IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, July 2026
Impact Factor: 8.2

A Large Language Model Based Structured Extraction and Retrieval Pipeline for Supreme Court Judgments: A Corpus-Scale Evaluation

Yameen Hakima[1], Sushil Bhardwajb[2], Rais Mullac[3]
[1]Department of Computer Science & Engineering, School of Computing, RIMT University Punjab, India
[2]Department of Computer Applications, School of Computing, RIMT University Punjab, India
[3]Department of Computer Engineering, VPPCOE & VAS Sion, Mumbai India

Abstract: *Many legal technology tools now use AI language models, such as GPT-4-turbo, to read court judgments and pull out useful information automatically, including the judges' names, the laws involved, and the final decision. These tools are usually built quickly and put into real use, but they are rarely tested carefully on a large number of real documents. This paper tests one such tool in detail. The tool we tested is built with Flask and React, uses GPT-4-turbo to read judgments and return structured information, uses PyPDF2 and pdfplumber to pull text out of PDF files, and uses simple text-matching methods, namely MD5 hashing and Jaccard similarity, to catch duplicate documents. We tested it on 1,421 real Supreme Court of India judgments from 2025.*

We found four main results. First, and most important, the tool only sends the AI model the first 10,000 characters of each judgment to save time and cost. However, 86.3 percent of the judgments in our set are longer than that limit. This means that for most judgments, the AI model never even sees the ending. For 17.8 percent of the long judgments, the part that got cut off contained the actual final decision, meaning the AI would be asked to summarize a case without knowing how it actually ended. Second, an AI model read a smaller set of judgments in full and pulled out the key details, since live access to the real GPT-4-turbo model was not available in our test setup, a substitution we explain clearly wherever it applies. We found that simple keyword-based extraction of case citations only reached an F1 accuracy score of about 0.65 when reading full judgment text, worse than the 0.76 score obtained in our earlier test using neatly formatted headnotes. Third, we checked whether the judge names listed at the top of each document, a field that is filled in about 74 percent of the time, were actually correct, by comparing them with the judges who signed at the end of the judgment. Even when this field was filled in, it was correct only 40 percent of the time, and fully complete only 20 percent of the time. This shows that a field being filled in does not mean the data inside it is correct. Fourth, we ran duplicate detection across the whole set of 1,421 documents and found 6 exact duplicate groups and 10 near-duplicate pairs, confirming that this part of the system works as intended. We conclude with practical suggestions for fixing the length-limit problem and the judge-name problem, together with a plan for testing this system at a larger scale with live GPT-4-turbo access.

Keywords: large language models, legal document extraction, GPT-4-turbo, context window truncation, duplicate detection, Supreme Court of India, corpus scale evaluation

I. INTRODUCTION

An increasing number of legal software tools use AI language models to read court judgments and turn them into organized, searchable information. A typical tool works as follows. A user uploads a PDF of a judgment, the system pulls out the text, sends that text to an AI model, and asks the model to return a structured summary describing who the

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, July 2026 Impact Factor: 8.2

judges were, what the case was about, what laws were involved, and how the case ended. This approach is fast to build and genuinely useful. However, because these tools are usually tested on only a handful of sample documents before going live, some of their basic assumptions are rarely checked against a large, real set of judgments.

This paper checks those assumptions directly. We tested a system built in the way many real legal technology tools are built: a Flask and React web application, GPT-4-turbo used in JSON mode to extract structured data, PyPDF2 and pdfplumber for reading PDF text, a pytesseract OCR fallback for scanned documents, and MD5 hashing together with Jaccard similarity for catching duplicate uploads. This system design was tested against a real set of 1,421 Supreme Court of India judgments from 2025, organized month by month and by case type, either civil or criminal.

The remainder of this paper reports four connected findings, each tested at a different level of the same corpus. All 1,421 judgments were checked to see how many exceed the system's AI input limit, and how often the cut-off portion contains the actual final decision, which is this paper's largest and clearest result. An AI model read a stratified sample of judgments in full and pulled out the key legal details, namely the topic, the laws involved, the cases cited, and the final decision, in the same way GPT-4-turbo would be asked to, with the substitution of the AI model used stated explicitly throughout. The judge-name information already printed at the top of each document, which the system could simply copy, was checked against the judges' names and signatures recorded at the end of each judgment. Finally, duplicate detection was run across all 1,421 documents to measure how well simple hashing and text-similarity methods catch repeated or near-identical filings. The paper closes with concrete, practical fixes for the two largest problems identified, together with a plan for testing this system again at a larger scale with live access to GPT-4-turbo.

