Proceedings of the Pakistan Academy of Sciences: A Physical and Computational Sciences 62(1): 41-51 (2025)

ISSN (Print): 2518-4245; ISSN (Online): 2518-4253
http://doi.org/10.53560/PPASA(62-1)880

Pakistan Academy of Sciences

Research Article

A Text Mining Approach for Automated Case Classification of Judicial Judgment

Saad Rasool[1], Israr Hanif[1], Qaisar Rasool[1], Humaira Afzal[1], and Muhammad Rafiq Mufti[2*]

[1]Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan
[2]Department of Computer Science, COMSATS University Islamabad, Vehari Campus, Pakistan

Abstract: Court judgments are based on legal reasoning, evidence, and judicial decisions. However, many judgments are difficult to understand due to their length and complex language. Judges and attorneys often cite legal rulings to interpret regulations effectively. Legal education for lawyers, judges, and trainees benefits from judicial insights into legal reasoning, legislative interpretation, and legal standards. Accurate classification of legal judgments requires sophisticated methodologies. The increasing number of new and pending cases adds to the courts' workload, highlighting the need for efficient classification. Since only a limited number of court rulings are decided each year, many cases remain unresolved and are carried forward to the following year. Despite efforts to improve classification accuracy across judicial judgment datasets, Pakistani court judgments still lack a proper classification system. To address this gap, we compiled a dataset of criminal judgments from the Pakistani High and Supreme Courts. We developed a machine learning model using neural networks and a transformer architecture to achieve accurate classification. Our approach incorporated Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and DistilBERT models, taking into account the dataset's volume and unique characteristics. Our research explores optimal classification algorithms tailored to Pakistan's legal landscape. After comparison, the DistilBERT transformer model outperformed the others, achieving an accuracy of 98%. It demonstrated an exceptional ability to understand contextual semantics and effectively handle the complexities of multi-label classification in legal judgments. The contributions of our work include the development of a new dataset of Pakistani court rulings, advancements in legal research, and an automated case classification system designed to streamline judicial procedures and enhance access to justice.

Keywords: Judicial Judgment, Judgment Classification, Pakistani Court, Machine Learning, Legal Knowledge.

1. INTRODUCTION

Legal judgments play a crucial role in judicial administration, as they represent the court's final decisions on verifiable issues and legal matters. Legal proceedings culminate in judgments that analyze evidence, legal arguments, and the court's conclusions. However, lawyers, lawmakers, and scholars often face significant challenges due to the sheer volume and complexity of court judgments. Court opinions frequently involve intricate language and reasoning, which can vary considerably. These judgments are often difficult to comprehend, as they include extensive legal vocabulary, precedents, and references. Lawyers spend countless hours analyzing lengthy opinions to identify relevant case precedents [1]. Legal specialists retrospectively examine cases to gain deeper insights into the factors that significantly influence judicial decisions [2]. Even individuals with strong comprehension skills must carefully read each judgment; therefore, legal determinations and case law are essential tools for legal education [3]. Classifying legal determinations is particularly challenging due to the diversity of legal language, the complexity of legal terminology, and the need for thorough analysis to distinguish between cases.

Since most people struggle to understand legal texts, lawyers must review and extract key information for end users [4]. The broad scope of legal document analysis is further complicated by the wide range of issues and concerns encountered by non-legal professionals and trial participants [5].

Received: January 2025; Revised: February 2025; Accepted: March 2025
* Corresponding Author: Muhammad Rafiq Mufti <rafiq_mufti@ciitvehari.edu.pk>

Judicial judgments employ specialized terminology, constitutional and case law references, and legal jargon. Traditional methods designed for general language processing often struggle to effectively classify this specialized legal language. Judicial judgments vary based on the authority, court level, and the nature of the case, with some involving more complex reasoning than others. Developing a predictive model is particularly challenging, as court proceedings depend on numerous factors that differ by case type [2]. Judicial language frequently contains ambiguity, as it blends legal reasoning with specific case facts. Criminal judgments span a broad spectrum of legal issues, encompassing both criminal and civil matters, which further complicates classification and information extraction.

Pakistan's court system has a massive backlog, and hundreds of rulings each year make problems worse. Although the courts have tried to reduce this backlog, many cases remain unresolved. These pending cases continue to add to the legal system's workload each year. New cases further increase court workloads, complicating the situation and delaying justice. In Australia, thousands of judgments are issued annually, but attorneys can only analyze a few, overlooking larger trends hidden within the cases [1]. A lack of adequate legal representation is a major reason contributing to the number of unsolved cases [6].

The backlog is exacerbated by administrative inefficiencies, such as procedural delays, inadequate court capacity, and insufficient funding. Moreover, the delays may be exacerbated by the competitive nature of lawsuits and the complex structure of judicial proceedings. These factors contribute to the growing number of cases, highlighting the need for efficient solutions to streamline judicial operations. As the number of pending cases continues to increase, there is an urgent need for strategies to reduce the court workload and improve legal system efficiency. Artificial Intelligence (AI) has the potential to address the challenges in both common law and civil law systems, including case congestion and delays in justice administration [3, 5]. Without intervention, the backlog will keep accumulating, resulting in a prolonged delay in justice. Efficient classification systems are essential to organize cases by their nature and category to tackle the court backlog.

