Y_{r,i}$ is the ground-truth value of the label i, $Y_{r,i}^p$ is the predicted value of the label i, and λ_r is the weight factor of the legal task r.

4.2.2. Unified multi-class classifier (UMCC):

This variant employs multi-class classification to predict a single class out of the multiple eligible classes as the outcome, as shown in Fig. 3(b). It is used to predict terms like fine and prison_type, which require a single label to be returned in the outcome. A sample judgment could be,

- 5000rs (fine)
- Concurrent (prison_type)

UMCC predicts a single label by applying the $argmax(c)$ function on c outcome probabilities from the final dense layer, where c points to the label count and varies based on the type of legal term to be predicted.

$$c = \left\{ \begin{array}{ll} 29 & \text{for fine,} \\ 3 & \text{for prison\_type} \end{array} \right\}$$

including a null value, indicating not applicable or null.

UMCC uses sigmoid as an output function in the dense layer.

$$Y_c^p = sigmoid(W_{c,t} E_{tc} + U_{c,t} h_{c,t-1} + b_{c,t}) \tag{12}$$

$$l_c^p = argmax(Y_c^p) \tag{13}$$

Here $W_{c,t}$ is the input weight; E_{tc} is the attention weights; $U_{c,t}$ is the hidden weight; $h_{c,t-1}$ is the previous hidden state; $b_{c,t}$ is the bias; Y_c^p is the predicted distributions of the legal task c; l_c^p is the final label predicted for the legal task c. Applied cross-entropy for the loss function given by,

$$L_c = -\lambda_c \sum_{i=1}^{|c|} Y_{c,i} log(Y_{c,i}^p) \tag{14}$$

where $|c|$ indicates the number of labels of the predicted legal term c, $Y_{c,i}$ is the ground-truth value of the label i, $Y_{c,i}^p$ is the predicted value of the label i, and λ_c is the weight factor of the legal task c.

4.3. Context-aware threshold modulation strategy:

After recognizing the significance of thresholds and their impact on multi-label and multi-class classification, especially in the legal domain, this work derived a novel context-aware threshold modulation strategy. This approach dynamically fine-tunes thresholds by automatically learning robust and label-specific thresholds to address the label imbalance problem commonly referred to as the data imbalance issue (Chen et al., 2022). This imbalance problem is well known in the legal context and arises due to the presence of both frequent and infrequent sections. Frequent sections are generally referred by a larger number of cases and have a higher probability of being predicted accurately. In contrast, infrequent sections are referenced in fewer cases and exhibit a lower prediction probability. Frequent sections often have higher predicted values from the model, requiring higher thresholds for applicability, whereas infrequent sections with lower predicted logits require lower thresholds for accurate applicability determination. Therefore, modulating label-specific thresholds based on their frequency of occurrence is essential and significantly impacts performance evaluation compared to using a fixed threshold.

The prediction of sections, various prison terms, and default terms is treated as a multi-label classification problem where multiple applicable labels are predicted in the output. In this approach, context-aware label-wise thresholds are modulated by evaluating the model's performance for various threshold values ranging from 0 to 1 with increments of 0.03. The threshold value that yields the best F1 score for each label is selected and used to convert the model predictions into binary values. Performance metrics such as precision, recall, F1 score, and accuracy are computed based on these binary predictions. For performance comparison, the F1 score is primarily utilized with a focus on macro-level metrics. For multi-label classification, macro measures of precision, recall, F1 score, and accuracy are calculated through the aggregated values (Madambakam et al., 2023).

The prediction of fine values and prison_type is formulated as a multi-class classification problem where a single label is selected from the multiple predicted labels as the final output. Similar to the multi-label classification process, context-aware label-wise thresholds are determined to convert model predictions into binary values. The `argmax` function is then used to determine the final predicted label. Performance metrics, including precision, recall, F1 score, and accuracy, are evaluated by comparing the predictions with ground truth labels using predefined functions. In this context, accuracy is used as the primary metric for performance comparison, with a focus on macro-level metrics.

Table 4
Outcome of the comprehensive dependency analysis sequential framework for judging the multi-task components of the judicial decision system.

S.No	Task	Runs	Epochs	Precision	Recall	F1 score	Accuracy
1	Sections	2	10	71.274	72.675	**71.21**	97.126
2	Prison1	1	10	85.644	85.825	**85.367**	98.075
3	Prison2	1	10	85.979	87.482	**86.39**	97.715
4	Prison3	1	10	96.408	96.444	**96.412**	99.642
5	Prison4	1	10	99.1	99.107	**99.103**	99.986
6	Default1	1	10	87.156	87.682	**87.192**	97.186
7	Default2	1	10	87.259	87.962	**87.564**	98.593
8	Default3	1	10	95.248	95.37	**95.307**	99.755
9	Default4	1	10	98.133	98.148	**98.14**	99.971
10	Fine1	6	100	9.51	16.84	9.37	**16.28**
11	Fine2	5	100	19.83	12.47	13.54	**37.21**
12	Fine3	7	100	44.03	46.17	44.98	**93.02**
13	Fine4	4	100	49.61	50	49.81	**99.22**
14	Prison_type	9	50	53.15	54.26	53.28	**54.26**

Table 5
Hyperparameters varied for predicted multi-task components of proposed dependency analysis sequential framework.

S.No	Hyperparameter	Sections	Prison	Fine	Default	Prison_type
1	Output layer	sigmoid	softmax	sigmoid	softmax	sigmoid
2	Count of classes including null	58	56	29	54	3
3	Count of multiple runs	2	1	6/5/7/4	1	9
4	Classification type	Multi-label	Multi-label	Multi-class	Multi-label	Multi-class
5	Epochs	10	10	100	10	50
6	Metric	Binary accuracy	Binary accuracy	Accuracy	Binary accuracy	Accuracy

5. Experimental results

This section presents the experimental results of the proposed sequential multi-task judicial decision system framework as summarized in Table 4. The model predicted a total of 14 different components present in the Indian judgment outcome, including applicable sections, various prison terms, default terms, fine terms, and prison_type. The evaluation measures considered for all the experiments in this work are precision (P), recall (R), F1 Score (F1), and accuracy (A), all computed using macro-averaging weights. This approach ensured equal weighting for each class to make it well-suited for handling imbalanced datasets. The experimental results in Table 4 demonstrate that the model performed significantly better on multi-label classification problems, such as section, prison, and default term prediction (measured by F1 score), compared to multi-class problems like prison_type and fine prediction (measured by accuracy). Specifically, the model showed stronger performance for section, prison, and default term predictions than for fine and prison_type predictions. The `'nan'` class indicates the non-applicability of consecutive and concurrent techniques in cases where only a single section is relevant rather than multiple sections. The limited number of records in the dataset with the prison_type classified as either consecutive or concurrent has led to insufficient training data, which resulted in reduced model performance.

Performance improved progressively from prison1 to prison4, default1 to default4, and fine1 to fine4. This improvement is attributed to the inclusion of dependent legal terms as part of the training data alongside section descriptions and an increase in null values in later terms such as section4, prison4, fine4, and default4. Adding dependent legal terms enhances the data volume used for training. The proposed sequential model achieved the highest performance on the IJMT dataset compared to existing models on Indian datasets for section prediction with an F1 score of 71.21%. The IJMT multi-task dataset developed here is well-suited for real-time applications, as evidenced by the results in Table 4, which highlight its effectiveness for multi-task predictions in judgment delivery.

Hyperparameters:

The common hyperparameters utilized for predicting various multi-task components, including section, prison, fine, default, and prison_type using the proposed dependency analysis sequential framework are as follows: The kernel initializer is set to he_uniform, and the kernel regularizer is l2 (0.01). Various learning rates were applied: 1e−5 for the BiGRU layer at the lexical and the vector dependency analysis phases. At the vector dependency analysis phase, the normalization layer used a learning rate of 1e−4, and the sent_attention layer applied 3e−5. A momentum of 0.95 was applied to the above layers. The dropout rate is 0.5, and the threshold step increment is 0.03 in the context-aware threshold modulation strategy. The optimizer is Adam, the activation function is ReLU, and the loss function used is binary cross-entropy. The batch size for training is 64.

