IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.19, No.3, July 2025, pp. 305~316
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.108377

Court Decision Prediction Model Using Natural Language Processing and Random Forest

Nasution[1], Suprapto[*2]
[1,2]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: **[1]nasution@mail.ugm.ac.id**, *[2]suprapto@mail.ugm.ac.id

Abstrak

Meningkatnya jumlah perkara pidana di Indonesia yang mencapai 288.472 pada tahun 2023 atau naik 15% dari tahun sebelumnya, menimbulkan beban kerja yang besar bagi aparat peradilan. Kondisi ini mendorong kebutuhan akan sistem pendukung keputusan berbasis kecerdasan buatan untuk mempercepat dan meningkatkan kualitas pengambilan keputusan hukum. Penelitian ini mengusulkan model prediksi putusan pengadilan menggunakan pendekatan Random Forest yang dikombinasikan dengan teknik Natural Language Processing (NLP). Data yang digunakan adalah 21.630 dokumen putusan dari Mahkamah Agung Republik Indonesia dalam format PDF yang telah dikonversi menjadi XML. Proses penelitian mencakup tahap praproses teks, pembentukan fitur menggunakan Word2Vec dan FastText, serta klasifikasi dengan Random Forest. Berbeda dengan penelitian sebelumnya yang menggunakan metode LSTM, BiLSTM, dan CNN dengan akurasi 49,14%–77,32%, pendekatan ini menunjukkan hasil yang lebih optimal. Hasil eksperimen menunjukkan bahwa model yang diusulkan mampu mencapai akurasi prediksi hingga 63%-81%. Temuan ini menunjukkan potensi signifikan penggunaan NLP dan Random Forest dalam pengembangan sistem prediksi berbasis dokumen hukum berbahasa Indonesia.

Kata kunci— *Prediksi keputusan pengadilan, Natural Language Processing, Random Forest, Machine Learning*

Abstract

The increasing number of criminal cases in Indonesia, which reached 288,472 in 2023, or rose by 15% from the previous year, has created a substantial workload for judicial professionals. This situation highlights the urgent need for artificial intelligence–based decision support systems to accelerate and improve the quality of legal decision-making. This study proposes a court decision prediction approach using the Random Forest algorithm combined with Natural Language Processing (NLP) techniques. The dataset consists of 21,630 court decisions from the Supreme Court of Indonesia, originally in PDF format and converted into XML. The research procedure includes text preprocessing, feature construction using Word2Vec and Fast Text, and Random Forest classification. Unlike previous studies employing LSTM, BiLSTM, and CNN methods with accuracy ranging from 49.14% to 77.32%, the proposed approach delivers better performance. Experimental results show that the model achieves a prediction accuracy of up to 63%-81% for Penalty Categories classification and up to 65%-80% for long punishment regression. These findings demonstrate the significant potential of applying NLP and Random Forest to develop predictive systems in Indonesian legal document analysis.

Keywords— *Court decision prediction, Natural Language Processing, Random Forest, Machine Learning*

Received June 25th,2025; Revised July 27th, 2025; Accepted July 31th, 2025

1. INTRODUCTION

Indonesia's judiciary is central in upholding justice, primarily through judge-issued sentencing decisions. These decisions must be grounded in legal evidence and statutory law. In 2023, criminal cases rose by 15% to 288,472, increasing the burden on judicial personnel to process and assess large volumes of court decisions efficiently. [1].

This condition underscores the urgency of developing artificial intelligence (AI)-based systems to support legal decision-making processes. Repetitive and pattern-based tasks can be automated using AI technologies, accelerating analysis and enhancing decision accuracy. Several prior studies have proposed the use of deep learning classification models such as Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Convolutional Neural Networks (CNN) with attention mechanisms to predict punishment categories or sentence durations. However, the predictive accuracy of these models remains relatively low, ranging between 49.14% and 77.32% [2].

Similar approaches have been examined in international contexts. Shaikh et al. [3] employed Logistic Regression to predict legal case outcomes with an accuracy of up to 92%. Noguti et al. [4] used Word2Vec and Recurrent Neural Networks (RNN) to classify legal texts from 17,740 documents, achieving 90% accuracy. Strickson and Iglesia [5] developed a Legal Judgment Prediction (LJP) model in the United Kingdom, reporting a maximum accuracy of 69.05%. Meanwhile, Malik et al. [6] and Mumcuoğlu et al. [7] developed predictions of decisions in India and Türkiye using a combination of NLP and Deep Learning with accuracy up to 93%. Mustari et al. [8]and Anantathanavit et al. [9] implemented similar models for the Supreme Courts of Bangladesh and Thailand with competitive results in the Asia Pacific region, and Abbara et al.[10] developed a system that uses deep learning and NLP to predict verdict results in Arabic with an accuracy of 88%. Recent work by Ansari et al. [11] demonstrated that combining the Random Forest algorithm with TF-IDF and Word2Vec achieved an accuracy of 95% in predicting legal verdicts from English-language documents. These studies collectively indicate that integrating Natural Language Processing (NLP) and machine learning algorithms offers highly accurate and relevant predictions in modern legal systems.