Many studies have classified judicial cases in several ways. Traditional human annotation methods are applied to advanced machine learning and deep learning algorithms. In manual annotation systems, skilled personnel label decisions based on content and context. While successful, these procedures can be costly and time-consuming. Advanced algorithms, including ML, and Natural Language Processing (NLP) models, can automate the classification of legal rulings based on their semantic structure and linguistic aspects. Kwak *et al.* [7] presented a unique and useful challenge by providing a Korean Legal dataset [8]. They also employed multiple methods to find a distinct fine-tuning approach for small datasets. All models, except Multinomial Naïve Bayes (NB), achieved accuracy values above 63%. The Paraphrase-mpnet-basev2, fine-tuned with SetFit, outperformed all other models with 70.5% accuracy, a difference of at least 6% points. The second-best model was SVM, with 64.5% accuracy. Barman *et al.* [9] showed the efficiency of ML in processing Hindi legal data using Convolutional Neural Network (CNN) architecture on the Hindi Legal Documents Corpus [2, 9-12]. They classified Indian legal data [13-16] into two groups using convolutional layers. This model learns Bail Prediction binary classification from the pre-processed dataset, improving forecast accuracy and showing how ML can support legal decision-making. The researcher improved bail prediction test accuracy to 93% using data from 20 districts in Uttar Pradesh.

To improve the reliability of training data, Javed and Li [17] addressed semantic bias in legal judgment. The semantic bias in the CAIL dataset [18-21] was classified and identified using general-purpose AI. AI models outperformed structured professional risk assessment techniques in the CAIL dataset, which contains hundreds of incidents. Creative AI-based technologies could help lawyers. Semantic bias classification was used to detect bias in legal judgment. One legal approach classifies and identifies bias using the CAIL dataset, enabling to uncovering of semantic biases in court rulings. SVM, NB, MLP, and K-Nearest Neighbor classifiers were applied, with the SVM classifier outperforming the others, achieving 96.90% accuracy. Abbara *et al.* [22] used Arabic case materials to predict child custody and marriage annulment decisions using deep learning and NLP. PDF text was utilized to generate an Arabic judicial judgment prediction dataset for Saudi Personal Status cases. All 49 cases in this dataset have legitimate verdicts. The new dataset was analyzed using various DL and ML models, including Term Frequency-Inverse Document Frequency (TF-IDF), word2vec, LSTM, BiLSTM, and Logistic Regression. In the experiments, the SVM model with word2vec made the most accurate custody predictions (88%), and the LR model with TF-IDF made the most accurate annulment predictions (78%). Strickson and Iglesia [23] developed the LJP model, which was customized for UK court cases. Machine learning models were applied to a 100-year-old tagged UK judgments dataset of court decisions, creating a reliable, understandable prediction model. Both vector space features performed well; however, TF-IDF and LR achieved the best F1 score. Using n-grams and topic clusters as predictive feature sets was successful, and SLP and MLP algorithms further improved the results. Various feature formats and algorithms were tested, with their best model achieving an F1 score of 69.02 and an accuracy of 69.05%.

Shelar *et al.* [24] proposed legal annotation ideas for Indian criminal bail petitions using machine learning. Multiple research tests were conducted on 17 common ML models and compared using standard evaluation methods. 'SmartLawAnnotator' was developed as an open-source legal annotation suggestion system. Legal professionals labeled 2,000 samples using this approach. In smaller sample sizes, SVM models achieved the highest accuracy reaching up to 82%. Arriba-Pérez *et al.* [25] suggested a hybrid system using machine learning to categorize multi-label judgments (sentences) in the Spanish Legal System. The system integrates visual and natural language descriptions for explanation, NLP techniques, and advanced legal reasoning to identify parties involved in the cases. Their advanced legal AI system predicts and explains court decisions using ML, along with visual and plain language representation at the feature level. Previously, no known attempt had been made to mechanically recognize court decisions using multiple labels and high-quality natural language explanations. Legal professional-annotated tagged data achieved a micro accuracy of up to 85%.

Zahir [26] used Arabic written accounts to train a deep learning system for predicting judicial case outcomes. A deep learning model was trained in-house on Moroccan Court of Cassation cases to predict judgment. An innovative data augmentation strategy improved prediction performance due to the small corpus. Several deep learning model architectures along with FastText and GloVe embedding settings, were tested and analyzed. The method achieved an accuracy of 80.51% in predicting Moroccan Court of Cassation outcomes across six categories. To differentiate between classifying and anticipating court outcomes, Medvedeva *et al.* [27] developed a standardized collection of European Court of Human Rights [28-30] judgments. The dataset includes submitted cases, admissibility decisions, and pre-processed final decisions. They demonstrate that predicting court outcomes is more challenging than classifying judgments and establishing a benchmark for future research. SVM, Hierarchical Bidirectional Encoder Representations from Transformers (H-BERT), and LEGAL-BERT were tested for both classification and prediction tasks. H-BERT and LEGEL-BERT outperformed SVM in classification, however, SVM achieved better Macro F-scores. Zheng *et al.* [31] aimed to identify the most common causes of Preprocessing, Processing, and Postprocessing (PPP) conflict, summarized the findings and predicted litigation outcomes. They identified 17 legal variables and analyzed their impact on litigation outcomes in 171 PPP lawsuit cases from 2013-2018, sourced from China Judgments Online [32-34]. Using these 171 instances, they trained and evaluated nine ML models with the "prediction approach". The ensemble models, i.e., GBDT, KNN, and MLP performed the best, surpassing the other nine ML models with a prediction accuracy of 96.42%.

