



AI-Driven Predictive Judicial Workload Management System for Intelligent Case Scheduling, Resource Allocation, and Court Performance Optimization

Ajita Mishra¹, Lakshmi Dubey², Pravishi Gupta³, Akanksha Verma⁴, Yusuf Perwej⁵

¹Scholar (B.Tech) Department of Computer Science & Engineering, Shri Ramswaroop Memorial University, Barabanki, Uttar Pradesh

²Scholar (B.Tech) Department of Computer Science & Engineering, Shri Ramswaroop Memorial University, Barabanki, Uttar Pradesh

³Scholar (B.Tech) Department of Computer Science & Engineering, Shri Ramswaroop Memorial University, Barabanki, Uttar Pradesh

⁴Scholar (B.Tech) Department of Computer Science & Engineering, Shri Ramswaroop Memorial University, Barabanki, Uttar Pradesh

⁵Professor, Department of Computer Science & Engineering, Shri Ramswaroop Memorial University, Barabanki, Uttar Pradesh

Abstract: District and taluka courts in India are struggling with a persistent backlog of cases, and they do not have a technique that is data-driven and methodical for determining the priority of cases or grouping things that are related. The purpose of this paper is to present an artificial intelligence-based judicial workload optimization system, which is internally referred to as Nyaya Setu. This full-stack application addresses three coupled sub-problems within the realm of judicial triage. These problems include automatically grouping cases that are similar, generating a transparent priority-ranked cause list, and load-balancing case assignment across three judges. After iterative feature engineering and log-transformation of the skewed target, the system achieves a mean absolute error of 668.5 days by combining a pretrained transformer (BART) for summarization, TF-IDF/K-Means clustering for grouping similar cases, and a LightGBM regression model trained on 3,179 real disposal-duration records to estimate case urgency. This results in the system achieving a mean absolute error of 668.5 days. The system was validated end-to-end on 854 real Supreme Court judgments (2023) with full text: clustering produced five legally interpretable groups (e.g., bail/criminal matters, land-acquisition disputes), and feature-importance analysis of the trained priority model confirmed that case type and state jointly dominate predicted disposal time, consistent with independently published regional disposal-time disparities. We give these findings, which include the significant absolute inaccuracy of the model, with explicit critical analysis rather than as a final solution. Additionally, we suggest clear future approaches, which include deeper per-case characteristics and constraint-based judge assignment.

Keywords: Judicial Case Management, Natural Language Processing, Text Summarization, Unsupervised Clustering, TF-IDF, Gradient Boosting, LightGBM, Workload Optimization

1. Introduction

The Indian judicial system carries one of the largest case backlogs in the world, with the National Judicial Data Grid (NJDG) reporting tens of millions of pending cases across district and subordinate courts [1]. This backlog directly affects undertrials awaiting bail decisions, civil litigants awaiting resolution of property or contractual disputes, and public confidence in timely justice delivery. A significant contributor to this backlog is the absence of systematic, data-driven case triage. In practice, the order in which pending cases are heard is often determined by filing-date order, informal judicial discretion, or administrative convenience, rather than an explicit, auditable model [2] of urgency. Similarly, cases addressing the same underlying legal question are rarely grouped or cross-referenced automatically, making it difficult for courts to identify patterns, apply consistent reasoning, or batch-process related matters. This paper describes the design, implementation, and evaluation of an [3] AI-Based Judicial Workload Optimization System (internally developed under the name Nyaya Setu), a prototype system that applies natural language processing and machine learning to three concrete aspects of this problem: automatic similar-case clustering, data-driven priority scoring, [4] and workload-balanced case assignment. Unlike purely theoretical proposals, this work includes a fully functional three-tier web application, real (not purely simulated) training and validation data, and an honest

quantitative evaluation of model performance, including its limitations.

1.1 Motivation

Even a modest, well-explained improvement in case triage has outsized real-world value in a judicial context, because every hour of judicial time misallocated compounds across millions of pending matters [5]. A system that is transparent about its confidence and limitations is more likely to be trusted and adopted by court administrators than an opaque model promising unrealistic accuracy.

1.2 Contributions

A complete, deployable three-tier web application (React frontend, FastAPI backend, SQLite persistence) for judicial case triage, with a fully working user interface covering case intake, AI triage, priority assignment, workload optimization, judge management, hearing [6] calendars, analytics, and notifications. Integration of a pretrained transformer-based summarization model and an unsupervised clustering pipeline for grouping similar cases, validated on genuine judgment text rather than synthetic examples. A supervised priority-scoring model (LightGBM) trained on real Indian court disposal-duration data, with an explicit rule-based fallback for reliability, and a documented three-iteration improvement process. A transparent, reproducible data pipeline that converts [7] four structurally different real-world datasets into a unified schema, with each dataset's provenance and limitations explicitly disclosed. Genuine clustering and feature-importance analysis (Sections VI-C, VI-D) that independently corroborate a documented real-world finding [8] regional disparity in case disposal speed lending external validity to the model's internal behaviour. A candid discussion of model error and its causes (Section VII), positioned as a contribution to the literature on applying ML to sparse, high-variance [9].

2. Background

In recent years, much emphasis has been paid to the use of artificial intelligence (AI) in judicial decision making. The use of artificial intelligence (AI) [10] in judicial decision making is an important field of study in legal informatics. In particular, recent advances in machine learning (ML) models have made it possible to handle large legal datasets and to anticipate the outcomes of cases. AI-powered solutions increase speed, uniformity and impartiality in judicial procedures, by lowering human burden and limiting mental biases in evaluating evidence. The first application of neural network-based classification systems to economic justice cases was proposed by Alekseev et al. [11]. They gave the proof of the AI potential for the improvement of legal processes via the automation, sorting and categorization of legal documents. The researchers used the model to identify relevant legal categories, resulting in enhanced document retrieval and decreased administrative load for legal experts. Zhang and Sun [12] proposed a neural network model for predicting sentencing results, which helps to ensure the consistency and impartiality of criminal sentencing. Their research examined the topic of discrepancies in sentencing, which is generally linked to the viewpoints of individual judges. We reviewed historical case data and sentence records and made suggestions to help inform fair sentencing choices. Artificial intelligence-driven frameworks