The varying hyperparameters for each outcome are listed in Table 5. The hyperparameters under the prison column represent the common settings applied to compute prison values, such as prison1, prison2, prison3, and prison4. The configurations under the default column outline the shared settings used to predict default1, default2, default3, and default4. The settings listed under the "fine" column indicate the common configurations employed to forecast fine1, fine2, fine3, and fine4. The softmax output function was applied for predicting various prison and default terms. When the sigmoid output function was applied, these terms resulted in imprisonment values exceeding the typical human lifespan. The count of multiple runs for the fine term is indicated by 6/5/7/4 means that the fine1 code is repeatedly run 6 times, the fine2 code 5 times, the fine3 code 7 times, and the fine4 code 4 times for performance improvement. Similarly, other values are also mentioned in the table.

5.1. Fine term evaluation

As per the study, Machine Learning (ML) methods tend to perform better than deep learning models when the dataset size is limited. Hence, regression techniques are initially applied to fine prediction. However, the performance of regression models is relatively poor when the proposed model is adapted as a regression problem for fine1 prediction. The error metrics, such as Root Mean Square Error (RMS), Relative Absolute Error (RAE), and R^2 are high compared to the traditional ML algorithms such as SVM with a polynomial kernel, ElasticNet, and SVR with the Radial Basis Function (RBF) kernel, which outperform other models in this context, as shown in Table 6. Among these, SVM with a polynomial kernel and SVM with an RBF kernel demonstrated the near-best performance in terms of RMS values. ElasticNet ranked next, followed by the proposed model, which has achieved the next

Table 6
Performance comparison of the proposed model as regression with the traditional ML regression techniques for fine1 prediction.

S.No	Model	RMS	R^2 Error	MAE
1	Proposed model	1118522588	−0.15461	12 271.9691
2	Linear regression model	2.66014×10^{13}	−27309.4077	4138251.882
3	SVM-polynomial kernel	1026920087	−0.0543	10 572.8972
4	Lasso	1.36832×10^{12}	−1403.7886	1181668.928
5	Ridge	3837332988	−2.9396	24 724.9732
6	ElasticNet	1088800633	−0.1178	20 030.9316
7	Random forest model	7240750975	−6.4337	79 504.4574
8	XGBoost regression model	2571965551	−1.6405	38 434.1217
9	Ensemble model for regression	981242434.9	−0.0074	15 389.3884
10	SVR with RBF kernel	1026898452	−0.0543	10 573.2346

Table 7
Performance comparison of various traditional machine learning regression algorithms and the proposed sequential model on the IJMT dataset for fine1 prediction.

S.No	Model	P	R	F1	A
1	Proposed model	9.51	16.84	9.37	**16.28**
2	Linear model	0	0	0	0
3	SVM-polynomial kernel	0.31	0.51	0.38	1.55
4	Lasso	0	0	0	0
5	Ridge	0	0	0	0
6	ElasticNet	0	0	0	0
7	Random forest	0	0	0	0
8	XGBoost	0	0	0	0
9	Ensemble model	0	0	0	0
10	SVR with RBF kernel	1.28	1.19	1.23	6.2

Table 8
Performance comparison of various traditional machine learning regression algorithms and the proposed sequential model on the IJMT dataset for fine2 prediction.

S.No	Model	P	R	F1	A
1	Proposed model	19.83	12.47	13.54	**37.21**
2	Linear model	0	0	0	0
3	SVM-polynomial kernel	3.81	3.16	3.45	**38.76**
4	Lasso	0	0	0	0
5	Ridge	0	0	0	0
6	ElasticNet	0	0	0	0
7	Random forest	0	0	0	0
8	XGBoost	0	0	0	0
9	Ensemble model	0	0	0	0
10	SVR with RBF kernel	1.37	3.45	1.96	12.4

Table 9
Performance comparison of various traditional machine learning regression algorithms and the proposed sequential model on the IJMT dataset for fine3 prediction.

S.No	Model	P	R	F1	A
1	Proposed model	44.03	46.17	44.98	**93.02**
2	Linear model	0	0	0	0
3	SVM-polynomial kernel	20	18.76	19.36	**93.8**
4	Lasso	0	0	0	0
5	Ridge	0	0	0	0
6	ElasticNet	0	0	0	0
7	Random forest	0	0	0	0
8	XGBoost	0	0	0	0
9	Ensemble model	0	0	0	0
10	SVR with RBF kernel	20	18.76	19.36	**93.8**

highest performance. The ensemble regression model used combines random forest and XGBoost as base models and linear regression as the meta-learner.

To improve the performance, a log transformation is applied to the y-values (y_train, y_test, y_val) using the $\log(y+1)$ function that can handle zero values, as $\log(0)$ is undefined. The predicted values were then inverse-transformed using the $\exp(x)-1$ function to normalize their distribution. The following hyperparameters were applied to the ML algorithms: a Doc2Vec encoder with a vector size of 512 for converting input text to numerical embeddings, an Adam optimizer, a batch size of 64, 50 epochs, a ReLU activation function, and a linear classification layer.

Table 10
Performance comparison of various traditional machine learning regression algorithms and the proposed sequential model on the IJMT dataset for fine4 prediction.

S.No	Model	P	R	F1	A
1	Proposed model	49.61	50.00	49.81	**99.22**
2	Linear model	0	0	0	0
3	SVM-polynomial kernel	50.00	49.61	49.81	**99.22**
4	Lasso	0	0	0	0
5	Ridge	0	0	0	0
6	ElasticNet	0	0	0	0
7	Random forest	0.04	1.32	0.08	3.10
8	XGBoost	0	0	0	0
9	Ensemble model	0	0	0	0
10	SVR with RBF	50.00	49.61	49.81	**99.22**

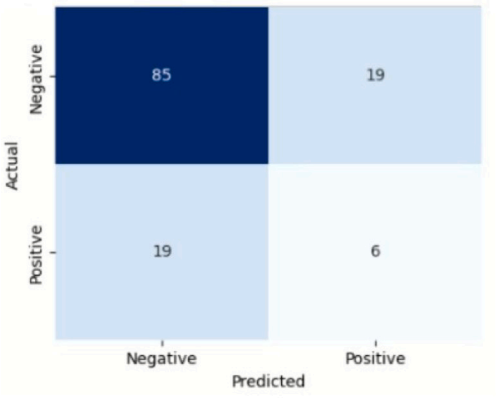
Deep learning models with multiple layers are generally capable of capturing complex non-linear relationships between the features and target variables, are scalable, and tend to perform well with larger datasets. However, given the small IJMT dataset having a size of 424 records, neither the proposed model as a regression technique nor ML regression algorithms were able to perform effectively. To address this limitation, the fine-terms prediction problem was reformulated from regression to a multi-class classification problem and solved using the proposed sequential dependency analysis model.

The performance metrics, such as precision, recall, F1 score, and accuracy of the proposed model are compared with those of the ML algorithms for fine1, fine2, fine3, and fine4 predictions as shown in Tables 7, 8, 9, and 10 respectively.

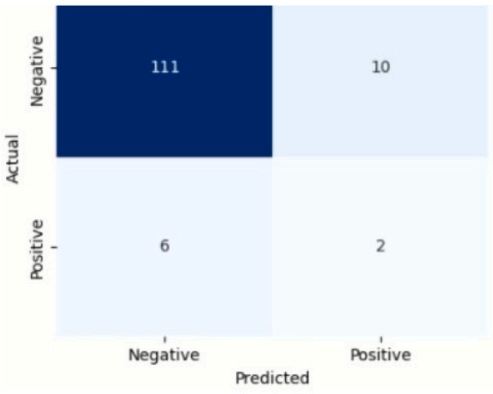
The proposed model has achieved the highest performance compared to traditional ML algorithms to predict fine1 with an accuracy of 16.28%, as shown in Table 7. SVR with polynomial kernel demonstrated the highest performance with 38.76% accuracy for predicting fine2, followed closely by the proposed model with an accuracy of 37.21%, as shown in Table 8. In fine3 prediction, both the SVM with polynomial kernel and SVR with Radial Basis Function (RBF) exhibited the highest and equal performance with 93.8% accuracy, while the proposed model showed a slightly lower but still impressive performance of 93.02%, as shown in Table 9. The proposed model, SVM with polynomial kernel, and SVR with RBF have achieved equal and the highest performance with 99.22% accuracy in predicting fine4, as shown in Table 10. The dataset contains the highest fine amounts in fine1, fine2, and fine3, while fine4 has a substantial number of null values.