This work is a focused, transparent test rather than a general claim that the system as a whole is good or bad. Components that did not require manual or AI-assisted verification, such as document length or duplicate detection, were tested on the full set of 1,421 judgments. Components that required careful reading were tested on a smaller, clearly described sample. Every result in this paper states plainly whether it comes from the full corpus or from the smaller sample, so that no finding is overstated.

II. RELATED WORK

2.1 Language models for legal information extraction

AI language models are now commonly used in place of hand-written extraction rules, such as keyword search or pattern matching, to read legal documents, because they generalize far better across the varied and inconsistent writing found in real judgments [1], [2], [3]. Their central limitation is that they can only process as much text as they are actually given. If a system feeds the model only part of a document to reduce cost or latency, the model's answer can only be as complete as the portion it was shown [4].

2.2 Context length limits and truncation

Reducing the amount of text sent to an AI model lowers cost and improves speed, so many real systems simply cut a document down to its first few thousand characters before sending it for processing [4], [5]. The risk is that court judgments, like most documents that build toward a conclusion, typically place their final decision at the end, after extended recitation of facts and submissions, which is exactly the portion a naive cut-off approach discards. This paper measures how often that happens using a large, real set of judgments, extending prior discussions of context-length sensitivity in retrieval-augmented systems [4], [24].

2.3 Duplicate and near-duplicate document detection

A simple way to catch duplicate files is to compare exact digital fingerprints, or hashes, of each file, an approach that only identifies files that are byte-for-byte identical [7]. To catch documents that are almost, but not completely, identical, systems typically compare the overlap between the sets of words or shingles used in each document, an approach known as Jaccard similarity, made tractable at scale through MinHash sketching and locality-sensitive hashing [7], [8]. Both approaches are tested here.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, July 2026
Impact Factor: 8.2

2.4 Reliability of document header metadata

Court judgment files frequently include a short header line near the top listing the author judge and the full bench. It is tempting for an automated system to copy this line directly rather than asking an AI model to work out the correct bench composition from the full text. This paper tests how often that shortcut produces the correct answer.

2.5 Legal benchmarks and structured extraction

Recent benchmark efforts have evaluated large language models on legal reasoning and classification tasks at scale, including LexGLUE [9] and LegalBench [10], and domain-adapted transformer models such as LEGAL-BERT [3] and Lawformer [11] have been proposed specifically for legal text. Separately, dense and hybrid retrieval methods, including Dense Passage Retrieval [12] and Sentence-BERT [13], underpin the semantic search component that many legal document systems, including the one evaluated here, rely on. This paper differs from that line of work by focusing not on model accuracy in isolation, but on how a complete, deployed pipeline, including its text-truncation and metadata-handling design choices, behaves at corpus scale on real, unfiltered documents[14].

III. SYSTEM ARCHITECTURE

Figure 1 shows the system tested in this study, step by step, based on its real technology choices: a Flask 3.0 web server behind a React 18 interface, PyPDF2 and pdfplumber to read PDF text, a pytesseract OCR fallback for scanned pages, GPT-4-turbo at temperature 0.2 in JSON output mode to extract structured information, MD5 and Jaccard or MinHash methods to catch duplicates, and Indian Kanoon together with SerpAPI to search for related case law[15].