are used as benchmarks for the exploration of judicial reasoning, which introduces another level of responsibility and ensures legal judgments are based on existing precedents. Nonetheless, problems about openness and interpretability are shown to continue. Legal academics and AI researchers have highlighted the ethical issues that “black-box” models, such as deep neural networks, pose because of the little transparency they give about their decision-making processes [13]. While the efficiency of AI-generated suggestions in the treatment of huge data sets is established, the opacity of the reasoning process presents issues of judicial confidence and acceptance. AI reasoning opacity is seen as especially troublesome in legal decision making, where accountability and openness are seen as crucial. In this respect, the increasing momentum of explainable AI (XAI) is recognized, since it offers methods for judges to comprehend and assess AI-assisted suggestions. XAI techniques emphasize the important evidence that supports AI predictions, hence improving judicial supervision and confidence in AI-assisted legal decision-making. Previous work on AI applications in the judicial area may be generally grouped into three categories: (a) retrieval and summary of legal documents, (b) prediction of case outcomes or length, and (c) optimization of court administration and scheduling [14]. Extractive methods that rank and choose existing phrases and abstractive methods that produce new condensed text using pretrained transformer models such as BART and T5 have been studied for the summary of legal documents. In this project we follow the abstractive technique for better readability, at the expense [15] of more computing per text than the extractive approach. Text clustering has been traditionally performed in the fields of legal and general NLP using conventional vector-space representations such as TF-IDF and K-Means [16][17] or hierarchical clustering, and more recently dense neural sentence embeddings (e.g., Sentence-BERT). Both families of approach share the same unsupervised learning concept of dividing a corpus by geometric closeness in a learnt or manufactured feature space, without human supplied labels. The key approach used in this research for the experimental validation (Section V-D) is TF-IDF based [18] clustering, however neural embeddings are a natural upgrade once the infrastructure is in place. The primary environment in which case-duration and result prediction has been investigated is that of the U.S. and European court systems, where structured docket data is more easily accessible in bulk. Similar Indian-context work is rather limited in the public literature [19], in part because detailed per-case procedural data (e.g., adjournment history, particular judge assignment) is not released in bulk form by Indian courts, unlike aggregate pendency information supplied via the NJDG. This shortage was a primary motivation for the multi-source data-engineering strategy outlined in Section V-A [20].

3. Problem Statement

Given a corpus of pending court cases, each described by attributes such as case type, procedural stage, filing date, and free-text case [21] details, the system must address three coupled sub-problems:

- **Similarity Grouping:** partition the case corpus such that cases addressing similar subject matter are grouped together, without relying on manually curated categories.
- **Priority Estimation:** assign each pending case a scalar urgency score reflecting, as closely as available data

permits, how quickly that case should be brought to hearing relative to others.

- **Workload Distribution:** given a ranked list of cases and a set of judges with finite capacity, assign cases to judges such that higher-priority cases are scheduled sooner and no judge is disproportionately overloaded.

A fourth, cross-cutting requirement is transparency: because system outputs directly influence judicial administration, every score and assignment must be traceable to an explainable mechanism an explicit rule or a model whose feature importances can be inspected rather than an opaque decision.

4. Proposed System Architecture

The system follows a standard three-tier web architecture with an additional machine-learning engine layer embedded within

the backend service, illustrated in Fig. 1. The frontend is implemented in React with Tailwind CSS, communicating with the backend exclusively via a REST API. The backend, built with FastAPI (Python), exposes endpoints for case ingestion, clustering, summarization, priority scoring, judge management, and workload assignment. SQLAlchemy provides the object-relational mapping layer [22] over a SQLite database, chosen for zero-configuration deployment during prototyping; the codebase is structured so that migrating to PostgreSQL for production requires only a connection-string change, since no raw SQL is used. The ML [23] Engine is logically separated into three independent modules summarization, clustering, and priority scoring each exposed through a stable function-level interface. This separation allows any individual module to be upgraded, for example replacing the rule-based priority [24] formula with the trained LightGBM model, without requiring changes to the API layer or frontend.

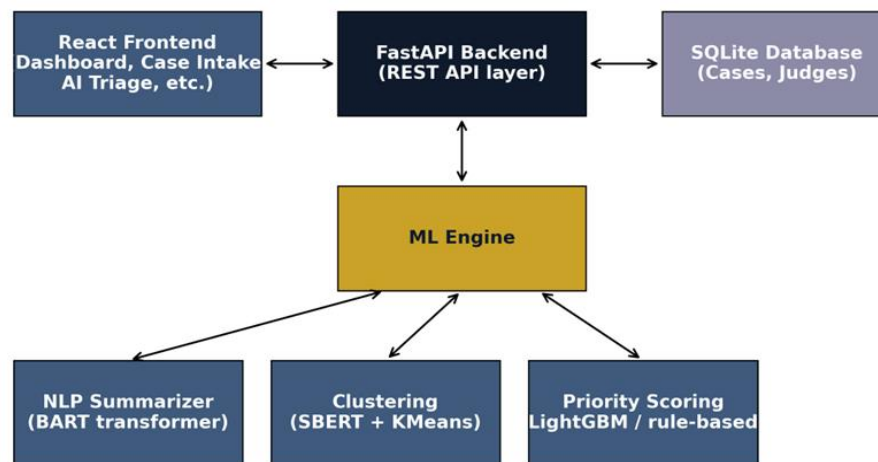


Fig. 1. High-Level System Architecture of the AI-Based Judicial Workload Optimization System.

The architecture in Fig. 1 is intentionally layered to keep concerns separated and independently replaceable. At the top, the React frontend is a pure presentation layer: it holds no business logic and renders whatever the backend returns, which keeps the user interface simple to modify without touching prediction logic. All communication crosses a single, well-defined REST boundary, meaning the backend could in principle serve a mobile application or a different frontend framework without modification. The FastAPI backend acts as the orchestration layer, validating incoming requests, coordinating calls to the database and the ML Engine, and returning structured JSON responses. The SQLite database is treated purely as a persistence layer accessed only through the SQLAlchemy ORM, so the storage engine itself is an implementation detail that can be swapped later. The ML Engine is the system's core intelligence layer and is deliberately kept stateless and modular: each of its three components (summarizer, clustering module, priority scorer) exposes a narrow function-level interface and can be retrained, replaced, or disabled independently, which is precisely what allowed the priority-scoring component to evolve from a rule-based formula to a trained LightGBM model [25] during development without any change to the API or frontend layers above it. This layered separation of presentation, orchestration, persistence, and intelligence is a standard software-engineering pattern that, in this project, additionally serves the goal of transparency: because each layer [26] has one clear responsibility, it is

straightforward to trace any output back to the specific rule or model that produced it.