In the context of the court system's rising workloads and constrained resources, the importance of automated solution aid in case classification is becoming ever more evident. So, the legal field has traditionally utilized natural language systems because of their text-centric nature. In recent years, there has been an increasing trend to apply NLP to a wider range of legal sectors. This research work aims to build an automated system of specifically "criminal" case classification for Pakistani courts through the integration of a traditional ML approach, neural network, and cutting-edge transformer model.

2. DATASET DEVELOPMENT

Sequentially, it describes the processes that were done to construct the dataset for this research, beginning with the collecting of data, followed by the formulation of the dataset, preprocessing, feature engineering, and ending with the splitting of the dataset. A description of the framework for this process can be found in Figure 1.

2.1. Data Collection

The legal judgments in our collection were acquired comprehensively. Collecting legal judgments, particularly criminal ones, involved systematically compiling court rulings and conclusions on criminal cases. This required obtaining relevant judicial documents that were not available online. Since legal judgments were not easily accessible via the 'PAKISTANLAWSITE' website [35], collecting these judgments required court attendance. To access judicial judgments, users had to purchase a subscription plan and create an account. Personal engagement with court officials authorized judgment collection. The judgments from the Pakistani High and Supreme Court were available online. To preserve text and structure, 110 criminal judgments were collected in Word file format.

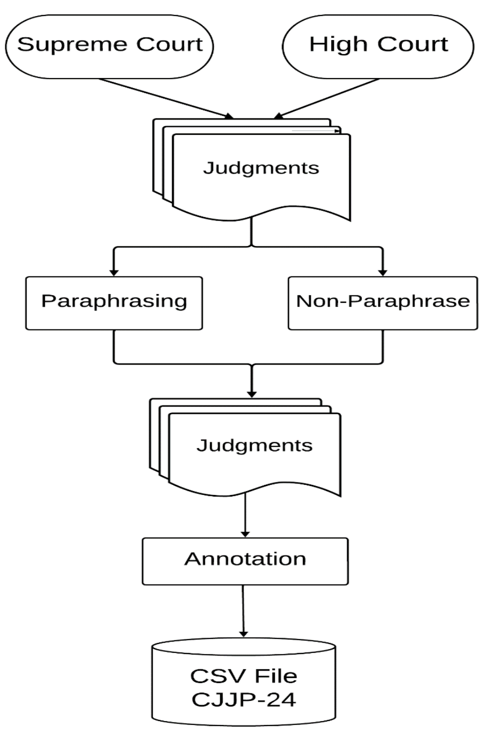

Fig. 1. Dataset creation process.

By utilizing the website's provided options, namely 'Full Judgment' and 'Head Notes', we were able to bypass a crucial step in summarizing judgments. In addition to the comprehensive and detailed judgments, the Head Notes, which originally represented the judge's actual order, were also included. Since complete judgments contain a significant amount of information on several non-essential features and components, we primarily gathered the Head Notes from the website. This method streamlined the data collection process and allowed us to concentrate exclusively on extracting the essential elements relevant to our classification objective.

2.2. Dataset Composition

Dataset construction is a crucial aspect of our work, as the integrity and organization of the data directly impact the study's conclusion. Acquiring a deep understanding of the dataset's nature is essential for effectively designing and developing its format during this phase. After careful deliberation, we determined that a multi-label classification is the most suitable approach for our dataset. Before proceeding with dataset generation, it is essential to fully understand the nature of the problem, which in this case is multi-label classification.

Unlike standard binary or multi-class classification, multi-label classification allows multiple labels to be assigned to each instance simultaneously. Legal field experts did the annotation; cross-checking among annotators assured precision and consistency. Most of the judgments targeted more than one label at once in the dataset. Multi-label text classification is an NLP task that involves assigning a text sequence to multiple categories at once [36]. This has significant implications for the administration of justice, as a single judgment may address multiple legal categories or offenses simultaneously. For instance, a judgment may encompass accusations of both theft and murder. Most of the judgments addressed more than one label at once in the dataset making it compulsory to adapt multi-label text classification nature.

2.3. Dataset Insights

To accurately represent the complexity of court decisions, particularly those involving multiple legal categories or offenses, we modified the multi-label classification process to construct our dataset. This approach lays the foundation for strong classification models capable of accurately predicting the presence of different legal categories in specific judgments. During the dataset formulation phase, our main focus was on developing a dataset optimized for multi-label classification. Initially, we identified six fundamental categories or subcategories relevant to the criminal domain: Murder, Dacoity, Rape, Fraud, Robbery or Theft, and Kidnapping.