These comparisons demonstrate that SVM with a polynomial kernel and SVR with the RBF kernel are the closest competitors to the proposed model, delivering similar performance. The hyperparameters used in the proposed model for fine-term prediction include the Adam optimizer, 100 epochs, binary cross-entropy as a loss function, a batch size of 64, the ReLU activation function, and a sigmoid output layer. Multiple runs were conducted to achieve the optimal performance.

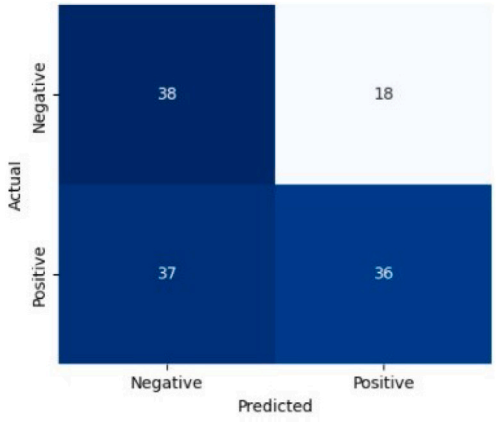
Several methods were employed to improve fine-term prediction: (a) Adopted the proposed deep learning model as a regression task. (b) Incorporated log transformation of the independent variable in the proposed DL regression model. (c) Configured the proposed DL



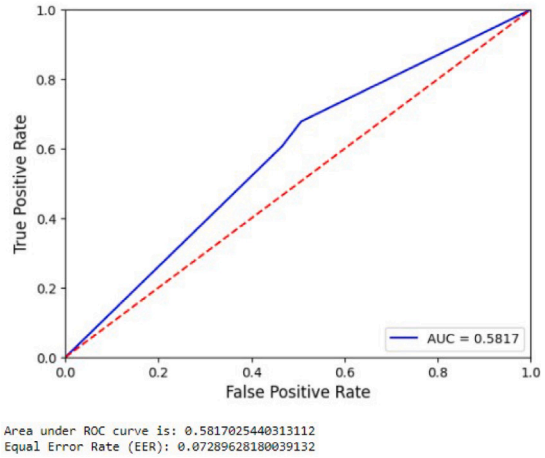
(a) Heatmap for fine1 classification: 5000rs fine class.



(b) Heatmap for fine2 classification: 500rs fine class.



(c) Heatmap for prison_type classification: concurrent Class.



(d) AUC-ROC curve for prison_type classification.

Fig. 4. Performance of the proposed model on the IJMT dataset for various predicted outcomes.

model for multi-class classification with multiple iterations conducted for fine1 to fine4 predictions. (d) Experimented with traditional ML regression techniques using XLNet embeddings, but errors arose in handling the input data format. (e) Tuned hyperparameters to optimize performance. (f) Implemented traditional ML regression methods with Doc2Vec embeddings. The outcomes of these efforts are summarized in Tables 6 to 10. Fine prediction typically noted less performance compared to the other multi-task components involved in this work. It is worth noting that most existing national and international works have not placed sufficient focus on prison and fine term prediction, and the results achieved are comparable.

The regression algorithms yielded lower performance, which can be attributed to the unstructured nature of the input text data. Regression algorithms are traditionally designed for structured numerical data, but when working with unstructured text inputs, numerical representations must first be derived. In this experiment, Doc2Vec embeddings were used for traditional regression algorithms, and XLNet was employed in the proposed model to transform unstructured text into meaningful numerical representations. Generally, fine performance is lower in existing works too.

This approach was validated as appropriate for comparing the proposed model with traditional regression techniques, as demonstrated in Tables 6 to 10 for fine prediction. Due to the inherent challenges in extracting relationships from unstructured text data for regression tasks, the performance remained suboptimal. To address this, the fine prediction task was reformulated as a multi-class classification problem. This change significantly improved the performance metrics, including macro precision, recall, F1 score, and accuracy, as shown in Table 7 through Table 10.

As the amount of training data is increased with fine1 values, the performance for predicting fine2 is improved. The training data includes fine1 and fine2 values along with their corresponding fields as dependent legal terms for fine3 prediction, resulting in a significant improvement in accuracy. Likewise, dependent legal terms incorporate fine1, fine2, fine3, and related values for fine4 prediction, which resulted in a substantial enhancement in accuracy. It is concluded that the overall performance is improved as the training data size increases with dependent legal terms. Fig. 4 illustrates the performance of selected legal term predictions presented through heatmaps and the AUC-ROC curve.

6. Performance evaluation

This section evaluates the performance of the IJMT dataset and the proposed sequential model against various datasets and baselines to validate the model's robustness and the dataset's real-time applicability for legal domain experiments. This is achieved by assessing the two key innovations of the framework: the model and the dataset.

6.1. Model evaluation and generalization

This subsection evaluates the generalizability of the model by analyzing its performance on datasets from various national and

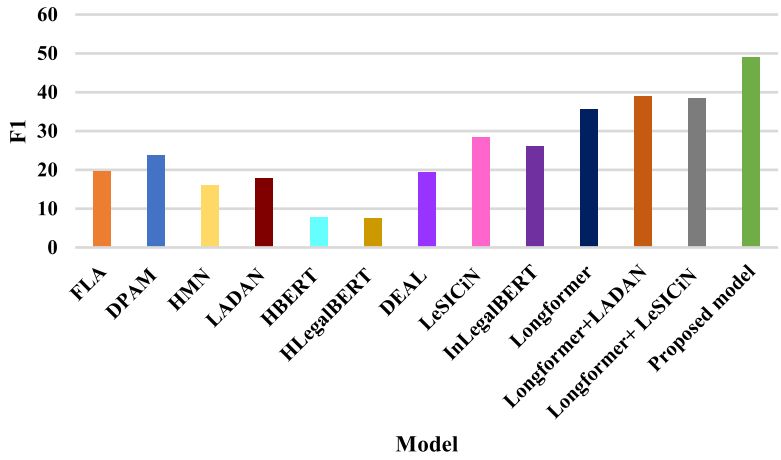


Fig. 5. Evaluation of the proposed sequential model on the ILSI dataset.

Table 11
Statistics of the datasets used in the evaluation work.

S.No	Dataset	Language	Training/Val/Test set	Labels	Territory	Total cases
1	ILSI	English	42,750/10,181/13,019	100	India	65,950
2	ECtHR-B	English	8,866/973/986	10	Europe	10,825
3	CAIL-2018 Small	Chinese	150,000/16,000/30,000	183	China	196,000

Table 12
Performance evaluation of the proposed model with existing models on the ILSI dataset.

S.No	Model	P	R	F1
1	FLA (Luo et al., 2017)	12.19	69.25	19.6
2	DPAM (Wang et al., 2018)	27.11	27.43	23.86
3	HMN (Wang et al., 2019)	10.27	57.3	16.12
4	LADAN (Xu et al., 2020)	12.54	46.17	17.93
5	HBERT (Chalkidis et al., 2019)	5.79	51.43	7.91
6	HLegalBERT (Chalkidis et al., 2020)	4.36	52.55	7.58
7	DEAL (Hao et al., 2021)	12.66	64.83	19.43
8	LeSICiN (Paul et al., 2021)	27.9	31.32	28.45
9	InLegalBERT (Paul et al., 2023)	58.75	19.29	26.22
10	Longformer (Beltagy et al., 2020)	53.98	28.73	35.73
11	Longformer+LADAN (Paul et al., 2024)	43.89	38.07	38.92
12	Longformer + LeSICiN (Paul et al., 2024)	45.34	35.3	38.38
13	Proposed sequential model	48.33	50	**48.97**

Table 13
Comparison of the proposed sequential model with LLMs on the ILSI dataset. Presented macro results.