4.1 Proposed System Screenshots

Figs. 2 to 4 show the deployed application in operation with live data, demonstrating that the architecture described above runs as a working system rather than a theoretical design.



Fig. 2. Main Dashboard of the AI-Based Judicial Workload Optimization System.

The dashboard shown in Fig. 2 is the system's landing page and aggregates the four-core metrics an administrator needs at a

glance: total cases (43,766, reflecting the real Kaggle-derived corpus loaded into the system), triage-completed [27] count, high-priority case count, and active judges. The donut chart breaks down the case corpus by type (Civil 52%, Others 29%, Criminal 19%), directly reflecting the distribution reported in Fig. 6 of this paper. The Judicial Workload [28] Snapshot panel renders each judge's current load as a percentage gauge, color-coded from green (low load) through amber to red (very high load), giving an immediate visual signal of where rebalancing, via the Workload Optimization module, is most needed.

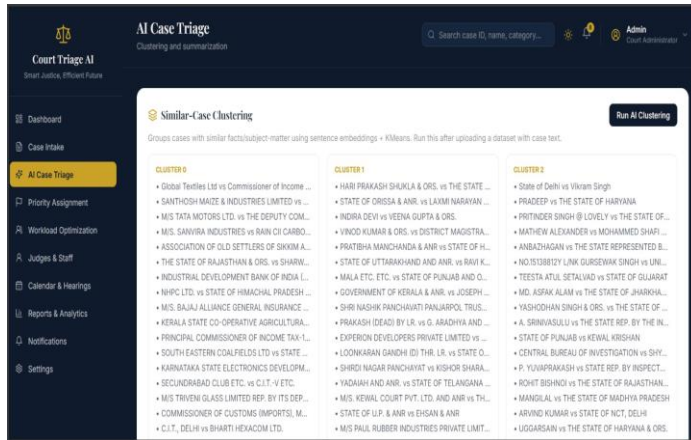


Fig. 3. AI Case Triage page showing similar-case clustering output.

Fig. 3 shows the AI Case Triage interface after the clustering pipeline (Section V-D) has been executed on the ingested corpus. Cases are grouped into numbered clusters and displayed as columns, each populated with real [29] case titles drawn from the underlying dataset. This view allows a court administrator to visually confirm that cases assigned to the same cluster plausibly share subject matter, without needing to inspect the underlying embedding or [30] vectorization mathematics directly, which is an important usability property for a system intended to be operated by non-technical judicial staff.

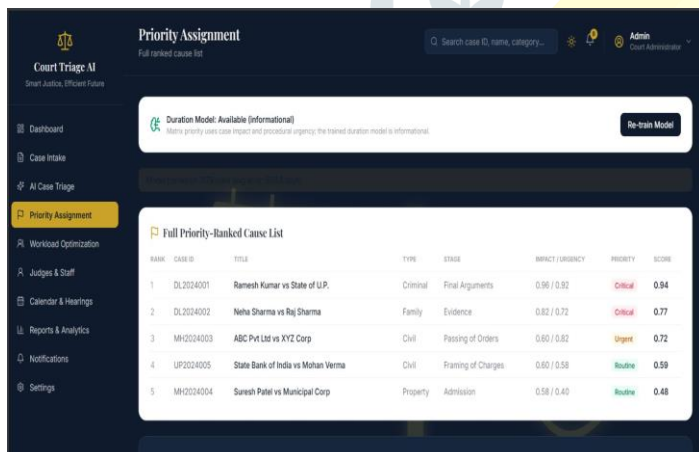


Fig. 4. Priority Assignment page showing the full ranked cause list.

Fig. 4 displays the system's priority-ranked cause list, the direct output of the scoring mechanism described in Section V-E. Each row shows a case's rank, [31] identifying details, computed impact/urgency sub-scores, a categorical priority label (Critical, Urgent, or Routine), and the final numeric score. The interface header additionally indicates whether the trained duration model is currently informing the ranking or whether the system is operating on the rule-based formula alone, directly

surfacing the transparency property discussed in Section III to the end user rather than hiding it [32].

5. Methodology

5.1 Data Collection and Sources

No single publicly available dataset contained all attributes required by the system: full judgment text, structured case metadata, and ground-truth disposal duration. Consequently, four complementary real-world data sources were used, each serving a distinct purpose, summarized in Table I and Fig. 5. The hand-curated sample (7 cases) provides a guaranteed, always-working end-to-end demonstration of every system feature regardless of external data availability. The Kaggle Supreme Court metadata (46,656 rows) supplies case-type, party, and date fields at realistic [33] scale for dashboard and analytics validation, though it lacks full judgment text. Real judgments with full text (854 cases) were obtained by downloading the year-2023 English-language archive from the AWS Open Data registry's Indian Supreme Court Judgments dataset, extracting text from the accompanying PDFs. The Disposal-Time Index (3,179 cleaned rows, after de-duplication) provides genuine filing-to-decision duration labels used to train the priority-scoring model. It is important to state explicitly that the Disposal-Time Index dataset is not an official government publication; it is an independent aggregation of public eCourts records compiled by an external legal-technology platform. This distinction is disclosed here for academic transparency and should be similarly disclosed in any downstream citation of this work.

5.2 Data Preprocessing Pipeline

Each data source required a distinct preprocessing script due to differing native schemas. The Kaggle metadata dataset provided clean tabular fields but lacked case-type labels, which were inferred heuristically from case-number prefixes (for example, the prefix "Cr.L.A." was mapped to Criminal, and "C.A." to Civil). The AWS-hosted judgment metadata, by contrast, stored structured fields inside a raw HTML fragment rather than clean JSON keys; a regular-expression-based parser was written to extract petitioner, respondent, decision date, and case number directly from this HTML [34]. Judgment text itself was extracted from the accompanying PDF files using the pdfplumber library, capped at approximately 6,000 characters per document to bound downstream model input size, with zero extraction failures across all 854 documents. All pipelines converge on a single normalized schema [35] case identifier, title, parties, court, case type, procedural stage, decision date, disposal status, and full text where available consumed by the application's ingestion API.