Each category defines unique criminal offenses commonly encountered in legal proceedings. To organize our dataset, we created separate columns for each class. This framework effectively highlights the presence or absence of each category in individual cases, illustrating the complex and overlapping nature of criminal verdicts. Figure 2 illustrates the dataset's characteristics, target categories, and general organization using a multi-label classification approach.

2.4. Judgments Paraphrasing

The paraphrasing technique was employed to expand the dataset. To increase the number of instances, we used a paraphrasing tool to modify certain judgments. Specifically, the QuillBot AI tool was employed to rewrite the judgments by altering phrase structure and incorporating synonyms while maintaining the original meaning [37].

By employing this paraphrasing tool, we identified two significant advantages:

- It incorporates small structural changes, wording modifications, and synonyms. These variations enhance the model's training efficacy by introducing more versatile terminology.
- This method increases the number of instances in our dataset.

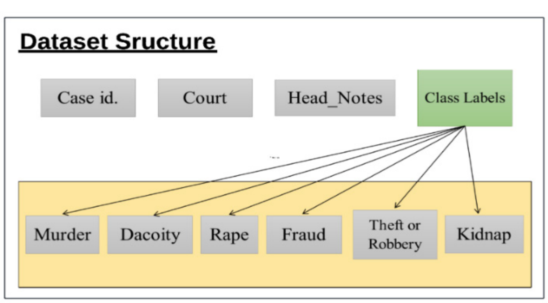

Fig. 2. Dataset structure.

A significant level of variability exists among court judgments regarding the judge's writing style [38]. By utilizing the paraphrasing technique, the total count of judgment instances in the dataset is now 221. Once the entire dataset was compiled, we designated it as "Criminal Judicial Judgments Pakistan (CJJP-24)". Paraphrased judgments were also thoroughly examined by legal annotators and experts to maintain their genuine legal value while providing textual versatility. Any type of ambiguous cases was discussed together to give the most suitable labels. The classification model will then be trained using this larger and more versatile corpus, which includes paraphrased judgments. This will improve its ability to generalize and identify subtle differences in legal language.

3. METHODOLOGY

Following a sequential approach, as illustrated in Figure 3, the procedures used to assess the performance of the adjusted methodology for this study are detailed.

3.1. Data Analysis and Visualization

The image aids in preprocessing and feature engineering tasks by displaying how the different text lengths are distributed in the dataset. It also provides an overview of the dataset's properties, which supports exploratory data analysis and informs decisions for subsequent models.

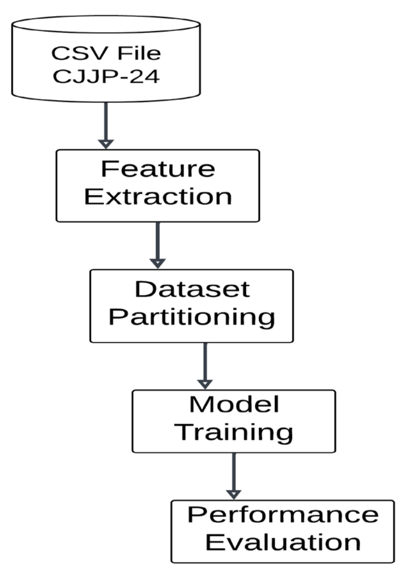

Fig. 3. Methodology process.

The text lengths in the dataset are shown in Figure 4. The highest point in the frequency distribution indicates that most judgment texts have word counts within a certain range. The figure also shows the distribution of judgment text lengths, highlighting any outliers or patterns. Additionally, the class correlation heatmap shown in Figure 5, shows the correlations between class labels in our multi-label classification dataset. This heatmap aids in classifying judicial judgments by showing how often specific classifications occur together. It displays the correlation coefficients between pairs of classes, with each column in the matrix representing the correlation value between two classes. The values range from -1 to 1, where '1' indicates a perfect positive correlation, meaning the presence of one category is consistently linked to the presence of the other. A correlation coefficient of '0' implies no association, while a negative value indicates a perfect negative correlation. Based on the heatmap, we can identify classes that frequently co-occur in judicial judgments as well as those that are more autonomous. This information is crucial for designing and interpreting our multi-label classification models.

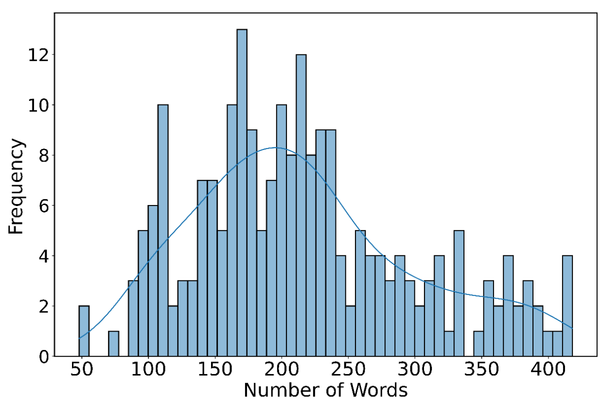

Fig. 4. Judgments text length graph.