S.N0	Model	ILSI		
		P	R	F1
1	GPT-3.5 0-shot (Joshi et al., 2024)	21.6	32.55	21.55
2	GPT-3.5 1-shot (Joshi et al., 2024)	27.06	22.07	22.61
3	GPT-3.5 2-shot (Joshi et al., 2024)	25.35	21.53	21.4
4	GPT-4 0-shot (Joshi et al., 2024)	25.31	26.74	23.99
5	GPT-4 1-shot (Joshi et al., 2024)	27.13	23.22	22.26
6	GPT-4 2-shot (Joshi et al., 2024)	25.16	20.89	20.53
7	LLaMa2 0-shot (Paul et al., 2024)	17.5	14.8	13.2
8	Proposed model	**48.33**	**50**	**48.97**

Table 14
Comparison of the proposed model with high-performing models on the ECtHR-B dataset.

S.No	Model	Macro F1
1	InLegalBERT (Paul et al., 2024)	75.88
2	InLegalBERT+LeSICiN (Paul et al., 2024)	74.46
3	InLegalBERT+LADAN (Paul et al., 2024)	78.05
4	LLaMa2 (Paul et al., 2024)	35.58
5	Proposed model	57.48

international legal systems. The statistics of the datasets used in this section for model evaluation is shown in Table 11.

6.1.1. Indian legal statute identification (ILSI) dataset

Legal research in India has primarily focused on the statute identification task due to the lack of a multi-task dataset that incorporates diverse decision components. The Indian legal statute identification dataset (Paul et al., 2021) is the most commonly used dataset containing facts and their corresponding applicable statutes in the English language. It shares similarities with the IJMT dataset in terms of unstructured format, but lacks evidence and exceptions. The evaluation of the proposed sequential framework against other models is demonstrated in Table 12 and Fig. 5 using macro-weighted performance metrics. The results showed that the proposed model surpasses the baseline models by exhibiting state-of-the-art performance on the ILSI dataset. Table 13 presents a comparison between the proposed model and various Large Language Models (LLMs) on the ILSI dataset. The results demonstrate that the proposed model outperforms all the LLMs.

6.1.2. ECtHR-B dataset

The ECtHR-B (Chalkidis et al., 2019) dataset is in the English language and derived from the European Court of Human Rights (ECtHR) case corpus designed to predict court decisions based on the case facts. The dataset provides detailed facts and violated articles, limiting

Table 15
Performance comparison of the proposed model with existing models on the CAIL-2018 small dataset.

S.No	Model	Section			Prison		
		P	R	F1	P	R	F1
1	FLA (Luo et al., 2017)	75.32	74.36	72.93	30.94	28.4	28.00
2	CNN (Kim, 2014)	76.02	74.87	73.79	33.07	29.26	29.86
3	HARNN (Yang et al., 2016)	75.26	76.79	74.90	34.66	31.26	31.40
4	Few-Shot (Chen et al., 2022)	77.80	77.59	76.09	35.07	26.88	27.14
5	Topjudge (Zhong et al., 2018)	79.77	73.67	73.60	34.73	32.73	29.43
6	MPBFN (Yang et al., 2019b)	76.30	76.02	74.78	31.94	28.60	29.85
7	LADAN (Xu et al., 2020)	78.24	77.38	76.47	36.16	32.65	32.49
8	BERT (Devlin et al., 2019)	81.85	82.61	81.13	39.64	38.74	38.47
9	CEEN (Lyu et al., 2022)	81.38	81.82	80.41	37.11	35.00	35.07
10	CEENBERT (Lyu et al., 2022)	83.21	83.73	82.32	41.15	39.72	40.03
11	Proposed model	**91.39**	**91.76**	**91.57**	**65.34**	**65.45**	**65.33**

Table 16
Performance comparison of the proposed model with baselines to predict section and prison_type predictions using the IJMT dataset.

S.No	Model	Sections				Prison_type			
		P	R	F1	A	P	R	F1	A
1	Longformer	62.524	64.319	62.173	95.429	49.68	53.49	50.47	53.49
2	Topjudge	68.959	70.611	69.184	97.099	27.79	52.71	36.39	52.71
3	Proposed model	**71.274**	**72.675**	**71.21**	**97.126**	**53.15**	**54.26**	**53.28**	**54.26**

the output to the prediction of applicable law sections. It serves as a valuable benchmark for assessing the performance of various natural language processing and machine learning models in the legal domain.

The proposed model surpasses the performance of the large language model LLaMa2, as illustrated in Table 14. Models leveraging InLegalBERT as the embedding method outperformed other models, as InLegalBERT is pre-trained on various countries' legal corpora, including India, enhancing its effectiveness in legal applications. In contrast, the XLNet embedding method used in the proposed model was not trained on any country's legal system.

6.1.3. CAIL dataset

CAIL (Xiao et al., 2018) is the only dataset available for multi-task prediction, but it is in the Chinese language. This work utilized the CAIL-2018 small dataset for experiments, which is a structured dataset comprising judgment components such as law article, charge, and prison term, i.e., penalty duration, but lacks a fine term. These components are determined by analyzing the case facts. Unlike the IJMT unstructured dataset, which contains hundreds of statements as facts and evidence, CAIL does not account for evidence and exceptions in delivering judgments and contains fewer sentences as facts.

Table 15 presents a comparison of the proposed model's performance with existing models extracted from Lyu et al. (2022) designed to predict multi-task components in the CAIL dataset. Since the proposed model focuses on predicting applicable sections and prison terms rather than charges, the evaluation was subsequently limited to these two components. Since the CAIL dataset includes only a single value for section, charge, and prison term, the prediction was limited to prison 1. The results demonstrate that the proposed model significantly outperformed the existing models in predicting the applicable section and prison terms.

6.2. Evaluation of the IJMT dataset

This section assesses the IJMT dataset by comparing the performance of the existing models with the proposed sequential model applied to the IJMT dataset. The baseline models used for comparison are `Topjudge` (Zhong et al., 2018) and `Longformer` (Beltagy et al., 2020). The reason for considering the topjudge model is that it employs a similar strategy to the proposed model by considering task dependencies for judgment prediction. The hyperparameters applied include a Word2Vec encoder with a vector size of 200 × 512 words, a learning rate of 1e−3, the Adam optimizer, 64 filters, a fully connected layer with 256 units, 10 epochs, a batch size of 128, a hidden layer with 256 neurons, and a sigmoid output layer. These settings align with the original model's hyperparameters.

The longformer is a state-of-the-art transformer model that incorporates task dependencies by utilizing inputs enriched with dependency information. The reason for considering the longformer model is that it efficiently captures long-range dependencies within the data and supports sequences of up to 4096 words. This work restricted the chunk size to a 512 word sequence to minimize unnecessary padding and reduce memory consumption. The longformer was implemented with an additional dense layer for representation.

A context-aware threshold modulation strategy was applied to all the comparative models with a threshold of $\geq$ 0.5 applied for all multi-label and multi-class problems, incorporating varying step threshold counts. The multi-class classification of various fine terms and prison_type was evaluated using accuracy due to the one-to-one mapping between predicted and true labels. In contrast, multi-label classification requires label-wise mapping of each pair, assessed using precision, recall, and F1-score.

Sections comparison:

Table 16 presents the performance of various models on applicable sections prediction using the IJMT dataset and highlights the superior performance of the proposed model. The prediction is carried out using a multi-label classification method enhanced by a context-aware threshold modulation strategy.