Nonetheless, research employing such approaches for Indonesian-language legal texts remains limited. For instance, Nuranti et al. [2] applied LSTM, BiLSTM, and CNN with attention mechanisms to court decision documents from the Supreme Court of Indonesia, achieving a maximum accuracy of only 77.32%. Moreover, existing approaches have not explicitly demonstrated their effectiveness in capturing the structure and linguistic style unique to Indonesian legal discourse.

Given this background, this study proposes an alternative approach that leverages the Random Forest algorithm combined with NLP techniques to enhance the predictive accuracy of court verdicts. This combination is expected to offer a more effective solution to address Indonesia's high volume of legal data.

The objectives of this research are to: (1) develop a predictive model for court verdicts in Indonesia by integrating Random Forest and NLP to improve accuracy over previous approaches, and (2) evaluate the effectiveness of various text-based feature representation techniques including TF-IDF, Word2Vec, and FastText in the context of Indonesian legal documents. The dataset used in this study was obtained from the Supreme Court of the Republic of Indonesia and comprises 21,630 court decision documents in PDF format that have been converted into structured XML.

This study's findings will make a tangible contribution to the development of decision-support systems in the legal domain and improve the efficiency and transparency of the judicial system in Indonesia.

2. METHODS

2.1 Research Workflow

This study followed systematic stages to develop a court verdict prediction model, starting from data collection, text preprocessing, feature extraction using NLP, model building, and performance evaluation, as shown in Figure 1.

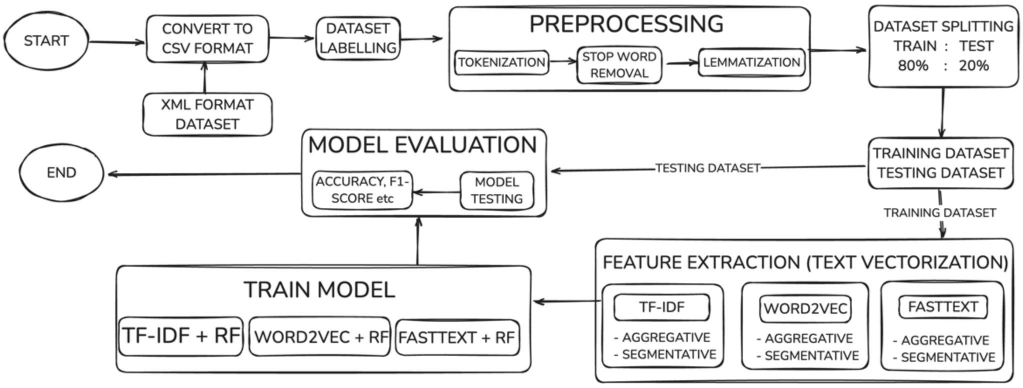

Figure 1. Research Workflow

The dataset comprises 21,630 Indonesian Supreme Court criminal verdicts in XML format. Sentencing durations were standardized into days and classified into four categories. Texts underwent preprocessing and were vectorized using TF-IDF, Word2Vec, and FastText with aggregative and segmentative approaches. Random Forest models were trained and evaluated using classification (accuracy, F1) and regression (R^2, MAE, MSE) metrics.

2.2 Dataset and Preprocessing

This study used 21,630 Indonesian criminal verdicts from the Supreme Court, converted from PDF to XML for structured extraction. Sentencing durations were extracted and converted into days for classification and regression tasks. Texts were preprocessed (tokenization, cleaning, stemming) and transformed into features using TF-IDF, Word2Vec, and FastText.

2.3 Feature Extraction

Preprocessed texts were vectorized using TF-IDF, Word2Vec, and FastText with two approaches: aggregative (combining all sections) and segmental (processing sections separately and concatenating vectors). The segmental approach preserves structural and semantic nuances of legal documents.

2. 3.1 TF-IDF (Term Frequency–Inverse Document Frequency)

Figure 2 illustrates the TF-IDF process, where raw text is tokenized and transformed into vectors based on term frequency and inverse document frequency to capture term importance and distinctiveness.