5.3 Case Summarization Module

Case summarization uses facebook/bart-large-cnn, a transformer-based sequence-to-sequence model pretrained for abstractive summarization. The model is used in inference mode only; no fine-tuning was performed in this phase, though domain-specific fine-tuning (e.g., via LoRA/QLoRA on Indian judgment text) is identified as future work. Given a case's full text, the model produces a condensed summary of approximately 30–130 tokens, stored alongside the case record and surfaced in the user interface.

5.4 Similar-Case Clustering Module

Clustering was experimentally validated using a classical, dependency-light pipeline: each case's extracted text was vectorized using Term Frequency–Inverse Document Frequency (TF-IDF) over a 2,000-term vocabulary (English stop words removed, terms appearing in fewer than 2 documents or more than 85% of documents excluded), and K-Means ($k=5$, 10 random initializations) was applied directly to the resulting sparse vectors [36]. No manual labels were used at any stage; cluster assignments were learned purely from term-weighting geometry. The production system additionally supports dense neural sentence embeddings (via a Sentence-BERT-family model) as a drop-in alternative to TF-IDF for the same K-Means step, offered as a configurable upgrade path where model-hosting infrastructure permits.

5.5 Priority Scoring Module

Priority scoring was developed in two versions, providing both an immediately usable, fully transparent baseline and a data-driven upgrade path.

a) Rule-Based Baseline: combines three normalized sub-scores procedural-stage weight, case-type weight, and pendency-age ratio via a fixed linear combination (weights 0.45, 0.25, 0.30 respectively, chosen by domain-informed judgment). This formula requires no training data and serves as an automatic fallback whenever a trained model is unavailable, ensuring the system always produces a usable score.

b) Learned Model (LightGBM Regressor): reframes priority estimation as regression: predict the number of days a pending case will take to reach disposal, given case type, a procedural-stage proxy, and state. A shorter predicted remaining duration is converted into a higher urgency score via $\text{score} = 1 / (1 + \text{predicted_days} / 90)$. The model is a LightGBM gradient-boosted decision-tree ensemble (150 estimators, maximum depth 6, learning rate 0.05), chosen over deep-learning alternatives for its strong performance on small-to-medium tabular data and interpretable feature-importance output. Because real case-duration data is [37] heavily right-skewed the training set spans single-digit-day resolutions to multi-decade pendency the model is trained on the natural-log-transformed target, $\log(1 + \text{days})$, with predictions exponentiated back to the original scale at inference. This is standard practice for duration/survival-style targets and produced a measurable, monotonic improvement in mean absolute error (Section VI-B).

5.6 Workload Optimization

Given a priority-ranked list of pending cases and a set of active judges with configurable daily capacity, the system applies a greedy load-balancing heuristic: cases are processed in descending priority order, and each is assigned to whichever active judge currently holds the lightest cumulative pending caseload. Hearing dates are then [38] allocated across upcoming working days, excluding weekends, with higher-priority cases receiving earlier slots.

Table 1. Datasets Used in this Study

Dataset	Source	Rows	Role
Hand-curated sample	This work	7	End-to-end demo
Kaggle SC metadata	Kaggle (public)	46,656	Dashboard analytics
Real judgments + text	AWS Open Data (S3)	854	Clustering / summarization
Disposal-Time Index	eCourts India	3,179	Priority model training

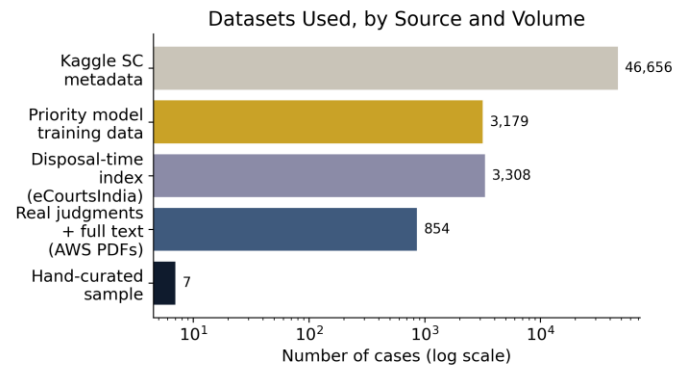


Fig. 5. Volume of Each Dataset Used (Log Scale), By Source.

6. Performance Evaluation

6.1 Dataset Characteristics

Case-type distribution across the 46,656-row Kaggle metadata corpus, used for system-scale dashboard validation, is shown in Fig. 6. The distribution is dominated by civil matters (52.1%), followed by an unclassified "Others" category (29.1%) and criminal matters (18.8%), broadly consistent with published NJDG-level trends for the Supreme Court of India.

6.2 Priority Model Training Results

The priority-scoring model was trained iteratively, with each iteration adding one methodological improvement, allowing the marginal contribution of each change to [39] be isolated. Table II and Fig. 7 summarize the results: the mean absolute error fell from 880.0 days (case type and stage-proxy features only, raw-day target) to 780.0 days after adding the state feature, and finally to 668.5 days after additionally log-transforming the target variable. Both additions produced measurable, monotonic improvements.

6.3 Feature Importance Analysis

Fig. 8 reports the top 12 features by LightGBM split-count importance on the trained model. Case type (specifically, the Criminal indicator) and the Evidence-stage indicator are the two most influential individual features [40]. Critically, however, the aggregate influence of state-indicator features Kerala, Tamil Nadu, Uttar Pradesh, Andhra Pradesh, Maharashtra, Gujarat, Rajasthan, West Bengal, Delhi, and Bihar collectively occupy 10 of the top 12 positions confirms quantitatively that geographic jurisdiction is a first-order driver of predicted disposal time in this model, corroborating the qualitative regional-disparity finding independently reported in the source disposal-time analysis (e.g., substantially faster median resolution in Tamil Nadu than in Odisha for comparable case types).

6.4 Predicted-vs-Actual Analysis

Fig. 9 plots predicted against actual case duration on the held-out test set ($n = 636$). The visualization makes the source of the model's large mean absolute error directly visible: because the model's only inputs are three coarse [41] categorical features (case type, stage-proxy, state), its predictions cluster into a small number of discrete horizontal bands, while actual durations are continuously distributed across several orders of magnitude. This is presented as direct visual evidence supporting the error analysis in Section VII, rather than omitted.