Comparison of prison terms:

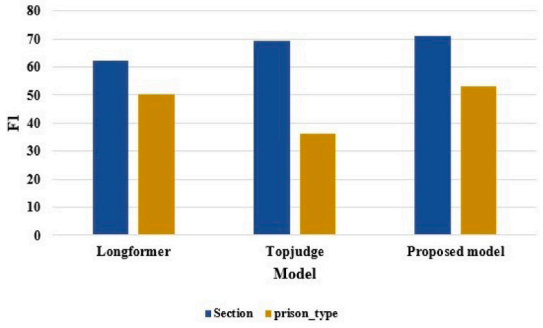
Table 17 presents the performance of various models to predict prison1, prison2, prison3, and prison4 terms using the IJMT dataset. Topjudge demonstrated performance close to the proposed model. Its performance is identical to the proposed model for prison4 and slightly outperforms the proposed model for prison3. Overall, the comparison models exhibited comparable performance to the proposed model in predicting prison terms. The prediction is performed using a multi-label classification approach with a context-aware threshold modulation strategy.

Comparison of default terms:

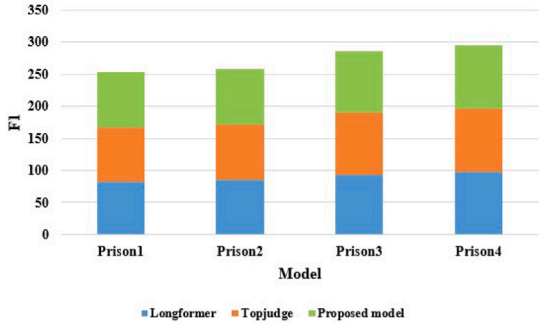
Table 17 presents the performance of various models to predict default1, default2, default3, and default4 terms using the IJMT dataset. Topjudge demonstrated performance close to the proposed model. Its performance is identical to the proposed model for default4 and slightly outperforms the proposed model for default2. Overall, the comparison models exhibited comparable performance to the proposed

Table 17
Performance comparison of the proposed model with baselines to predict various tasks of prison and default terms using the IJMT dataset.

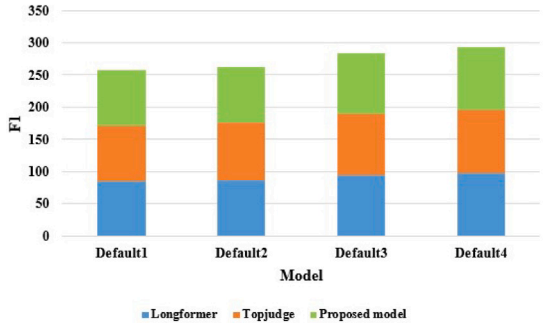
Model	Prison1				Prison2				Prison3				Prison4			
	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A
Longformer	82.066	83.008	82.473	98.006	85.762	86.6	86.108	98.297	92.781	92.822	92.80	99.778	97.307	97.307	97.307	99.944
Topjudge	84.763	85.566	84.97	97.328	85.762	86.607	86.112	**98.311**	**97.158**	**97.186**	**97.172**	**99.916**	**99.1**	**99.107**	**99.103**	**99.986**
Proposed model	**85.644**	**85.825**	**85.367**	**98.075**	**85.979**	**87.482**	**86.39**	97.715	96.408	96.444	96.412	99.642	**99.1**	**99.107**	**99.103**	**99.986**
Model	**Default1**				**Default2**				**Default3**				**Default4**			
	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A
Longformer	84.251	85.162	84.665	98.09	86.333	87.022	86.633	98.564	94.322	94.186	94.231	99.239	97.207	97.2	97.204	99.928
Topjudge	86.103	87.037	86.528	**98.133**	**88.421**	**88.807**	**88.603**	**99.196**	94.444	94.444	95.00	99.629	**98.133**	**98.148**	**98.14**	**99.971**
Proposed model	**87.156**	**87.682**	**87.192**	97.186	87.259	87.962	87.564	98.593	**95.248**	**95.37**	**95.307**	**99.755**	**98.133**	**98.148**	**98.14**	**99.971**



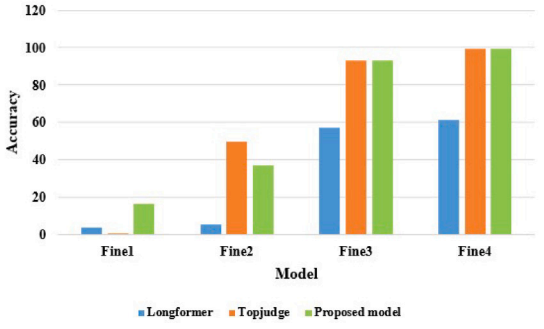
(a) Assessment of the sections and prison_type terms using the IJMT dataset



(b) Assessment of various prison terms using the IJMT dataset



(c) Assessment of various default terms using the IJMT dataset



(d) Assessment of various fine terms using the IJMT dataset

Fig. 6. Performance evaluation of the proposed model with baselines on the IJMT dataset.

Table 18
Performance comparison of the proposed model with baselines to predict various fine terms using the IJMT dataset. Used accuracy as the evaluation metric.

Model	Fine1	Fine2	Fine3	Fine4
Longformer	3.88	5.43	57.36	61.24
Topjudge	0.78	**49.61**	93.00	**99.22**
Proposed model	**16.28**	37.21	**93.02**	**99.22**

Table 19
Runtime comparison of the proposed model with the baselines on the IJMT dataset. Table represents the values in milliseconds.

Model	Sections	Prison1	Fine1	Default1	Prison_type
Longformer	470	491	488	502	504
Topjudge	4246	4258	4393	4371	4258
Proposed model	2205	2999	2029	2845	2175

model in predicting default terms. The prediction is achieved through a multi-label classification approach utilizing a context-aware threshold modulation strategy.

Comparison of fine terms:

Table 18 showcases the performance of various models to predict fine1, fine2, fine3, and fine4 terms using the IJMT dataset. Topjudge demonstrated the equivalent performance to the proposed model for fine4 and outperformed the proposed model for fine2 in terms of accuracy. The predictions were achieved through a multi-class classification approach employing a context-aware threshold modulation strategy followed by the argmax function. The performance of fine1 and fine2 is lower due to the wide variation of fine values, ranging from 0 to lakhs, and their dependence on multiple factors such as property loss and

Table 20
Statistical comparison of the proposed model with the baselines on the IJMT dataset for sections prediction. Considered macro F1 values for comparison.

S.No	Model	Corrected paired two-tailed t-test		Wilcoxon signed ranks test		Paired t-test	
		t_stat	P_value	w_stat	P_value	t_stat	P_value
1	Longformer	14.5878	0.0000	0.0	0.03125	15.9801	0.00001747
2	Topjudge	10.9995	0.0001	0.0	0.03125	12.0493	0.00006950

incurred damage. This will be addressed in future work with enhanced logic.

Performance of the term prison_type:

Table 16 showcases the performance of various models on prison_type prediction using the IJMT dataset and highlights the superior performance of the proposed model. The prediction is achieved through a multi-class classification approach utilizing a context-aware threshold modulation strategy followed by the argmax function.

Fig. 6 presents a graphical representation of the competing models' performance on the IJMT dataset. It demonstrated that the proposed sequential model outperformed the baseline models, which are the strong contenders in this work. Compared to all the models evaluated using both the existing datasets and the IJMT dataset in this section, the proposed model exhibited superior performance with the new IJMT dataset. This highlights the robustness of the proposed sequential dependency analysis model and the real-time adaptability of the IJMT dataset for legal experiments.

6.2.1. Runtime analysis

Table 19 presents the runtime comparison of the proposed model against the baseline models on the IJMT dataset. To improve performance, all models were trained over multiple runs and varying epochs. For consistency, the table reports the time taken (in milliseconds) for a single run over 10 epochs. Among all the models, the longformer transformer has recorded the lowest runtime, primarily because it does not have any additional layers on top of it for data processing other than the dense layer for prediction. In contrast, the proposed model integrates multiple additional layers on top of the XLNet transformer for data processing at both lexical and vector levels to extract enriched semantic features, along with the layers of the prediction module. The topjudge model also includes additional layers on top of the Word2Vec encoder for data processing and prediction. However, when compared to the topjudge, the proposed model demonstrated improved efficiency by consuming less runtime while maintaining the enriched semantic analysis capabilities.