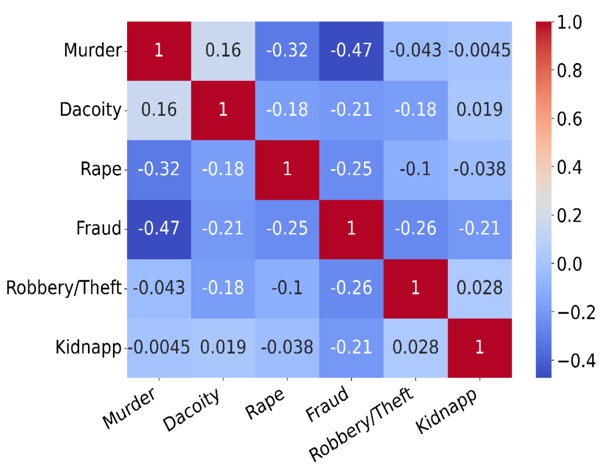

Fig. 5. Class correlation heatmap.

3.2. Data Preprocessing

Before building or using the prediction model, the judgment text must be preprocessed and feature extracted. This stage involves several essential processes to transform raw textual data into structural, model-ready data. To improve training data efficiency and quality, we prioritize text cleaning, tokenization, and vectorization. Additionally, implementing model modifications, such as encoding tokens and padding sequences, ensures compatibility with each model's input requirements. During the initial stage of text preprocessing, stop words are eliminated to improve the performance of subsequent processing steps. Removing frequently occurring stop words, such as 'and', 'the', and 'is', helps refine the text for classification purposes. The Natural Language Toolkit (NLTK) library has been employed to eliminate stop words for Support Vector Machine (SVM) processing. The NLTK stop words list includes 179 words, commonly used English words [39]. However, advanced neural network models, such as DistilBERT and Long Short-Term Memory (LSTM) do not require traditional stop-word removal for NLP tasks. In particular, Transformer models excel at extracting context-based information from complete phrase input sequences, even when stop words are present. This contrasts with traditional machine learning models, which primarily focus on word frequency.

The given text is separated into tokens, where long sentences are broken down into individual words. Tokens are separated using whitespaces and line breaks [40]. When training a machine to comprehend human language, tokenization, and vectorization strategies play a crucial role [41]. Machine learning models categorize text using feature information. Legal judgment classification involves extracting textual properties, such as words or sequences that convey meaningful patterns. These features are then converted into vectors, which are sequences of integers, so that, machine learning algorithms can process them. Most features are extracted automatically from large volumes of text. TF-IDF can be split into two components: TF and IDF. To evaluate the usefulness of word frequency, a phrase's frequency in the document must be examined. Inverse document frequency (IDF) measures a word's corpus prevalence. The number of documents d with a phrase t is referred to as document frequency.

We construct feature matrices for training and test sets by vectorizing judgments dataset using N-grams derived from the training set. N-grams are generated based on term co-occurrence frequencies within the corpus. The Count Vectorizer, a text preprocessing tool, converts a collection of text documents into a matrix of character frequencies. The Count Vectorizer tokenizes text data into words, building a lexicon with all unique terms from the training dataset. Each term in this dictionary is assigned a unique integer index. The text data is then transformed into a sparse matrix, where rows represent court judgments and columns correspond to dictionary terms. The matrix shows the frequency of each term within the given document. The LSTM model utilizes Count Vectorizer output, allowing it to analyze textual input by converting legal judgments into word frequency vectors. Applying the Count Vectorizer to the training set of judgments helps the LSTM model identify and understand text patterns and properties.

4. RESULTS AND DISCUSSION

The ML workflow requires model performance assessment to evaluate the efficacy and accuracy of our trained models. Three models were trained on our judicial judgments' dataset for classification. We have selected these three models based on two reasons; 1) To show the comparative analysis and difference in the results of using the machine learning model, LSTM, and the advanced transformer model, DistilBERT so that we can see how effective these advanced models are in performance compare to other. 2) As per our literature analysis, we conclude that in the machine learning category, SVM performs better in my legal text classification work among other ML models. Similarly, LSTM and BERT-based models have depicted outstanding performance for complex judgment text classification and sequence modeling. To determine the optimal parameters, we conducted multiple training and testing iterations. First, we vectorized the dataset and trained the SVM model. Then, we trained LSTM and DistilBERT models, carefully tracking training and validation loss. Pre-trained models were used for testing. Experimental results showed that the state-of-the-art transformer model achieved the best performance.

For correct model evaluation, we partitioned the dataset into training and testing parts, allocating 80% for training and 20% for testing, the 'train test split' technique. This technique split the whole dataset into two portions, one for training, and the other for testing the model. This partitioning method is widely used to ensure a lot of training data while maintaining a fair evaluation. We evaluated the effectiveness of SVM, LSTM, and DistilBERT, for classifying judicial judgments with multiple labels. Each model was assessed using key performance metrics, including accuracy, precision, recall, and F1 score. Table 1 provides a comprehensive comparison of their performances.