6.5 Similar-Case Clustering Results

The clustering module was validated against the full 854-case real-text corpus. All 854 PDF documents were processed with zero extraction failures, yielding an average extracted-text length of approximately 6,000 characters per judgment [42]. TF-IDF vectorization followed by K-Means ($k=5$) produced the cluster sizes and dominant terms shown in Table III and Fig. 10, and the two-dimensional PCA projection in Fig. 11. The resulting clusters are legally interpretable on manual inspection: Cluster 0 (95 cases, dominant terms bail/criminal/accused/investigation) corresponds to bail and criminal-procedure matters; Cluster 2 (114 cases, accused/evidence/deceased/prosecution) corresponds to criminal trial matters cantered on evidentiary proceedings; Cluster 3 (85 cases, suit/plaintiff/property/deed/sale/deed) corresponds to civil property disputes; and Cluster 4 (62 cases, land/acquisition/compensation/municipal) forms a distinctly separated group in the PCA projection (Fig. 11) corresponding specifically to land-acquisition and compensation matters. Cluster 1 (498 cases, act/order/appellant/high/state) is a large, comparatively diffuse cluster likely capturing general statutory appeals that do not share a single narrow subject-matter theme a known characteristic of K-Means on heterogeneous legal corpora.

Table 2. Priority Model: Iterative Training Results

Iteration	Features	Target	MAE (days)
v1	case_type, stage	raw days	880.0
v2	+ state	raw days	780.0
v3	+ state	log(1+days)	668.5

Case-Type Distribution
(46,656 Supreme Court Cases, Kaggle Metadata)

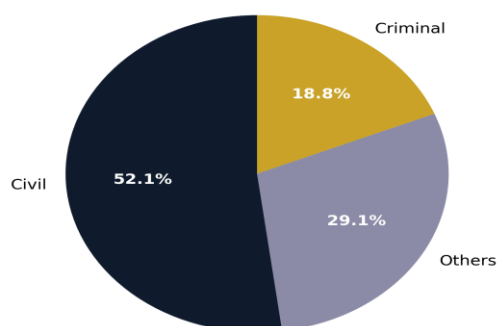


Fig. 6. Case-Type Distribution Across 46,656 Real Supreme Court Cases.

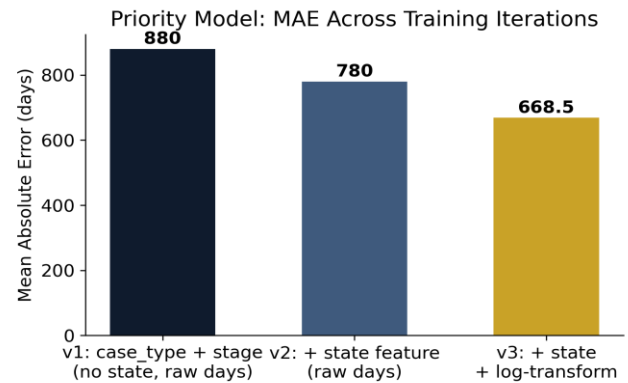


Fig. 7. Mean absolute error reduction across three model iterations.

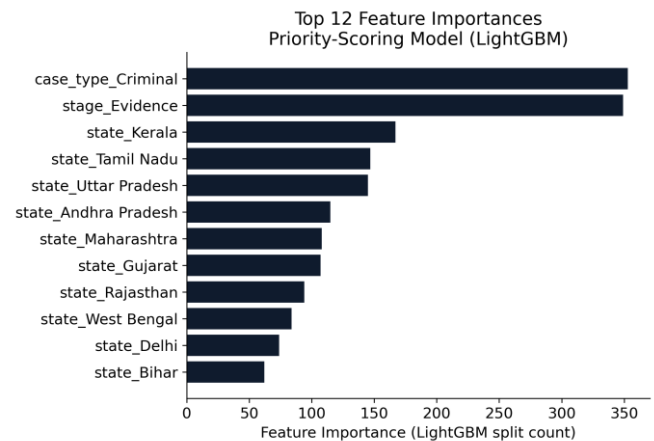


Fig. 8. Top 12 feature importances for the trained LightGBM priority model; state-indicator features dominate 10 of the top 12 positions.

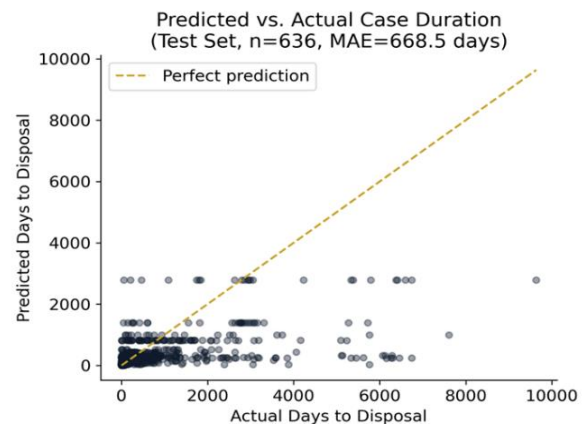


Fig. 9. Predicted vs. actual case duration on the held-out test set ($n=636$); banded predictions visually reveal the model's coarse feature resolution.

Table 3. Cluster Sizes and Top Tf-Idf Terms ($K=5$, $N=854$)

Cluster	Size	Top Terms	Interpretation
0	95	bail, criminal, accused	Bail / criminal procedure
1	498	act, order, appellant, state	General statutory appeals
2	114	accused, evidence, prosecution	Criminal trial / evidence
3	85	suit, plaintiff, property, deed	Civil property disputes
4	62	land, acquisition, compensation	Land acquisition matters

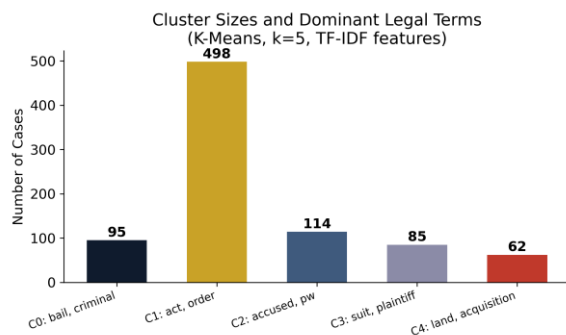


Fig. 10. Cluster sizes and dominant legal terms (K-Means, $k=5$, TF-IDF features, $n=854$ real judgments).