6.2.2. Statistical comparison

This work considered the corrected paired two-tailed t-test (Jiang et al., 2019b), paired t-test (Demšar, 2006), and the Wilcoxon signed-ranks test (Jiang et al., 2019a) for statistical comparison of the proposed model with the baselines for sections prediction. Specifically, the paired t-test is considered because the label imbalance problem arises when applying k-fold cross-validation, and the IJMT dataset features separate train and test sets that are balanced in labels and do not overlap. All proposed and baseline models are repeatedly run for six different epoch counts to draw the model results.

Interpretation:

The results in Table 20 show that the proposed model significantly outperforms both the baseline models across all three statistical tests.

- **Corrected paired two-tailed t-test:** Both comparisons yield very high t-statistics and extremely low p-values (< 0.001), indicating a strong statistically significant improvement in the performance of the proposed model. This indicates that the proposed model consistently outperformed the baseline models.

Table 21
Performance comparison of the proposed model with the state-of-the-art LLMs on the IJMT dataset for sections prediction. The table represents the macro weights.

Model	P	R	F1
GPT-4.1	10.34	8.91	9.43
GPT-4o	5.17	6.32	5.21
GPT-3.5	8.62	7.18	7.70
Proposed model	**71.274**	**72.675**	**71.21**

- **Wilcoxon signed ranks test:** The test showed a statistically significant difference in favor of the proposed model (W = 0.0, P_value = 0.03125), indicating that the proposed model consistently and significantly outperformed the baseline models.
- **Paired t-test:** The high t-statistics and very small P_values again confirm the superiority of the proposed model, providing additional statistical evidence.

The results from both parametric (corrected paired two-tailed and paired t-tests) and non-parametric (Wilcoxon signed ranks test) methods consistently demonstrated that the proposed model has achieved statistically significant improvement in performance compared to the baseline models of longformer and topjudge. These findings strongly support the effectiveness and robustness of the proposed approach.

6.2.3. Comparison with LLMs

The case documents in the IJMT dataset are lengthy, containing hundreds of sentences, i.e., thousands of tokens per document. To address the token limitations of large language models, only ten documents were selected for training, the minimum count required to fine-tune the LLMs effectively. Additionally, five documents were randomly chosen from the test set for evaluation, with the assumption that model performance would be consistent across the remaining test documents.

The performance of state-of-the-art LLMs on the IJMT dataset for applicable law sections prediction is presented in Table 21. Despite fine-tuning, these LLMs exhibited relatively poor performance, primarily due to their limited ability to interpret and understand lengthy, unstructured legal texts that contain complex legal vocabulary, which is a known challenge in the legal domain. A similar performance is observed in Table 13, which reports LLM performance on the ILSI dataset, where models were not adequately fine-tuned.

These results highlight the need for domain-specific models, such as the proposed model, which are better equipped to handle the inherent challenges of legal texts. Unlike general-purpose pretrained models such as PLMs and LLMs, specialized models offer improved interpretation and more reliable outcomes in legal tasks.

7. Conclusion and future work

This study contributes to the advancement of legal judgment prediction for the Indian legal system by addressing its inherent complexities. A standardized format for Indian criminal case judgment outcomes has been developed through the analysis of thousands of real-time solved cases. This format contains all essential components that influence judicial decisions, providing a structured foundation for further modeling and analysis.

Table A.22
Description of MT_Dataset_IPC_1860_Act_Des dataset columns.

S.No	Column	Description
1	Section	Section number
2	Description	IPC section description of the crime act
3	IPC_prison	Boolean value: Yes/No. Specifies whether the person is eligible to impose prison under the section
4	IPC_prison_val	Maximum prison term to be imposed
5	IPC_prison_type	Type of imprisonment, either simple or rigorous
6	IPC_fine	Boolean value: Yes/No. Specifies whether the person is eligible to impose fine under the section
7	IPC_fine_val	Maximum fine amount to be imposed under the section
8	Both	Both prison and fine can be imposed under the section

To address the lack of a comprehensive dataset for multi-task prediction of the Indian judicial system, the IJMT dataset was created, which facilitates the prediction of key judgment components for murder cases. The dataset integrated the case facts with evidence and exceptions to ensure a realistic and contextually accurate representation of judicial records, which is crucial for developing reliable AI-based legal tools.

A sequential dependency analysis model incorporating a lexical-vector dependency analysis engine was developed to predict 14 critical components, including applicable IPC sections, various prison terms, fines, default terms as penalties, and prison_type. A context-aware threshold modulation strategy was devised to mitigate the prevalent data imbalance issue in the legal context. The superior performance of the proposed model when evaluated on the IJMT dataset compared with various baselines and datasets underscores its robustness and the dataset's real-time adaptability for legal experiments.

Future extensions aim to improve prediction accuracy for fines and prison_type through refined methodologies and expand the framework to predict prison_imprison_type and default_imprison_type. These advancements will enhance the utility of the IJMT dataset and framework in addressing the diverse legal challenges.

CRediT authorship contribution statement

Prameela Madambakam: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Himangshu Sarma:** Writing – review & editing, Supervision, Resources, Project administration, Investigation, Conceptualization.

Consent to participate

All participants provided informed consent to participate in the study.

Consent for publication

All authors consent to the publication of this manuscript.

Ethics approval

This research study was conducted in compliance with all relevant ethical standards and guidelines.

Disclosure of AI Tools

The authors declared that no generative AI and AI-assisted technologies were used during the preparation of this work.

Funding

No external funding was received for this research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors express their heartfelt gratitude to Mr. Mallikarjun Rao, an advocate practicing in the Tirupati District Court, Andhra Pradesh, for his invaluable insights into the application of IPC provisions, the judicial process involved in delivering verdicts, and the feasibility of constructing a multi-task dataset incorporating IPC sections for addressing murder cases.

The authors also thank Ms. Sunitha, an advocate practicing at the Venkatagiri Judicial Magistrate Court, Andhra Pradesh, for generously sharing her knowledge on IPC sections.

Special thanks are extended to Mr. Gada, an advocate and retired public prosecutor, as well as a former special vigilance prosecutor in the Odisha State Court, for his expert guidance on IPC sections related to murder cases. His support greatly aided the creation of the multi-task dataset corpus by identifying murder cases within a broader criminal case information source. His expertise significantly influenced the methodology of the IJMT dataset creation and the development of the proposed approach.

Additionally, the authors extend their appreciation to the following students for their valuable contributions to the dataset creation: Ms. C. Bhavishya, Ms. Chalimadugu Tejaswi Reddy, Mr. Paladgula Yashwanth, Mr. Ravi Ram Sai Tej, Mr. Sajja Balaji Sai Surya, and Ms. Pithani Harshitha. Their dedication and efforts were instrumental in expediting the process.

Appendix

This section provides a list of tables used in the creation of the IJMT dataset. Table A.22 outlines the fields of the supplementary dataset titled MT_Dataset_IPC_1860_Act_Des, while Table A.23 presents the fields of the final comprehensive dataset named IJMT_IPC_Des.

Data availability

Data will be made available on request.

Table A.23
Description of IJMT_IPC_Des dataset columns.