In summary, the evaluation results highlight the strengths and weaknesses of each model. SVM proves to be a reliable model for classifying legal judgments, exhibiting a well-balanced performance with a high precision of 95% and recall of 96%. The LSTM model also demonstrates strong performance, achieving a precision of 95% and an accuracy of up to 94%; however, its recall is relatively lower at 86%, indicating that it may not be able to identify specific pertinent classes. The DistilBERT transformer model remains the most efficient for this project, as its popularity is attributed to its high performance across several NLP tasks [42]. It outperforms all other models across key metrics, including precision, recall, and accuracy. Figure 6 illustrates the evaluation results for these models.

Table 1. Models performance analysis.

Models	Precision	Recall	F1-Score	Accuracy
SVM	95%	96%	95%	88%
LSTM	95%	78%	86%	94%
DistilBERT	95%	98%	96%	98%

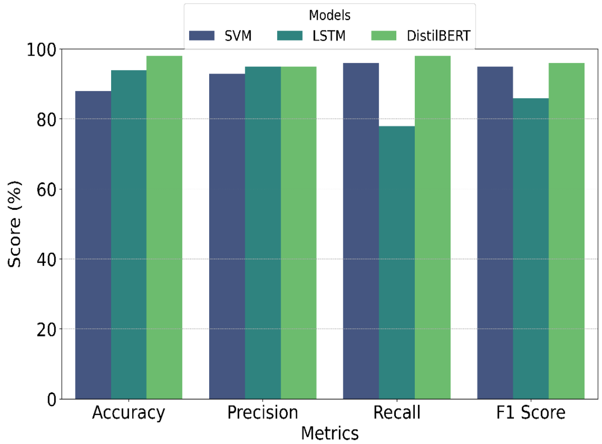

Fig. 6. Performance analysis graph.

The DistilBERT model is the most proficient and reliable methodology for the multi-label classification of judicial judgments. The SVM model ranks second, followed by the LSTM model as the third most proficient and reliable. This comprehensive assessment highlights the potential of advanced neural network models, particularly transformer-based models like DistilBERT, to achieve exceptional performance in complex classification problems. We conducted an experimental analysis by training all three models on the original judgment set and validating them using the 'train test split' technique. The dataset was partitioned into an 80-20 ratio for training and testing, respectively. The number of epochs was set to 5 for the LSTM model and 7 for the DistilBERT model.

Based on the results in Table 2, the SVM model exhibits high precision but moderate accuracy, reaching up to 0.55. This suggests that while the model effectively identifies true positive instances with few false positives, it struggles to capture all relevant classes, leading to a lower recall of 0.67. The LSTM model is adept at capturing the sequential relationships within judicial judgments. However, its relatively lower recall of 0.70 suggests that, despite excelling at generating accurate predictions, it may overlook certain categories. DistilBERT demonstrates outstanding performance, achieving a high accuracy of up to 0.98, along with precision, and recall, resulting in a remarkable F1 score of 0.95. This is attributed to its ability to capture contextual information and subtle subtleties in the text.

The results illustrated in Figure 7, were obtained by these models on the paraphrased judgments dataset. Our findings demonstrate that SVM may not handle the complexity and variability introduced by paraphrasing the judgments as effectively as neural network-based models [9, 13], leading to a lower accuracy of 0.59. While LSTM can better capture sequential dependencies, it may still struggle with the variability in paraphrased data, affecting recall (0.60). Despite being a powerful transformer-based model, DistilBERT's high precision of 0.95 but low recall of 0.53 suggests that while it is highly confident about its predictions, it misses many relevant instances, possibly due to overfitting. The results show that the meaning content of the paraphrased judgments was kept, but the models had a hard time with the new syntax that the paraphrasing tools added. This is especially true in situations with more than one label where the lines between labels can be fuzzy. It shows how much the models depend on original legal wording patterns and how hard it is to capture legal meanings through fake enhancement.

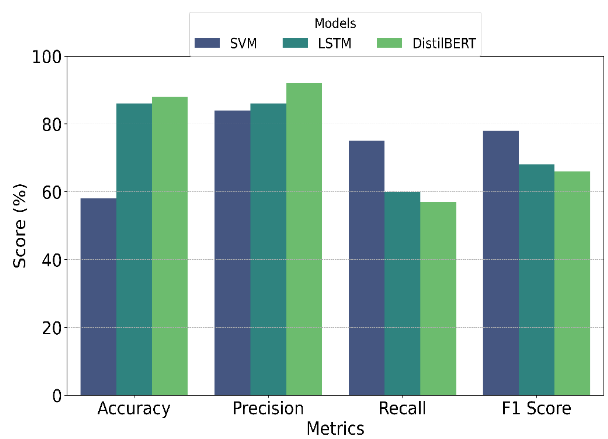

Fig. 7. Results on paraphrased judgments set.

Table 2. Results on the original judgments dataset.