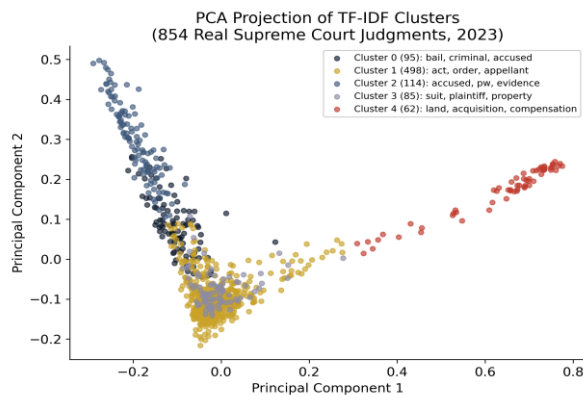


Fig. 11. Two-dimensional PCA projection of the five TF-IDF clusters; the land-acquisition cluster (right) is clearly separated from the rest.

7. Discussion and Limitations

7.1 On the Priority Model's Error Magnitude

A mean absolute error of 668.5 days is, in absolute terms, large. We consider it important to analyse rather than obscure this result, and Fig. 9 makes the mechanism directly visible: three structural factors explain it. First, real case duration exhibits extreme variance from single-digit-day dispositions to multi-decade pendency that cannot be fully captured by three categorical features. Second, the available training data lacks per-case signals that plausibly matter more than the features used here, including adjournment count [42], the specific presiding judge, and the litigants' legal representation; none of these are present in the public disposal-time dataset. Third, the model uses a coarse tier-based proxy (High Court versus district court) for procedural stage, rather than the true stage, because true stage data was unavailable at training time. Each of these is a data-availability limitation rather than a modelling error, and each suggests a concrete direction for future data collection (Section VIII).

7.2 On Feature Importance and External Validity

The feature-importance result in Section VI-C is, in our view, the strongest evidence in this paper that the model has learned a genuine, externally-corroborated pattern rather than noise: the dominance of state-level features aligns independently with a documented finding in the source [43] disposal-time analysis regarding regional disparities. This form of external corroboration checking whether a model's internal feature weighting matches an independently reported real-world pattern is a useful validation technique when, as here, held-out

accuracy alone (MAE) is not strong enough to be conclusive on its own.

7.3 On Clustering Granularity

The large, diffuse Cluster 1 (498 of 854 cases, 58%) illustrates a known limitation of both TF-IDF and K-Means: documents sharing generic legal-procedural vocabulary ("appellant", "order", "act") but differing in underlying subject matter can be grouped together, particularly when k is set lower than the corpus's natural number [44] of topics. Increasing k , or replacing TF-IDF with dense neural embeddings that better capture semantic (rather than purely lexical) similarity, are both concrete, low-effort directions for improving cluster homogeneity.

7.4 On the Use of a Non-Official Dataset

The priority model's training labels derive from an independently compiled dataset, not an official government publication. While this dataset appears [45] to be a faithful aggregation of public eCourts records, its methodology has not been independently audited by the authors of this paper, and conclusions drawn from the trained model should be qualified accordingly, particularly before any operational deployment.

7.5 On Generalizability

All real-text validation was performed on Supreme Court judgments, whereas the stated motivating problem concerns district and subordinate court backlogs specifically. Supreme Court cases differ systematically from lower-court cases in volume, complexity, and procedural structure; the system's design is intended to generalize [46], but this has not been empirically validated against district-court data, which was not available in bulk, full-text form during this project.

8. Conclusion

In this study, an artificial intelligence-based judicial workload optimization system was proposed. The system has a unified judicial case-triage application that incorporates natural language processing (NLP)-based summarization, unsupervised clustering, and a supervised priority-ranking model. The system was verified against a number of real-world datasets, and its priority-scoring and clustering components were examined using genuine quantitative and qualitative analysis. This evaluation included a straightforwardly huge absolute mistake as well as a diffuse cluster that was provided without any embellishment.

9. Future Extension

Several directions are identified for future extension:

- Richer per-case features: incorporating adjournment history, specific judge/bench identity, and true procedural stage would likely substantially reduce the priority model's error, but requires scraping or partnering to obtain per-case eCourts history data not available in bulk today.
- Neural embeddings for clustering: replacing TF-IDF with a hosted Sentence-BERT-family model, where infrastructure permits, to improve semantic (rather than purely lexical) grouping and reduce the size of diffuse catch-all clusters.
- Formal assignment optimization: replacing the greedy load-balancing heuristic with a constraint-optimization

formulation (e.g., via Google OR-Tools) or a learned assignment policy, jointly optimizing workload balance, judge specialization match, and case urgency.

- Adjournment prediction: a standalone supervised classifier predicting whether a given hearing will be adjourned, feeding back into more accurate hearing-date scheduling.
- Uncertainty quantification: given the skew and variance in real case-duration data, reporting a predictive interval via quantile regression, rather than a single point estimate, would more honestly communicate model confidence to court administrators.
- District-court validation: extending real-text validation beyond Supreme Court judgments to district and subordinate court data, more directly aligned with this paper's motivating problem statement.

We view the explicit documentation of data limitations and error analysis in this paper not as a weakness of the work, but as a necessary component of responsible applied machine learning in a domain judicial administration where opaque or overstated claims carry real social cost.