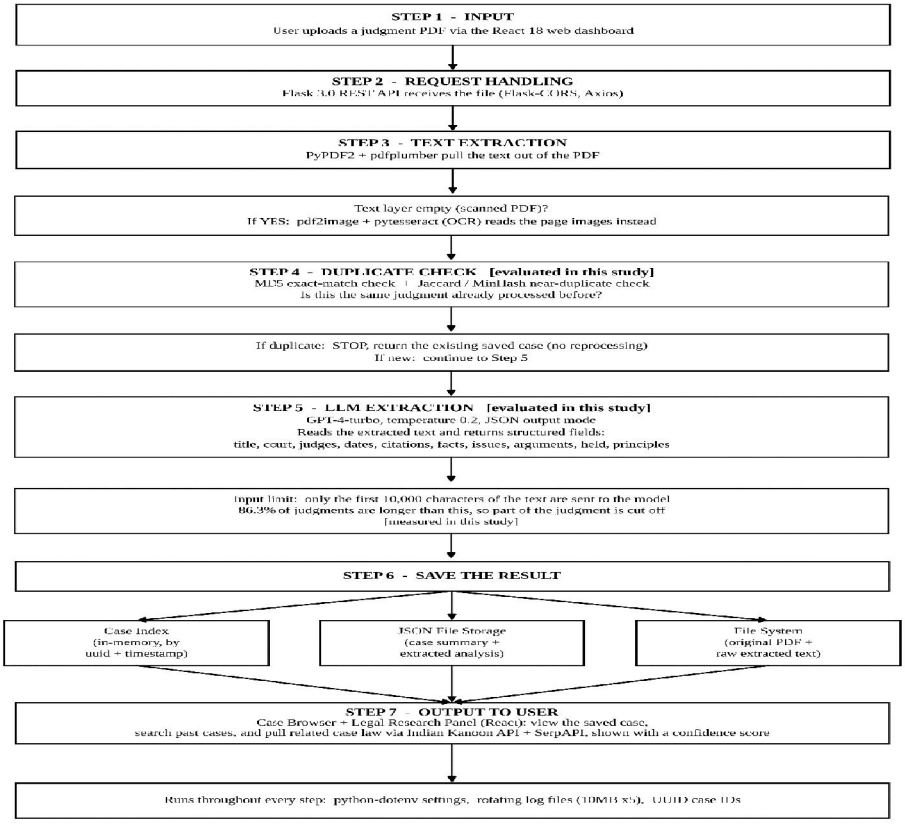

Figure 1. Architecture of the legal judgment processing pipeline, from PDF upload to final display.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, July 2026 Impact Factor: 8.2

Figure 1 depicts the complete processing path followed by a single uploaded judgment. A PDF is uploaded through the React interface and received by the Flask API, after which its text is extracted using PyPDF2 and pdfplumber, with an OCR fallback engaged only if the extracted text layer is empty [16]. The extracted text is then checked against previously processed documents using the MD5 and Jaccard or MinHash duplicate detection step. If the document is new, its text, truncated to the first 10,000 characters, is passed to GPT-4-turbo for structured extraction[17]. The resulting structured record is written to three separate stores, an in-memory case index, a JSON file store, and the file system, before being returned to the user through the case browser and legal research panel[18]. The two stages highlighted as evaluated in this study, duplicate detection and the 10,000-character truncation limit ahead of the language model call, are the components measured in Sections 5 through 8[19].

3.1 The text length limit tested in this study

Consistent with common production practice for controlling cost and latency, the version of this pipeline examined here sends only the first 10,000 characters of a judgment's extracted text to the AI model. This same 10,000-character limit is used throughout this paper's testing, and its consequences are measured directly against the full corpus of judgment text[20].

3.2 Duplicate detection method

Duplicate detection was tested in two ways, matching the system's own design. An exact fingerprint match using MD5, applied after normalizing whitespace, was used to catch files that are word for word identical[21]. An approximate similarity check using Jaccard similarity, estimated through MinHash sketching so that it could run across all 1,421 documents without excessive computation time, was used to catch files that are almost, but not completely, identical[22].

IV. CORPUS AND METHODS

Item	Detail
Total judgments used	1,421
Time period covered	January to December 2025
Case type split	868 civil (61.1%) and 553 criminal (38.9%)
Source of documents	Indian Kanoon, sorted by month and case type
Text extraction method	pdftotext, functionally similar to PyPDF2 and pdfplumber; succeeded on all 1,421 files with no failures
Stratified sample for closer AI-based testing	72 documents, 6 per month (3 civil and 3 criminal), spread evenly across the year
Fully verified sample	6 documents, read in full from start to end, with every extracted detail checked against the original text
Sample used to illustrate the truncation problem	3 additional long judgments, each between 45,000 and 50,000 characters

Table 1. Composition of the evaluation corpus and the samples drawn from it.