Figure 2. TF-IDF Architecture

1) *Term Frequency (TF)*

Term Frequency measures how frequently a term appears in a document. In the TF-IDF formulas, several symbols are used to represent term statistics. The symbol t refers to a specific term or word, while d the symbol represents a document. The notation ft,d indicates the frequency of a term $t's\ frequency$ within a document d. The denominator in the TF formula

$\sum t' \in dft', d$ sums the frequencies of all terms in the document d, providing normalization. The formula is shown in equation (1)

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (1)$$

2) *Inverse Document Frequency (IDF)*

Inverse Document Frequency quantifies the importance of a term across the entire corpus. It penalizes standard terms and highlights more unique or informative terms. The IDF formula D denotes the whole set of documents (corpus) and N is the total number of records. The expression $|\{d \in D: t \in d\}|$ counts how many documents contain the term t. The logarithmic function helps reduce the weight of standard terms across the corpus. The formula is shown in equation (2).

$$IDF(t,D) = \log\left(\frac{N}{|\{d \in D: t \in d\}|}\right) \quad (2)$$

3) TF-IDF

The final TF-IDF score is the product of TF and IDF, providing a weighted representation of each term. The formula is shown in equation (3)

$$TF - IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

2.3.2 Word2Vec

Word2Vec learns distributed word representations using two main architectures: CBOW and Skip-Gram, as illustrated in Figure 3.

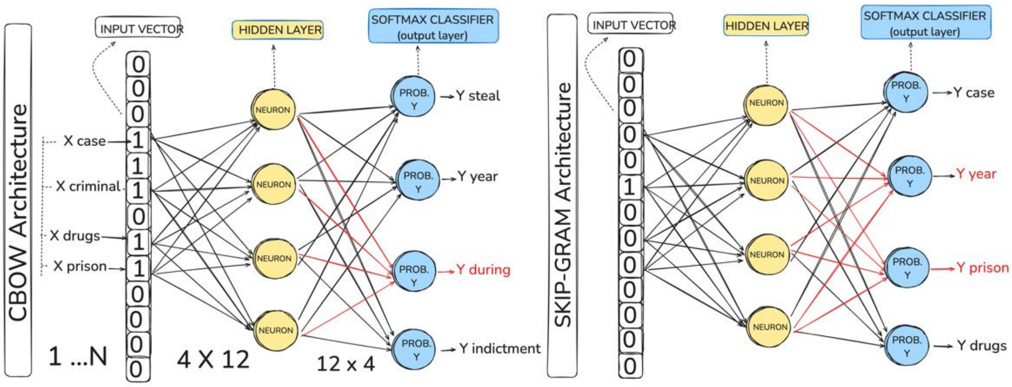

Figure 3. Architecture CBOW and SKIP-GRAM (Word2vec, Fast Text)

Word2Vec represents words using dense vectors through two main models: Continuous Bag-of-Words (CBOW) and Skip-Gram, as illustrated in *Figure 3*. The CBOW model predicts a target word w_t based on its surrounding context words within a window n. It aggregates the embeddings of the context words and optimizes the probability of correctly predicting the center word. The formula is shown in equation (4).

$$J_\theta^{CBOW} = \frac{1}{T}\sum_{t=1}^{T} \log p\,(w_t \mid w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n}) \quad (4)$$

Here, T the total number of tokens, w_t is the target word at a position t, and the surrounding words within the context window n serve as inputs. The probability $p(wt \mid \cdot)$ is calculated using a SoftMax function over the combined context embeddings. The model parameters θ are optimized to maximize this likelihood across all positions. In contrast, the Skip-Gram model uses the center word w_t to predict each context word $w_{(t+j)}$ $j \in [-n, n], j \neq 0$. It is particularly effective for learning representations of rare words. The formula is shown in equation (5).

$$J_\theta^{SG} = \frac{1}{T}\sum_{t=1}^{T} \sum_{-n \le j \le n j \neq 0} \log p\left(w_{t+j} \mid w_t\right) \tag{5}$$

2.3.3 FastText

FastText adopts the same training objective as the Skip-Gram model but introduces a key difference in input representation. Instead of learning a single embedding per word like in Word2Vec, FastText represents each word w_t as the sum of its character-level n-gram embeddings $z(w\_t)$. This allows the model to capture subword information and genera generalize rare or unseen words. The formula is shown in equation (6).

$$J_\theta^{FT} = \frac{1}{T}\sum_{t=1}^{T} \sum_{-n \le j \le n j \neq 0} \log p\left(w_{t+j} \mid z(w_t)\right) \tag{6}$$