S.No	Column name	Description
1	Id	Indian Kanoon document ID used to create the IJMT legal dataset corpus, serving as a reference to the original source case file.
2	Split	Specifies whether the document belongs to the training, testing, or validation set.
3	District	Name of the district court where the case was processed.
4	Year	Specifies the year in which the case was resolved, as documented in the Indian Kanoon records.
5	Facts and evidence	Describes facts of the crime event, evidence collected and proved, and exceptions requested to consider during the judgment process.
6	Punishment	Specifies the punishment status of the case, where the accused is either convicted or acquitted. This dataset considered only convicted cases.
7	Section1	Specifies one of the seven murder sections (299, 300, 302, 304, 304A, 304B, and 307) under which major crime is proved with evidence.
8	Sec_1_des	Description of murder section1 listed under MT_Dataset_IPC_1860_Act_des dataset referred from the IPC Act 1860. It illustrates the crime event, limitations, and applicability of prison1 and fine1 terms.
9	Prison1	Imprisonment for the murder section1, decided after considering the severity of the major crime event and exceptions requested in the facts and evidence by following the limitations mentioned in sec_1_des.
10	Prison1_imprision_type	Clarifies the nature of implementing prison1 imprisonment which can be either simple or rigorous.
11	Fine1	Fine amount for the murder section1, decided after considering the prison1 term, the severity of the major crime event, and exceptions requested in the facts and evidence by following the limitations mentioned in sec_1_des.
12	Default1	Default imprisonment for the murder section1, decided after considering the fine1 term, the severity of the major crime event, and exceptions requested in the facts and evidence.
13	Default1_imprision_type	Clarifies the nature of implementing default1 imprisonment, which can be either simple or rigorous.
14	Section2	Specifies one of the additional non-murder sections from the total 50 under which a supporting crime one has been proved with evidence.
15	Sec_2_des	Description of non-murder section2 listed under MT_Dataset_IPC_1860_Act_des dataset referred from the IPC Act 1860. It illustrates the crime event, limitations, and applicability of prison2 and fine2 terms.
16	Prison2	Imprisonment for the supporting crime section2, decided after considering the severity of the crime event and exceptions requested in the facts and evidence by following the limitations mentioned in sec_2_des.
17	Prison2_imprision_type	Clarifies the nature of implementing prison2 imprisonment which can be either simple or rigorous.
18	Fine2	Fine amount for the supporting crime section2, decided after considering the prison2 term, the severity of the crime event, and exceptions requested in the facts and evidence by following the limitations mentioned in sec_2_des.
19	Default2	Default imprisonment for the supporting crime section2, decided after considering the fine2 term, the severity of the crime event, and exceptions requested in the facts and evidence.
20	Default2_imprision_type	Clarifies the nature of implementing default2 imprisonment, which can be either simple or rigorous.
21	Section3	Specifies one of the additional non-murder sections from the total 50 under which a supporting crime two has been proved with evidence.
22	Sec_3_des	Description of non-murder section3 listed under MT_Dataset_IPC_1860_Act_des dataset referred from the IPC Act 1860. It illustrates the crime event, limitations, and applicability of prison3 and fine3 terms.
23	Prison3	Imprisonment for the supporting crime section3, decided after considering the severity of the crime event and exceptions requested in the facts and evidence by following the limitations mentioned in sec_3_des.
24	Prison3_imprision_type	Clarifies the nature of implementing prison3 imprisonment which can be either simple or rigorous.
25	Fine3	Fine amount for the supporting crime section3, decided after considering the prison3 term, severity of the crime event, and exceptions requested in the facts and evidence by following the limitations mentioned in sec_3_des.
26	Default3	Default imprisonment for the supporting crime section3, decided after considering the fine3 term, severity of the crime event, and exceptions requested in the facts and evidence.
27	Default3_imprision_type	Clarifies the nature of implementing default3 imprisonment, which can be either simple or rigorous.
28	Section4	Specifies one of the additional non-murder sections from the total 50 under which a supporting crime three has been proved with evidence.
29	Sec_4_des	Description of non-murder section4 listed under MT_Dataset_IPC_1860_Act_des dataset referred from the IPC Act 1860. It illustrates the crime event, limitations, and applicability of prison4 and fine4 terms.
30	Prison4	Imprisonment for the supporting crime section4, decided after considering the severity of the crime event and exceptions requested in the facts and evidence by following the limitations mentioned in sec_4_des.
31	Prison4_imprision_type	Clarifies the nature of implementing prison4 imprisonment which can be either simple or rigorous.
32	Fine4	Fine amount for the supporting crime section4, decided after considering the prison4 term, the severity of the crime event, and exceptions requested in the facts and evidence by following the limitations mentioned in sec_4_des.
33	Default4	Default imprisonment for the supporting crime section4, decided after considering the fine4 term, severity of the crime event, and exceptions requested in the facts and evidence.
34	Default4_imprision_type	Clarifies the nature of implementing default4 imprisonment which can be either simple or rigorous.
35	Prison_type	Determines the total imprisonment duration for a criminal to be punished when more than one prison term is imposed. It is indicated by either consecutive or concurrent.

(*continued on next page*)

Table A.23 (*continued*).

S.No	Column name	Description
36	Total_prison	Specifies the total imprisonment duration for a criminal to be punished by considering the prison_type value.
37	Total_fine	Determines the total fine to be paid by the offender, calculated as the $sum(Fine1, Fine2, Fine3, Fine4)$.
38	Other_judgment	Includes the other statements delivered as part of the judgment.

References

Aumiller, D., Chouhan, A., Gertz, M., 2022. EUR-lex-sum: A multi- and cross-lingual dataset for long-form summarization in the legal domain. arXiv:2210.13448. URL https://arxiv.org/abs/2210.13448.

Beltagy, I., Peters, M.E., Cohan, A., 2020. Longformer: The long-document transformer. ArXiv abs/2004.05150. URL https://api.semanticscholar.org/CorpusID:215737171.

Chalkidis, I., Androutsopoulos, I., Aletras, N., 2019. Neural legal judgment prediction in english. In: Korhonen, A., Traum, D., Màrquez, L.s. (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp. 4317–4323. http://dx.doi.org/10.18653/v1/P19-1424, URL https://aclanthology.org/P19-1424.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I., 2020. LEGAL-BERT: The muppets straight out of law school. In: Cohn, T., He, Y., Liu, Y. (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics, Online, pp. 2898–2904. http://dx.doi.org/10.18653/v1/2020.findings-emnlp.261, URL https://aclanthology.org/2020.findings-emnlp.261.

Chalkidis, I., Jana, A., Hartung, D., Bommarito, M., Androutsopoulos, I., Katz, D., Aletras, N., 2022. LexGLUE: A benchmark dataset for legal language understanding in english. In: Muresan, S., Nakov, P., Villavicencio, A. (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Dublin, Ireland, pp. 4310–4330. http://dx.doi.org/10.18653/v1/2022.acl-long.297, URL https://aclanthology.org/2022.acl-long.297.

Chen, Y.-S., Chiang, S.-W., Wu, M.-L., 2022. A few-shot transfer learning approach using text-label embedding with legal attributes for law article prediction. Appl. Intell. 52 (3), 2884–2902. http://dx.doi.org/10.1007/s10489-021-02516-x.

Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. ArXiv abs/1412.3555. URL https://api.semanticscholar.org/CorpusID:5201925.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. 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). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. http://dx.doi.org/10.18653/v1/N19-1423, URL https://aclanthology.org/N19-1423.

Fang, B., Cohn, T., Baldwin, T., Frermann, L., 2023. Super-SCOTUS: A multi-sourced dataset for the supreme court of the US. In: Preotiuc-Pietro, D., Goanta, C., Chalkidis, I., Barrett, L., Spanakis, G., Aletras, N. (Eds.), Proceedings of the Natural Legal Language Processing Workshop 2023. Association for Computational Linguistics, Singapore, pp. 202–214. http://dx.doi.org/10.18653/v1/2023.nllp-1.20, URL https://aclanthology.org/2023.nllp-1.20.

Hao, Y., Cao, X., Fang, Y., Xie, X., Wang, S., 2021. Inductive link prediction for nodes having only attribute information. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. IJCAI '20.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Jiang, L., Zhang, L., Li, C., Wu, J., 2019a. A correlation-based feature weighting filter for naive Bayes. IEEE Trans. Knowl. Data Eng. 31 (2), 201–213. http://dx.doi.org/10.1109/TKDE.2018.2836440.

Jiang, L., Zhang, L., Yu, L., Wang, D., 2019b. Class-specific attribute weighted naive Bayes. Pattern Recognit. 88, 321–330. http://dx.doi.org/10.1016/j.patcog.2018.11.032, URL https://www.sciencedirect.com/science/article/pii/S0031320318304205.

Joshi, A., Paul, S., Sharma, A., Goyal, P., Ghosh, S., Modi, A., 2024. IL-TUR: Benchmark for Indian legal text understanding and reasoning. In: Ku, L.-W., Martins, A., Srikumar, V. (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Bangkok, Thailand, pp. 11460–11499. http://dx.doi.org/10.18653/v1/2024.acl-long.618, URL https://aclanthology.org/2024.acl-long.618.