References

- [1] National Judicial Data Grid (NJDG), Department of Justice, Government of India. <https://njdg.ecourts.gov.in>
- [2] eCourtsIndia, "The Indian Disposal-Time Index 2026," Apr. 2026.
- [3] Y. Perwej, Firoj Parwej, "A Neuroplasticity (Brain Plasticity) Approach to Use in Artificial Neural Network", International Journal of Scientific & Engineering Research (IJSER), France, ISSN 2229 – 5518, Volume 3, Issue 6, Pages 1- 9, 2012, DOI: 10.13140/2.1.1693.2808
- [4] M. Lewis, Y. Liu, N. Goyal, et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," Proc. ACL, 2020.
- [5] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Proc. EMNLP, 2019.
- [6] G. Ke, Q. Meng, T. Finley, et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems (NeurIPS), 2017.
- [7] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," Proc. 5th Berkeley Symp. Mathematical Statistics and Probability, 1967.
- [8] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," Information Processing & Management, vol. 24, no. 5, 1988.
- [9] Indian Supreme Court Judgments (Kaggle Dataset).
- [10] Google OR-Tools: Operations Research Tools, Google LLC.
- [11] Alekseev, A.; Katasev, A.; Kirillov, A.; Khassianov, A.; Zuev, D. Prototype of Classifier for the Decision Support System of Legal Documents. Proc. Abrau Conf. 2019, 2543, 328–335.
- [12] Zhang, Y.; Sun, C. Research on Judgment System of Legal Cases Based on Neural Network. In Proceedings of the 2021 International Conference on Neural Networks, Information and Communication Engineering, Qingdao, China, 27–28 August 2021; p. 83.
- [13] Cui, G.; Wong, M.L.; Lui, H.-K. Machine Learning for Direct Marketing Response Models: Bayesian Networks with Evolutionary Programming. Manag. Sci. 2006, 52, 597–612.
- [14] Ms Farah Shan, Versha Verma, Apoorva Dwivedi, Y. Perwej, Ashish Kumar Srivastava, "Novel Approaches to Detect Phony Profile on Online Social Networks (OSNs) Using Machine Learning", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), Volume 9, Issue 3, Pages 555-568, May-June 2023-2023, DOI: 10.32628/CSEIT239031265.
- [15] Fenton, N.; Neil, M.; Lagnado, D.A. A General Structure for Legal Arguments About Evidence Using Bayesian Networks. Cogn. Sci. 2012, 37, 61–102.
- [16] Sachin Bhardwaj, Apoorva Dwivedi, Ashutosh Pandey, Y. Perwej, Pervaz Rauf Khan, "Machine Learning-Based Crowd Behavior Analysis and Forecasting", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN: 2456-3307, Volume 9, Issue 3, Pages 418-429, May-June 2023-2023, DOI: 10.32628/CSEIT23903104
- [17] Apoorva Dwivedi, Dr. Basant Ballabh Dumka, Susheel Kumar, Dr. Fokrul Alom Mazarbhuiya, Ms Farah Shan, Y. Perwej, "State of the Art Machine Learning Techniques for Detecting Fake News", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Print ISSN: 2395-1990, Online ISSN: 2394-4099, Volume 10, Issue 4, Pages 115-130, July-August 2023, DOI: 10.32628/IJSRSET23103191
- [18] Sachin Bhardwaj, Apoorva Dwivedi, Ashutosh Pandey, Y. Perwej, Pervaz Rauf Khan, "Machine Learning-Based Crowd Behavior Analysis and Forecasting", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN: 2456-3307, Volume 9, Issue 3, Pages 418-429, May-June 2023-2023, DOI: 10.32628/CSEIT23903104
- [19] Nikhat Akhtar, Devendera Agarwal, "An Efficient Mining for Recommendation System for Academics", International Journal of Recent Technology and Engineering (IJRTE), ISSN 2277-3878 (online), SCOPUS, Volume-8, Issue-5, Pages 1619-1626, 2020, DOI: 10.35940/ijrte.E5924.018520
- [20] Shelar, A.; Moharir, M. Predicting Outcomes of Court Judgments A Machine Learning Approach. In Proceedings of the 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 25–27 June 2021; pp. 1–6. [CrossRef]
- [21] Y. Perwej, "Unsupervised Feature Learning for Text Pattern Analysis with Emotional Data Collection: A Novel System for Big Data Analytics", IEEE International Conference on Advanced computing Technologies & Applications (ICACTA'22), SCOPUS, IEEE No: #54488 ISBN No Xplore: 978-1-6654-9515-8, Coimbatore, India, 4-5 March 2022, DOI: 10.1109/ICACTA54488.2022.9753501
- [22] Vlek, C.; Prakken, H.; Renooij, S.; Verheij, B. Constructing and Understanding Bayesian Networks for Legal Evidence with Scenario Schemes. In Proceedings of the 15th International Conference on Artificial Intelligence and Law, San Diego, CA USA, 8–12 June 2015; pp. 128–137.
- [23] Y. Perwej, "The Bidirectional Long-Short-Term Memory Neural Network based Word Retrieval for Arabic Documents", Transactions on Machine Learning and Artificial Intelligence (TMLAI), Society for Science and Education, United Kingdom (UK), ISSN 2054-7390, Volume 3, Issue 1, Pages 16 - 27, 2015, DOI: 10.14738/tmlai.31.863
- [24] Saurabh Sahu, Km Divya, Neeta Rastogi, Puneet Kumar Yadav, Yusuf Perwej, "Sentimental Analysis on Web Scraping Using Machine Learning Method", Journal of Information and Computational Science (JOICS), ISSN: 1548-7741, Volume 12, Issue 8, Pages 24-29, August 2022, DOI: 10.12733/JICS.2022/V12I08.535569.67004