2.4 Classification Model Architecture of Random Forest

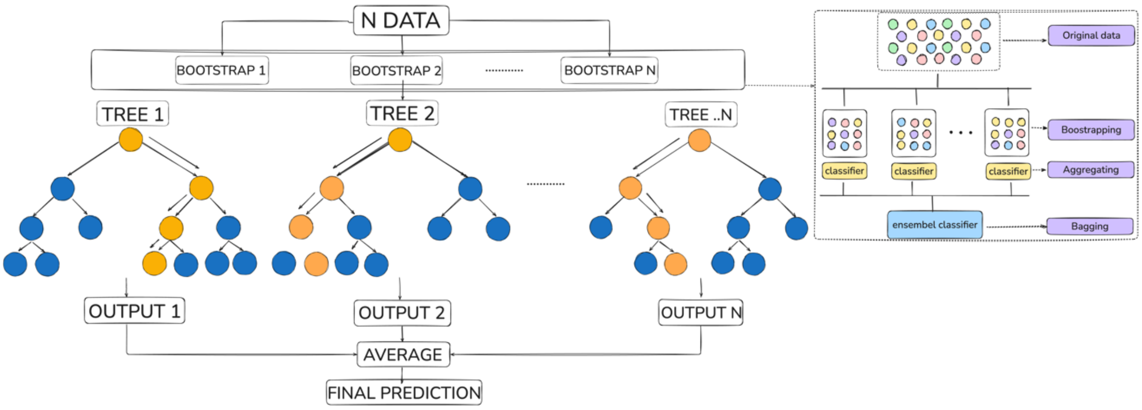

Figure 4. Random Forest Architecture

As illustrated in Figure 4, Random Forest is an ensemble method that builds multiple decision trees using bootstrapped data and random feature selection. It enhances generalization and reduces overfitting for both classification and regression.

3. RESULTS AND DISCUSSION

3.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to examine legal court texts' characteristics, quality, and distribution, providing insights that inform model development.

3.1.1 Descriptive Statistics and Preprocessing Efficiency

Thirteen primary text columns were preprocessed in parallel using 10 CPU cores. Narrative-heavy columns like *FaktaKasus* required more time. Overall, preprocessing reduced text length by over 30%, effectively removing non-informative content. The result is shown in Table 1.

Table 1. Average text length before and after preprocessing on the key column

Column	Original Length	After Preprocessing	Reduction (%)
Fakta Kasus	33,785	22,703	32.8%
Riwayat Dakwaan	7,248	5,271	27.3%
Pertimbangan Hukum	11,456	7,296	36.3%

3.1.2 Distribution of Punishment Categories

The punishment category distribution is relatively balanced, with *Sangat Ber*at slightly dominant. Stratified sampling and macro F1-score were used to address class proportions. Most documents are under 10,000 tokens, but *Ringan* cases tend to be longer, indicating a potential length bias in minor case narratives.

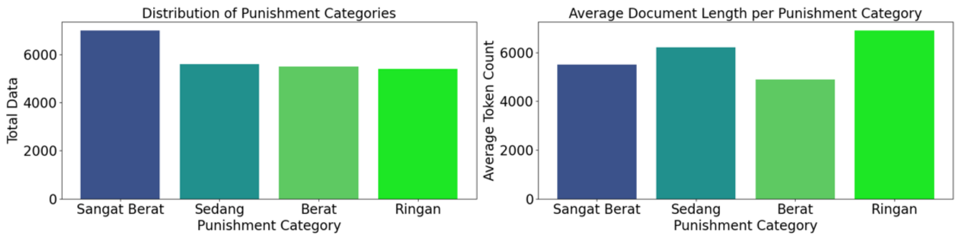

Figure 5. Distribution of Punishment Categories

3.1.3 Word Frequency and Dominant Vocabulary

Word frequency analysis confirms that preprocessing retained relevant terms such as *"dakwa", "saksi", "barang", "pidana",* and *"bukti"*. Category-specific terms also emerged *"narkotika"* and *"sabu"* in *Berat*, and *"uang"* and *"barang"* in Ringan.

3.1.4 Token Statistics, Rare Words, and Vocabulary Richness

The dataset has 99.7% repeated tokens and only 0.3% unique words, with 14% appearing just once. This favors models like FastText and TF-IDF, though careful tuning of min_count and min_df is needed to handle rare terms.

3.2 Classification Results (Punishment Category Prediction)

This section reports classification results using TF-IDF, Word2Vec, and FastText with Random Forest, tested under aggregative and segmentation strategies. Performance was evaluated using Accuracy and Macro F1-score to ensure balanced assessment across categories.

3.2.1 Aggregative Representation

The aggregative approach combines all document segments into a single text column, treating the entire document as a unified input for feature representation.

1) TF-IDF Aggregative Results

The TF-IDF model outperformed others, achieving 80.91% accuracy and a 0.78 macro F1-score using optimized n-grams (1–5), min_df=0.01, max_df=0.95, and Chi-Square feature selection.