Models	Precision	Recall	F1-Score	Accuracy
SVM	91%	67%	77%	55%
LSTM	91%	70%	79%	91%
DistilBERT	98%	93%	95%	98%

5. CONCLUSIONS

Court decisions are based on legal reasoning, facts, and judicial determinations. Judges and lawyers rely on legal rulings as precedents to interpret legislation. Classifying legal judgments requires sophisticated techniques to accurately categorize them. The increasing number of new and pending cases places a significant burden on courts, highlighting the need for efficient classification systems. Considerable efforts have been made to address these challenges and improve the categorization of legal decision datasets. However, the classification of Pakistani judicial verdicts remains underdeveloped. To address this gap, we created a large database of criminal case rulings from Pakistan's District, High, and Supreme Courts. Our data classification approach incorporates a standard machine learning model, a neural network, and an advanced transformer model. Traditional ML techniques such as TF-IDF Vectorizer with SVM classifier, were employed to handle the dataset's volume and unique characteristics. Additionally, we implemented more advanced models like LSTM and DistilBERT, which further enhanced the performance. Key results from our experiments include: DistilBERT outperformed the other models, achieving the highest accuracy of 98% on our dataset of judicial judgments. The SVM model demonstrated satisfactory performance, albeit with a slightly lower accuracy of 88%. On the other hand, the LSTM model achieved promising accuracy of up to 94%, however, it exhibited deficiencies in a recall. Future classification studies will examine more legal judgments and can be created by including constitutional, family, civil, and other rulings.

6. CONFLICT OF INTEREST

The authors declare no conflict of interest.

7. REFERENCES

1. W. Hwang, D. Lee, K. Cho, H. Lee, and M. Seo. A multi-task benchmark for korean legal language understanding and judgement prediction. *Advances in Neural Information Processing Systems* 35: 32537-32551 (2022).
2. V. Vaissnave and P. Deepalakshmi. A keyword-based multi-label text categorization in the Indian legal domain using bi-LSTM. Soft Computing: Theories and Applications*:* In Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing. T.K. Sharma, A.W. Ahn, O.P. Verma, and B.K. Panigrahi (Eds.). Volume 1380. *Springer, Singapore* pp. 213-227 (2022).
3. S. Agrawal. How To Read Court Judgments Quickly and Efficiently (2023). https://www.scribd.com/document/628208457/How-to-Read-Court-Judgments-Quickly-and-Efficiently.
4. B. Shukla, S. Gupta, A.K. Yadav, and D. Yadav. Challenges and issues in legal documents classification. *AIP Conference Proceedings* 2754(1): 020001 (2023).
5. J. Li, G. Zhang, L. Yu, and T. Meng. Research and design on cognitive computing framework for predicting judicial decisions. *Journal of Signal Processing Systems* 91(10): 1159-1167 (2019).
6. G. Sukanya and J. Priyadarshini. A meta-analysis of attention models on legal judgment prediction system. *International Journal of Advanced Computer Science and Applications* 12(2): 531-538 (2021).
7. A.S. Kwak, C. Jeong, J.W. Lim, and B. Min. A Korean Legal Judgment Prediction Dataset for Insurance Disputes. *arXiv preprint arXiv* 2401: 14654 (2024).
8. A.J. Rawat, S. Ghildiyal, and A.K. Dixit. Topic Modeling Techniques for Document Clustering and Analysis of Judicial Judgements. *International Journal of Engineering Trends and Technology* 70(11): 163-169 (2022).
9. A. Barman, D. Roy, D. Paul, I. Dutta, S.K. Guha, S. Karmakar, and S.K. Naskar. Convolutional Neural Networks can achieve binary bail judgement classification. *arXiv preprint arXiv* 2401: 14135 (2024).
10. K. Shirsat, A. Keni, P. Chavan, and M. Gosavi. Legal judgement prediction system. *International Research Journal of Engineering and Technology* 8(5): 734-738 (2021).
11. R.A. Shaikh, T.P. Sahu, and V. Anand. Predicting outcomes of legal cases based on legal factors using classifiers. *Procedia Computer Science* 167: 2393-2402 (2020).
12. V. Malik, R. Sanjay, S.K. Nigam, K. Ghosh, S.K. Guha, A. Bhattacharya, and A. Modi. ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. *arXiv preprint arXiv* 2105: 13562 (2021).
13. N. Sivaranjani, V.K.S. Ragavan, and G. Jawaherlalnehru. A Neural Network model for Indian Supreme Court legal judgment prediction. *Journal of Intelligent & Fuzzy Systems* 1: 235936 (2024).