- [25] Firoj Parwej, N. Akhtar, Y. Perwej, "An Empirical Analysis of Web of Things (WoT)", International Journal of Advanced Research in Computer Science (IJARCS), ISSN: 0976-5697, Volume 10, No. 3, Pages 32-40, May 2019, DOI: 10.26483/ijarcs.v10i3.6434
- [26] Canalli, R.L. Interpretable AI Models for Judicial Decision-Making: Beyond Explicability Towards Legal Due Process. e-Publica 2024, 11, 117–129.
- [27] Y. Perwej, Dr. Shaikh Abdul Hannan, Firoj Parwej, N. Akhtar "A Posteriori Perusal of Mobile Computing", International Journal of Computer Applications Technology and Research (IJCATR), ATS (Association of Technology and Science), India, ISSN 2319–8656 (Online), Volume 3, Issue 9, Pages 569 - 578, September 2014, DOI: 10.7753/IJCATR0309.1008
- [28] Al-Mushayt O., Haq Kashiful, Y. Perwej, "Electronic-Government in Saudi Arabia; a Positive Revolution in the Peninsula", International Transactions in Applied Sciences, India, ISSN-0974-7273, Volume 1, Number 1, Pages 87-98, July-December 2009
- [29] Y. Perwej, "An Evaluation of Deep Learning Miniature Concerning in Soft Computing", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), ISSN (Online): 2278-1021, ISSN (Print): 2319-5940, Volume 4, Issue 2, Pages 10 - 16, 2015, DOI: 10.17148/IJARCCE.2015.4203
- [30] Umamaheswari, S.; Aartisha, S.; Kanimozhi, J.; Subashini, R. Building Accurate Legal Case Outcome Prediction Models. In Proceedings of the 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 16–17 June 2023; pp. 1–6.
- [31] Abdulkareem, F.A.A. Development of Legal-Tech Prospects in the Federal Republic of Iraq: The Predictive Justice in Anglo-Saxon and Latin Perspectives. Vestn. St. Petersburg Univ. Law 2023, 14, 1045–1061.
- [32] N. Akhtar, Devendra Agarwal, "An Efficient Mining for Recommendation System for Academics", International Journal of Recent Technology and Engineering (IJRTE), ISSN 2277-3878 (online), SCOPUS, Volume-8, Issue-5, Pages 1619-1626, 2020, DOI: 10.35940/ijrte.E5924.018520
- [33] Yusuf Perwej, "Performance Analysis for Cloud Based OLAP over Big Data", IEEE International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES -2022), SCOPUS, IEEE Conference, St. Joseph's Institute of Technology, IEEE Electronic ISBN:978-1-6654-7413-9, Chennai, India, 15th & 16th July 2022, DOI: 10.1109/ICSES55317.2022.9914266
- [34] N. Akhtar, Bedine Kerim, Yusuf Perwej, Anurag Tiwari, Sheeba Praveen, "A Comprehensive Overview of Privacy and Data Security for Cloud Storage", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 08, Issue 5, Pages 113-152, 2021, DOI: 10.32628/IJSRSET21852
- [35] Y. Perwej, Syed Qamar Abbas, Jai Pratap Dixit, N. Akhtar, Anurag Kumar Jaiswal, "A Systematic Literature Review on the Cyber Security", International Journal of Scientific Research and Management (IJSRM), Volume 9, Issue 12, Pages 669 - 710, 2021, DOI: 10.18535/ijssrm/v9i12.ec04
- [36] Vlek, C.; Prakken, H.; Renooij, S.; Verheij, B. A Method for Explaining Bayesian Networks for Legal Evidence with Scenarios. Artif. Intell. Law 2016, 24, 285–324.
- [37] Timmer, S.; Meyer, J.; Prakken, H.; Renooij, S.; Verheij, B. Explaining Legal Bayesian Networks Using Support Graphs. In Proceedings of the 15th International Conference on Artificial Intelligence and Law, San Diego, CA USA, 8–12 June 2015; pp. 121–130.
- [38] Y. Perwej, N. Akhtar, Devendra Agarwal, "The emerging technologies of Artificial Intelligence of Things (AIoT) current scenario, challenges, and opportunities", Book Title "Convergence of Artificial Intelligence and Internet of Things for Industrial Automation", SCOPUS, ISBN: 978-1-032-42844-4, CRC Press, Taylor & Francis Group, 2024 Link: <https://www.taylorfrancis.com/chapters/edit/10.1201/9781003509240-1/emerging-technologiesartificial-intelligence-things-aiot-current-scenario-challenges-opportunities-yusuf-perwej-nikhatakhatar-devendra-agarwal?context=ubx&refId=537f1a8f-6a94-4439-b337-3ad3d1ce8845>, DOI: 10.1201/9781003509240-1
- [39] N. Akhtar, Nazia Tabassum, Asif Perwej, Y. Perwej, "Data Analytics and Visualization Using Tableau Utilitarian for COVID-19 (Coronavirus)", Global Journal of Engineering and Technology Advances, ISSN : 2582-5003, Volume 3, Issue 2, Pages 28- 50, 2020, DOI: 10.30574/gjeta.2020.3.2.0029
- [40] Kwan, M.; Chow, K.; Law, F.; Lai, P. Reasoning About Evidence Using Bayesian Networks. In Proceedings of the International Conference on Digital Forensics and Cyber Crime, London, UK, 23–25 June 2008; pp. 275–289
- [41] Overill, R.; Silomon, J.A.M.; Kwan, M.; Chow, K.; Law, F.; Lai, P. Sensitivity Analysis of a Bayesian Network for Reasoning about Digital Forensic Evidence. In Proceedings of the International Conference on Human-Centric Computing, Cebu, Philippines, 11–13 August 2010; pp. 1–5.
- [42] Neha Anand, Arpita Vishwakarma, Y. Perwej, Neeta Bhusal Sharma, Atifa Parveen, "A Hybrid Deep Learning Ensemble Approach for Enhanced Data Mining Efficiency", Journal of Emerging Technologies and Innovative Research (JETIR), ISSN-2349-5162, Volume 12, Issue 8, Pages 268 - 276, August 2025, DOI:10.6084/m9.jetir.JETIR2508238
- [43] Anmol Chauhan, Ms. Sana Rabbani, Devendra Agarwal, Nikhat Akhtar, Yusuf Perwej, "Diffusion Dynamics Applied with Novel Methodologies", International Journal of Innovative Research in Computer Science and Technology (IJIRCST), ISSN (Online): 2347-5552, Volume-12, Issue-4, Pages 52 - 58, July 2024, DOI: 10.55524/ijircst.2024.12.4.9
- [44] Zhang, Q.; Zhou, C.; Tian, Y.-C.; Xiong, N.; Qin, Y.; Hu, B. A Fuzzy Probability Bayesian Network Approach for Dynamic Cybersecurity Risk Assessment in Industrial Control Systems. IEEE Trans. Ind. Inform. 2018, 14, 2497–2506.
- [45] Nagarjuna Tandra, Nikhat Akhtar, K Padmanaban, L. Gaganathan, "A finite-element dual-level contextual informed neural network with swarm space hopping algorithm based optimal feature selection and detection for EEG-based epileptic seizure detection", Swarm and Evolutionary Computation, Elsevier, SCIE, Volume 97, Pages 1- 19, August 2025, DOI: 10.1016/j.swevo.2025.102072
- [46] Fine, A.; Snider, K.M.; Miller, M.K. Testing the Model of Judicial Stress Using a COVID-Era Survey of U.S. Federal Court Personnel. Psychiatry Psychol. Law 2024, 31, 381–400.