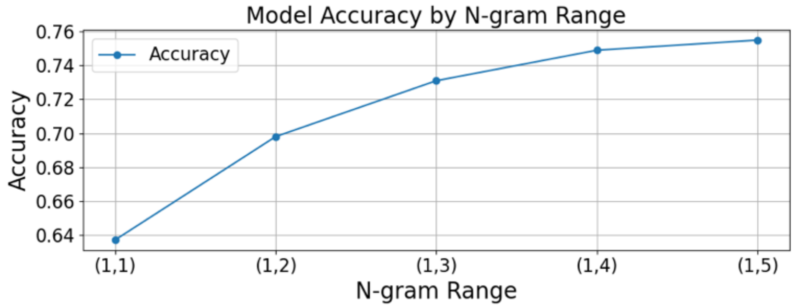

Figure 6. TF-IDF Classification Performance by N-Gram Range

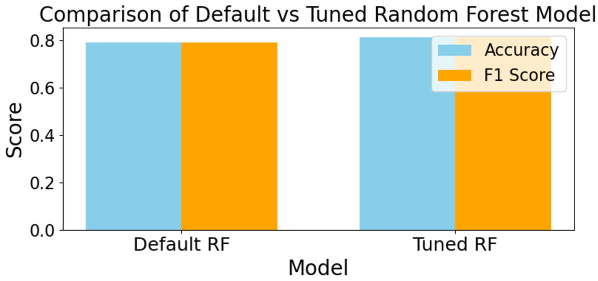

Figure 7. Final result of TF-IDF after Random Forest hyperparameter tuning

These improvements reflect the importance of context in legal text classification. Longer n-grams improved discriminative power, especially with domain-specific terms like *"narkotika golong i bukan tanam."*

2) Word2Vec Aggregative Results

Experiments with Word2Vec showed moderate results, with a best accuracy of 62.53% and an F1 Score of 0.6079 using a vector size of 50, min_count=10, and 20 training epochs. CBOW outperformed Skip-Gram in both efficiency and classification results.

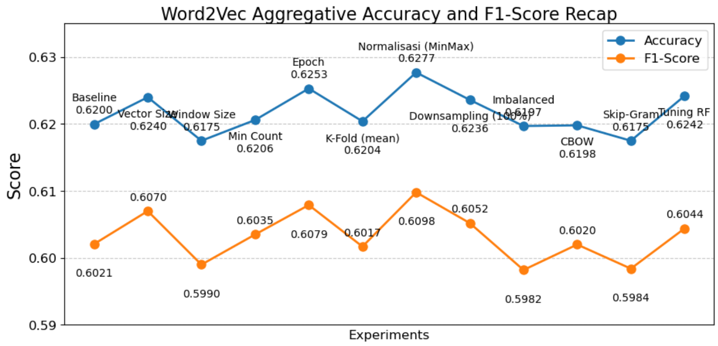

Figure 8. Aggregative Word2Vec Accuracy by Parameter Tuning

3) FastText Aggregative Results

As shown in Figure 9, FastText produced comparable results to Word2Vec, reaching its highest accuracy of 61.58% and F1-score of 0.6133 with vector size 100, window 10, min_count 5, and 20 epochs. Despite its subword modeling capabilities, FastText struggled to distinguish patterns in lengthy formal texts, likely due to the high occurrence of duplicate tokens.

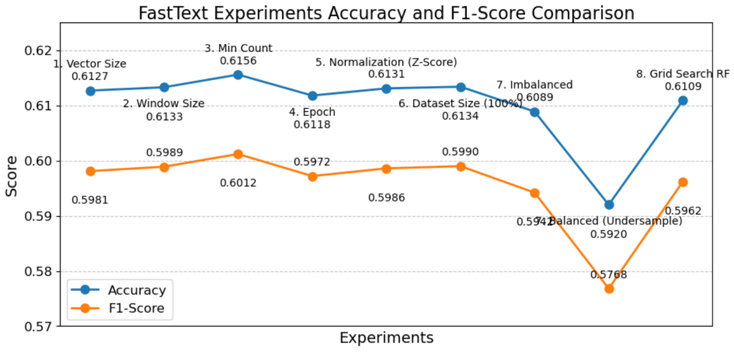

Figure 9. Aggregative FastText Accuracy by Parameter Tuning

3.2.2 Segmentative Representation

The segmentative approach decomposes legal documents into predefined segments (e.g., *FaktaKasus_Prep*, *PertimbanganHukum_Prep*, etc.). Each segment underwent individual vectorisation, and the resulting feature vectors were concatenated to preserve localised contextual meaning.

1) TF-IDF Segmentate Results

The TF-IDF model with segmentate representation achieved high performance, reaching 79.19% accuracy and 0.7900 F1-score. The best configuration used an n-gram range of (1,5), min_df=0.01, max_df=0.95, and feature selection using Chi-Square (k=16000). Although slightly below the aggregate setup, this configuration maintains classification stability and effectively exploits document structure.

2) Word2Vec Segmentative Results

With the segmental approach, Word2Vec's performance improved over its aggregative counterpart. The best configuration used vector size 200, window size 10, and min_count 10, combined with Z-Score normalization. The model reached 64.05% accuracy and 0.5826 F1-score, indicating that localized semantic preservation benefits distributional representations.

3) FastText Segmentative Results

FastText also showed performance gains under segmental processing. Using vector size 100, window size 5, and min_count 5, the model achieved 63.75% accuracy and 0.5769 F1-score. Compared to aggregate FastText, this setup captured intra-segment nuances more effectively, albeit still lagging behind TF-IDF and Word2Vec.