14. S.S. Megala. Classification of Legal Judgement Summary using Conditional Random Field Algorithm. *International Journal of Computer Sciences and Engineering* 6(5): 23-33 (2018).
15. R. Sil and A. Roy. A novel approach on argument based legal prediction model using machine learning. *IEEE International Conference on Smart Electronics and Communication, (10-12 September 2020), Trichy, India* pp. 487-490 (2020).
16. P. Bhilare, N. Parab, N. Soni, and B. Thakur. Predicting outcome of judicial cases and analysis using machine learning. *International Research Journal of Engineering and Technology* 6(3): 326-330 (2019).
17. K. Javed and J. Li. Artificial intelligence in judicial adjudication: Semantic biasness classification and identification in legal judgement. *Heliyon* 10(9): e30184 (2024).
18. A. Fan, S. Wang, and Y. Wang. Legal Document Similarity Matching Based on Ensemble Learning. *IEEE Access* 12: 33910-33922 (2024).
19. Z. Yang, J. Ge, T. Hu, W. Yu, Y. Zheng, and Y. Dong. Text Classification of Judgement Documents Considering Sample Imbalance. *17th IEEE Conference on Industrial Electronics and Applications (16-19 December 2022)*, *Chengdu, China* pp. 1459-1462 (2022).
20. Y. Wang, J. Gao, and J. Chen. Deep learning algorithm for judicial judgment prediction based on BERT. *5th IEEE International Conference on Computing, Communication and Security (14-16 October 2020)*, Patna, India pp. 1-6 (2020).
21. B. Chen, Y. Li, S. Zhang, H. Lian, and T. He. A deep learning method for judicial decision support. *19th IEEE International Conference on Software Quality, Reliability and Security Companion (22-26 July 2019)*, *Sofia, Bulgaria* pp. 145-149 (2019).
22. S. Abbara, M. Hafez, A. Kazzaz, A. Alhothali, and A. Alsolami. ALJP: An Arabic Legal Judgment Prediction in Personal Status Cases Using Machine Learning Models. *arXiv preprint arXiv* 2309: 00238 (2023).
23. B. Strickson and B.D.L. Iglesia. Legal judgement prediction for UK courts. *3rd International Conference on Information Science and Systems (19 - 22 March, 2020)*, *Cambridge, United Kingdom* pp. 204-209 (2020).
24. A. Shelar, M. Moharir, V. Nama, J. Jaikumar, V. Patil, and A. Edupuganti. SmartLawAnnotator: A Machine Learning-Based Annotation Recommendation for Judgment Prediction. In: International Conference on Advances in Data-driven Computing and Intelligent Systems. *Springer, Singapore* pp. 415-430 (2023).
25. F. de Arriba-Pérez, S. García-Méndez, F.J. González-Castaño, and J. González-González. Explainable machine learning multi-label classification of Spanish legal judgements. *Journal of King Saud University-Computer and Information Sciences* 34(10): 10180-10192 (2022).
26. J. Zahir. Prediction of court decision from a rabic documents using deep learning. *Expert Systems* 40(6): e13236 (2023).
27. M. Medvedeva, A. Üstün, X. Xu, M. Vols, and M. Wieling. Automatic judgement forecasting for pending applications of the European Court of Human Rights. *Proceedings of the Fifth Workshop on Automatec Semantic Analysis of Information in Legal Text*. *CEUR Workshop Proceedings* pp. 12-23 (2021).
28. C. O'Sullivan and J. Beel. Predicting the outcome of judicial decisions made by the european court of human rights. *arXiv preprint arXiv* 1912: 10819 (2019).
29. M. Medvedeva, M. Vols, and M. Wieling. Judicial decisions of the European Court of Human Rights: Looking into the crystal ball. *13th Annual Conference on empirical legal studies (9-10 November, 2018), University of Michigan Law School, USA* pp. 24 (2018).
30. M. Medvedeva, M. Vols, and M. Wieling. Using machine learning to predict decisions of the European Court of Human Rights. *Artificial Intelligence and Law* 28(2): 237-266 (2020).
31. X. Zheng, Y. Liu, J. Jiang, L.M. Thomas, and N. Su. Predicting the litigation outcome of PPP project disputes between public authority and private partner using an ensemble model. *Journal of Business Economics and Management* 22(2): 320-345 (2021).
32. Y. Fang, X. Tian, H. Wu, S. Gu, Z. Wang, F. Wang, and Y. Weng. Few-shot learning for Chinese legal controversial issues classification. *IEEE Access* 8: 75022-75034 (2020).
33. L. Liu, D. An, Y. Wang, X. Ma, and C. Jiang. Research on legal judgment prediction based on Bert and LSTM-CNN fusion model. *3rd IEEE world symposium on artificial intelligence (18-20 June 2021)*, *Guangzhou, China* pp. 41-45 (2021).
34. Y. Liu and Y. Chen. A two-phase sentiment analysis approach for judgement prediction. *Journal of Information Science* 44(5): 594-607 (2018).
35. https://www.pakistanlawsite.com/Login/MainPage.
36. M. Omar, S. Choi, D. Nyang, and D. Mohaisen.

Robust natural language processing: Recent advances, challenges, and future directions. *IEEE Access* 10: 86038-86056 (2022).

37. T.N. Fitria. QuillBot as an online tool: Students' alternative in paraphrasing and rewriting of English writing. *Englisia: Journal of Language, Education, and Humanities* 9(1): 183-196 (2021).
38. K. Clark, U. Khandelwal, O. Levy, and C.D. Manning. What Does Bert Look At? An Analysis of Bert's Attention. *arXiv preprint arXiv* 1906: 04341 (2019).
39. S. Sarica and J. Luo. Stopwords in technical language processing. *Plos One* 16(8): e0254937 (2021).
40. R.C. Staudemeyer and E.R. Morris. Understanding LSTM-a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv* 1909: 09586 (2019).
41. G. Sukanya and J. Priyadarshini. Analysis on word embedding and classifier models in legal analytics. *AIP Conference Proceedings* 2802(1): 140001 (2024).
42. D.W. Otter, J.R. Medina, and J.K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on Neural Networks and Learning Systems* 32(2): 604-624 (2020).