Kim, Y., 2014. Convolutional neural networks for sentence classification. In: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. EMNLP, Association for Computational Linguistics, Doha, Qatar, pp. 1746–1751. http://dx.doi.org/10.3115/v1/D14-1181, URL https://aclanthology.org/D14-1181.

Kowsrihawat, K., Vateekul, P., Boonkwan, P., 2018. Predicting judicial decisions of criminal cases from thai supreme court using bi-directional GRU with attention mechanism. In: 2018 5th Asian Conference on Defense Technology. ACDT, pp. 50–55. http://dx.doi.org/10.1109/ACDT.2018.8592948.

Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. ICML '14, JMLR.org, pp. II–1188–II–1196.

Lippi, M., Pał ka, P.a., Contissa, G., Lagioia, F., Micklitz, H.-W., Sartor, G., Torroni, P., 2019. CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service. Artif. Intell. Law 27 (2), 117–139. http://dx.doi.org/10.1007/s10506-019-09243-2.

Luo, B., Feng, Y., Xu, J., Zhang, X., Zhao, D., 2017. Learning to predict charges for criminal cases with legal basis. In: Palmer, M., Hwa, R., Riedel, S. (Eds.), Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Copenhagen, Denmark, pp. 2727–2736. http://dx.doi.org/10.18653/v1/D17-1289, URL https://aclanthology.org/D17-1289.

Lyu, Y., Wang, Z., Ren, Z., Ren, P., Chen, Z., Liu, X., Li, Y., Li, H., Song, H., 2022. Improving legal judgment prediction through reinforced criminal element extraction. Inf. Process. Manage. 59 (1), 102780. http://dx.doi.org/10.1016/j.ipm.2021.102780, URL https://www.sciencedirect.com/science/article/pii/S0306457321002600.

Madambakam, P., Rajmohan, S., 2022. A study on legal judgment prediction using deep learning techniques. In: 2022 IEEE Silchar Subsection Conference. SILCON, pp. 1–6. http://dx.doi.org/10.1109/SILCON55242.2022.10028879.

Madambakam, P., Rajmohan, S., Sharma, H., Gupta, T.A.C.P., 2023. SLJP: Semantic extraction based legal judgment prediction. arXiv:2312.07979.

Malik, V., Sanjay, R., Kumar Nigam, S., Ghosh, K., Guha, S., Bhattacharya, A., Modi, A., 2021. ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. http://dx.doi.org/10.18653/v1/2021.acl-long.313.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. In: Proceedings of Workshop at ICLR 2013. p. 2013.

Nigam, S., Sharma, A., Khanna, D., Shallum, N., Ghosh, K., Bhattacharya, A., 2024. Legal judgment reimagined: Predex and the rise of intelligent AI interpretation in Indian courts. In: Ku, L.-W., Martins, A., Srikumar, V. (Eds.), Findings of the Association for Computational Linguistics: ACL 2024. Association for Computational Linguistics, Bangkok, Thailand, pp. 4296–4315. http://dx.doi.org/10.18653/v1/2024.findings-acl.255, URL https://aclanthology.org/2024.findings-acl.255.

Paul, S., Bhatt, R., Goyal, P., Ghosh, S., 2024. Legal statute identification: A case study using state-of-the-art datasets and methods. In: Yang, G.H., Wang, H., Han, S., Hauff, C., Zuccon, G., Zhang, Y. (Eds.), Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024. ACM, pp. 2231–2240. http://dx.doi.org/10.1145/3626772.3657879.

Paul, S., Goyal, P., Ghosh, S., 2020. Automatic charge identification from facts: A few sentence-level charge annotations is all you need. In: Scott, D., Bel, N., Zong, C. (Eds.), Proceedings of the 28th International Conference on Computational Linguistics. International Committee on Computational Linguistics, Barcelona, Spain (Online), pp. 1011–1022. http://dx.doi.org/10.18653/v1/2020.coling-main.88, URL https://aclanthology.org/2020.coling-main.88.

Paul, S., Goyal, P., Ghosh, S., 2021. LeSICiN: A heterogeneous graph-based approach for automatic legal statute identification from Indian legal documents. http://dx.doi.org/10.48550/arXiv.2112.14731.

Paul, S., Mandal, A., Goyal, P., Ghosh, S., 2023. Pre-trained language models for the legal domain: A case study on Indian law. In: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law. ICAIL '23, Association for Computing Machinery, New York, NY, USA, pp. 187–196. http://dx.doi.org/10.1145/3594536.3595165.

Pennington, J., Socher, R., Manning, C., 2014. Glove: Global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543. http://dx.doi.org/10.3115/v1/D14-1162, URL https://aclanthology.org/D14-1162.

Tuggener, D., von Däniken, P., Peetz, T., Cieliebak, M., 2020. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In: Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, pp. 1235–1241, URL https://aclanthology.org/2020.lrec-1.155.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS '17, Curran Associates Inc., Red Hook, NY, USA, pp. 6000–6010.

Wang, P., Fan, Y., Niu, S., Yang, Z., Zhang, Y., Guo, J., 2019. Hierarchical matching network for crime classification. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '19, Association for Computing Machinery, New York, NY, USA, pp. 325–334. http://dx.doi.org/10.1145/3331184.3331223.

Wang, P., Yang, Z., Niu, S., Zhang, Y., Zhang, L., Niu, S., 2018. Modeling dynamic pairwise attention for crime classification over legal articles. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. SIGIR '18, Association for Computing Machinery, New York, NY, USA, pp. 485–494. http://dx.doi.org/10.1145/3209978.3210057.

Xiao, C., Zhong, H., Gu, J., Tu, C., Liu, Z., Sun, M., 2018. CAIL2018: A large-scale legal dataset for judgment prediction. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1885–1897.

Xu, N., Wang, P., Chen, L., Pan, L., Wang, X., Zhao, J., 2020. Distinguish confusing law articles for legal judgment prediction. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, pp. 3086–3095. http://dx.doi.org/10.18653/v1/2020.acl-main.280, URL https://aclanthology.org/2020.acl-main.280.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V., 2019a. XLNet: generalized autoregressive pretraining for language understanding. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA.

Yang, W., Jia, W., Zhou, X., Luo, Y., 2019b. Legal judgment prediction via multi-perspective bi-feedback network. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization, pp. 4085–4091. http://dx.doi.org/10.24963/ijcai.2019/567.

Yang, S., Tong, S., Zhu, G., Cao, J., Wang, Y., Xue, Z., Sun, H., Wen, Y., 2022. MVE-flk: A multi-task legal judgment prediction via multi-view encoder fusing legal keywords. Knowl.-Based Syst. 239, 107960. http://dx.doi.org/10.1016/j.knosys.2021.107960, URL https://www.sciencedirect.com/science/article/pii/S095070512101090X.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E., 2016. Hierarchical attention networks for document classification. In: Knight, K., Nenkova, A., Rambow, O. (Eds.), Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, San Diego, California, pp. 1480–1489. http://dx.doi.org/10.18653/v1/N16-1174, URL https://aclanthology.org/N16-1174.

Yao, F., Sun, X., Yu, H., Yang, Y., Zhang, W., Fu, K., 2020. Gated hierarchical multi-task learning network for judicial decision prediction. Neurocomputing 411, 313–326. http://dx.doi.org/10.1016/J.NEUCOM.2020.05.018.

Yao, F., Sun, X., Yu, H., Zhang, W., Fu, K., 2021. Commonalities-, specificities-, and dependencies-enhanced multi-task learning network for judicial decision prediction. Neurocomputing 433, 169–180. http://dx.doi.org/10.1016/J.NEUCOM.2020.10.010.

Zheng, L., Guha, N., Anderson, B.R., Henderson, P., Ho, D.E., 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. ICAIL '21, Association for Computing Machinery, New York, NY, USA, pp. 159–168. http://dx.doi.org/10.1145/3462757.3466088.

Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., Sun, M., 2018. Legal judgment prediction via topological learning. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, pp. 3540–3549. http://dx.doi.org/10.18653/v1/D18-1390, URL https://aclanthology.org/D18-1390.