3.2.3 Comparison result of punishment classification

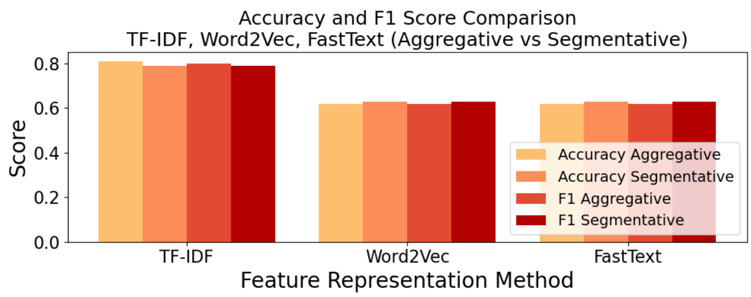

Figure 10. Comparison result of punishment classification

Figure 10's bar chart compares accuracy and F1 scores across TF-IDF, Word2Vec, and FastText methods under aggregative and segmentation frameworks. TF-IDF consistently outperforms others, achieving the highest values on both metrics. Word2Vec performs moderately, while FastText yields the lowest, especially under segmentation settings. Segmentative approaches tend to improve F1 slightly, but aggregative methods achieve better overall accuracy, emphasizing the strength of holistic text representation in classification tasks.

3.3 Regression Results (Punishment Duration Prediction)

This section presents the sentence duration prediction task results using TF-IDF, Word2Vec, and FastText under a segmentative framework. The target variable is the numeric duration of punishment in days. Evaluation was based on three metrics: R^2 (coefficient of determination), MSE (mean squared error), and MAE (mean absolute error).

3.3.1 Aggregative Representation

1) TF-IDF Aggregative Results

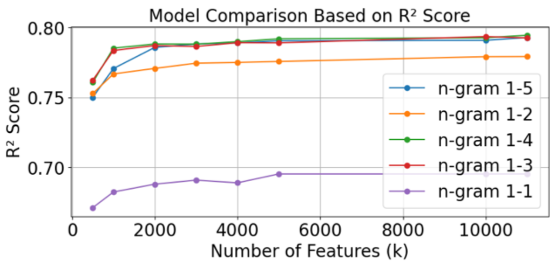

Figure 11. Comparison result using R2 by n-gram TF-IDF punishment prediction (aggregate)

Figure 11 displays the regression outcomes of the aggregative approach, where all text segments are combined before vectorization. Using n-gram (1,5) with 16,000 features, its best configuration achieved an R^2 of 0.7963, MSE of 207,520, and MAE of 245.69. The corresponding scatter plot for this optimal setup is illustrated in Figure 12, reflecting the model's ability to capture global linguistic patterns from the full document.

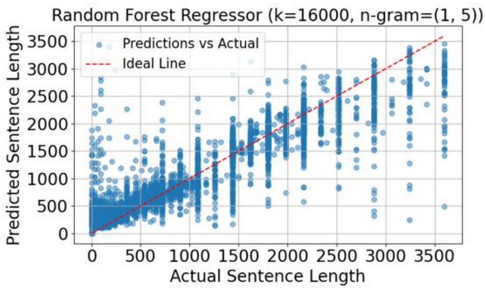

Figure 12. Scatter plot result for the best-performing TF-IDF aggregate

2) Word2Vec Aggregative Results

Under aggregative settings, Word2vec achieved its highest performance with tuned min count and epoch parameters, producing $R^2 = 0.519$, MSE = 488,500, and MAE = 440.8. This suggests that Word2Vec can be a viable regression feature with proper tuning.

Table 2. Word2Vec Aggregative Regression Summary

Configuration	R^2	MSE	MAE
Best (min count/epoch)	0.519	488500	440.8
Baseline	0.504	549117	454.9

3) FastText Aggregative Results

The best FastText configuration under aggregative mode (epoch tuning) resulted in R^2 = 0.5083, MSE = 500,929, and MAE = 450.04, showing modest improvements over the baseline.

Table 3. FastText Aggregative Regression Summary

Configuration	R^2	MSE	MAE
Best (epoch tuned)	0.5083	500929	450.04
Baseline	0.4808	574904	474.74

3.3.2 Segmentative Representation

1) TF-IDF Segmentative Results

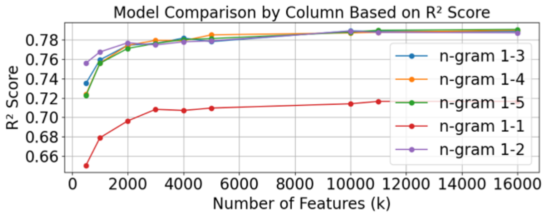

Figure 13. Comparison result R2, MSE, and MAE by n-gram TF-IDF punishment prediction (segmentate)

Figure 13 shows that the segmentation TF-IDF model achieved strong regression results, with its best setup (n-gram 1–5, 16,000 features) yielding an R^2 of 0.7904, MSE of 137,515, and MAE of 220.65, slightly below the aggregative counterpart. The corresponding scatter plot of predictions for this optimal model is presented in Figure 14.

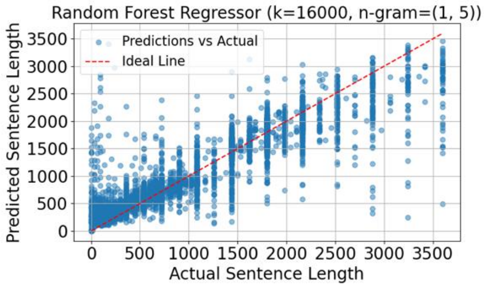

Figure. 14 Scatter plot result for the best-performing TF-IDF segment

2) Word2Vec Segmentative Results

Word2Vec with segmentative representation achieved its best result with tuning on vector size and epoch, yielding R^2 = 0.500, MSE = 513,336, and MAE = 451. Although behind TF-IDF, it demonstrates the potential of contextual embeddings.

Table 4. Word2Vec Segmentative Regression Summary

Configuration	R^2	MSE	MAE
Best (vector/epoch tuned)	0.500	513336	451
Baseline	0.450	590808	471

3) FastText Segmentative Results

FastText underperformed in segmentative regression. Its best configuration using tuned epochs achieved R^2 = 0.454, MSE = 604,000, and MAE = 475.

Table 5. FastText Segmentative Regression Summary

Configuration	R^2	MSE	MAE
Best (epoch tuned)	0.454	604000	475
Baseline	0.437	623043	487.6

3.3.3 Comparison result of punishment duration

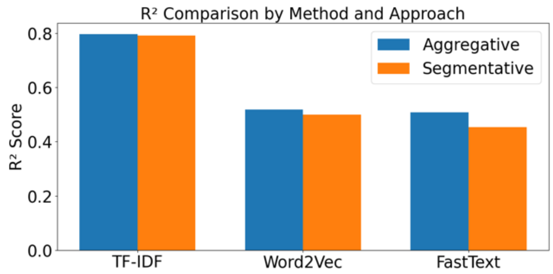

Figure 15. Comparison result of punishment prediction

Figure 15 shows that TF-IDF achieves the best regression performance, with the highest R^2 (0.7963) and lowest MAE (220.65). Aggregative representation generally explains more variance (R^2), while segmentation yields lower prediction errors (MAE). Word2Vec benefits moderately from tuning (R^2 up to 0.519), but FastText performs the weakest overall, with the lowest R^2 (0.454) and highest MSE (604000).

3.4 Feature Ablation and Segment Importance

To assess the relative contribution of each document segment, we conducted a feature ablation analysis by iteratively removing one segment at a time and observing the change in model performance. The evaluation was performed on classification (punishment category) and regression (punishment duration) tasks using the TF-IDF with a Random Forest model.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10	Model 11	Model 12	Model 13
KepalaPutusan_Prep	✔		✔	✔	✔	✔	✔	✔	✔	✔	✔		✔
IdentitasTerdakwa_Prep	✔	✔		✔	✔	✔	✔	✔	✔	✔	✔		✔
RiwayatPerkara_Prep	✔	✔	✔		✔	✔	✔	✔	✔	✔	✔		✔
RiwayatPenahanan_Prep	✔	✔	✔	✔		✔	✔	✔	✔	✔	✔		✔
RiwayatDakwaan_Prep	✔	✔	✔	✔	✔		✔	✔	✔	✔	✔		✔
RiwayatTuntutan_Prep	✔	✔	✔	✔	✔	✔		✔	✔	✔	✔		
FaktaKasus_Prep	✔	✔	✔	✔	✔	✔	✔		✔	✔	✔		
FaktaHukum_Prep	✔	✔	✔	✔	✔	✔	✔	✔		✔	✔	✔	
PertimbanganHukum_Prep	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔	✔	
PenutupDokumen_Prep	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔	
Accuracy (%)	79.28	79.45	79.23	79.5	79.58	80.18	68.23	79.28	79.43	79.3	79.25	79.58	62.06
Training Duration	00:00:26	00:00:26	00:00:26	00:00:26	00:00:26	00:00:21	00:00:26	00:00:22	00:00:25	00:00:24	00:00:25	00:00:20	00:00:22

Figure 16. The results of the TF-IDF Category of Punishment with Random Forest

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10	Model 11	Model 12	Model 13
KepalaPutusan_Prep	✔		✔	✔	✔	✔	✔	✔	✔	✔	✔		✔
IdentitasTerdakwa_Prep	✔	✔		✔	✔	✔	✔	✔	✔	✔	✔		✔
RiwayatPerkara_Prep	✔	✔	✔		✔	✔	✔	✔	✔	✔	✔		✔
RiwayatPenahanan_Prep	✔	✔	✔	✔		✔	✔	✔	✔	✔	✔		✔
RiwayatDakwaan_Prep	✔	✔	✔	✔	✔		✔	✔	✔	✔	✔		✔
RiwayatTuntutan_Prep	✔	✔	✔	✔	✔	✔		✔	✔	✔	✔		
FaktaKasus_Prep	✔	✔	✔	✔	✔	✔	✔		✔	✔	✔		
FaktaHukum_Prep	✔	✔	✔	✔	✔	✔	✔	✔		✔	✔	✔	
PertimbanganHukum_Prep	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔	✔	
PenutupDokumen_Prep	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔	
MSE	137515.1804	137411.6227	137260.9569	135713.6148	136407.3246	143714.5754	210847.1951	135552.7526	136275.9259	138698.7132	136776.0597	151504.3902	239888.5112
R^2	0.7904	0.7906	0.7908	0.7932	0.7921	0.781	0.6787	0.7934	0.7923	0.7886	0.7915	0.7691	0.6344
Training Time	00:17:47	00:17:50	00:16:59	00:17:02	00:18:44	00:16:57	00:17:47	00:16:29	00:18:23	00:17:55	00:17:23	00:14:00	00:13:55

Figure 17. The results of the TF-IDF Duration of Punishment with Random Forest

As shown in Figures 17 and 18 (classification) and Table 18 (regression), the *"Fakta Hukum"* (Legal Facts) and *"Pertimbangan Hakim"* (Judicial Considerations) segments were the most influential. Removing these components caused the most significant drops in accuracy and R^2 scores, indicating their central role in legal reasoning and sentencing decisions. On the other hand, segments like *"Identitas Terdakwa"* (Defendant Identity) and *"Kepala Putusan"* (The head of the verdict) showed minimal impact when excluded, suggesting they contribute less predictive information. These findings are visually supported in Figure 16, where performance declines sharply with the omission of core interpretative sections. This confirms that structural segmentation enhances model interpretability and helps isolate key informational sources within legal documents.

CONCLUSIONS

This study presents a predictive framework for sentencing outcomes in Indonesian legal texts using TF-IDF, Word2Vec, and FastText combined with Random Forest. Among all methods, TF-IDF emerged as the most reliable across both classification and regression tasks, achieving up to 81% accuracy for sentence category prediction and an R^2 of 0.80 for sentence duration. These results outperform previous LSTM/CNN-based models, which reached up to 77% accuracy. The findings reaffirm the strength of frequency-based representations like TF-IDF in formal domains and suggest future enhancements through transformers, hybrid embeddings, and domain-specific preprocessing for deeper semantic modelling.

REFERENCES

[1] F. S. Pratiwi, "Data Jumlah Kejahatan di Indonesia pada 2023," 2023. [Online]. Available: https://dataindonesia.id/varia/detail/data-jumlah-kejahatan-di-indonesia-pada-2023

[2] E. Q. Nuranti, E. Yulianti, and H. S. Husin, "Predicting the Category and the Length of Punishment in Indonesian Courts Based on Previous Court Decision Documents," *Computers*, vol. 11, no. 6, Jun. 2022, doi: 10.3390/computers11060088.

[3] R. A. Shaikh, T. P. Sahu, and V. Anand, "Predicting Outcomes of Legal Cases based on Legal Factors using Classifiers," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 2393–2402. doi: 10.1016/j.procs.2020.03.292.

[4] M. Y. Noguti, E. Vellasques, and L. S. Oliveira, "Legal Document Classification: An Application to Law Area Prediction of Petitions to Public Prosecution Service," Oct. 2020, doi: 10.1109/IJCNN48605.2020.9207211.

[5] B. Strickson and B. De La Iglesia, "Legal Judgement Prediction for UK Courts," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Mar. 2020, pp. 204–209. doi: 10.1145/3388176.3388183.

[6] V. Malik *et al.*, "ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation," May 2021, [Online]. Available: http://arxiv.org/abs/2105.13562

[7] E. Mumcuoğlu, C. E. Öztürk, H. M. Ozaktas, and A. Koç, "Natural language processing in law: Prediction of outcomes in the higher courts of Turkey," *Inf Process Manag*, vol. 58, no. 5, 2021, doi: 10.1016/j.ipm.2021.102684.

[8] N. Mustari, S. K. Sen, A. Banik, M. H. K. Mehedi, and A. A. Rasel, "Techniques to Estimate the Status of Legal Proceedings Considering Sequential Text Data," in *2023 International Conference on Emerging Smart Computing and Informatics, ESCI 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ESCI56872.2023.10099995.

[9] R. Anantathanavit, J. Chongthanakorn, W. Kongsantinart, P. Praiwattana, and T. Thaipisutikul, "Utilizing AI and Natural Language Processing to Predict Supreme Court Decisions in Thailand," in *KST 2024 - 16th International Conference on Knowledge and Smart Technology*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 45–50. doi: 10.1109/KST61284.2024.10499665.

[10] 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," Jul. 2023, [Online]. Available: http://arxiv.org/abs/2309.00238

[11] T. Ansari, H. S. Dhillon, and M. Singh, "Machine Learning Model to Predict Results of Law Cases," in *2024 4th International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies, ICAECT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICAECT60202.2024.10468789.