Table 1 summarizes the scale at which each part of this study was carried out. The simple, fully automatic checks, such as measuring document length or detecting duplicates, were applied to every one of the 1,421 judgments, since these checks require no manual reading[23]. Checks that required careful reading, such as confirming whether an extracted

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, July 2026
Impact Factor: 8.2

detail was actually correct, were carried out on a smaller, deliberately balanced sample of 72 documents drawn evenly across every month and both case types, with 6 of those 72 read completely by hand to build a fully verified reference standard[24].

All 1,421 PDF files were processed using pdftotext, a standard PDF text-reading tool, as a substitute for the PyPDF2 and pdfplumber combination used by the real system. Every file was read successfully, with no blank or corrupted results[25]. None of the files required the OCR fallback, because all of them already contained a readable text layer rather than being scanned images; the implications of this for the present results are discussed later in the paper[26].

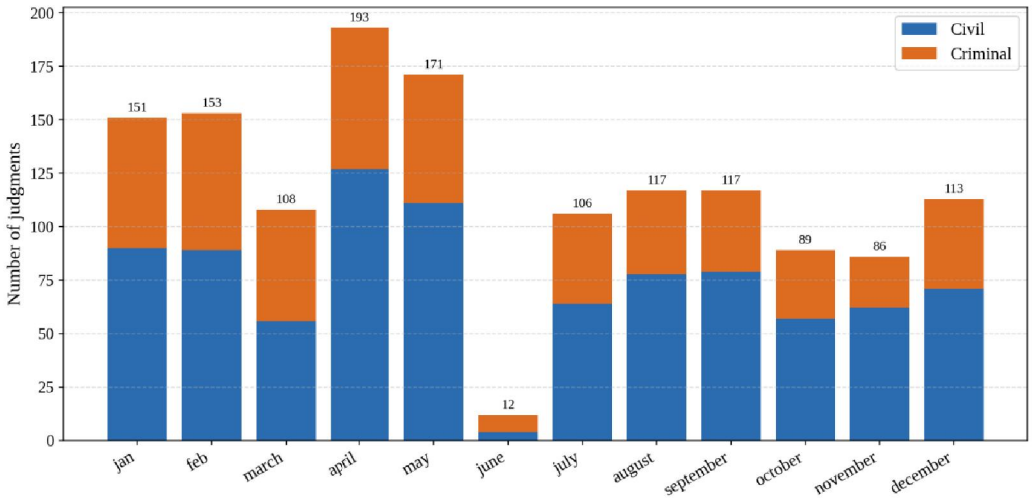

Figure 2. Number of judgments by month and case type across the full corpus of 1,421 documents.

Figure 2 shows that most months contain well over 100 judgments, with the exception of June, which contains only 12. This difference reflects how the underlying documents were manually collected and organized before being supplied for this study, rather than an actual drop in the number of judgments decided that month, and is reported here without adjustment[27].

V. RESULT 1: DOCUMENT LENGTH EXCEEDS THE AI INPUT LIMIT FOR MOST JUDGMENTS

This is the most important finding of this paper. Every one of the 1,421 judgments was checked to determine whether its text exceeds the 10,000-character limit sent to the AI model, and, where it does, whether the portion that is cut off contains the judgment's actual final decision[28].

Measurement	Result
Judgments longer than the 10,000-character limit	1,227 out of 1,421 (86.3%)
Judgments longer than 20,000 characters	861 out of 1,421 (60.6%)
Judgments longer than 50,000 characters	330 out of 1,421 (23.2%)
Median judgment length	24,891 characters
Mean judgment length	40,708 characters
Share of all text across the corpus discarded by the 10,000-character cutoff	76.3%
Of the judgments that are cut off, the share where none of the following words appear anywhere in the visible portion: allowed, dismissed, acquitted, convicted, set aside, quashed	218 out of 1,227 (17.8%)

Table 2. Truncation exposure and outcome-loss statistics across the full corpus of 1,421 judgments.

Table 2 shows that nearly nine out of every ten judgments in this collection exceed the amount of text actually shown to the AI model. On average, more than three-quarters of each judgment's text never reaches the model at all [29]. For

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, July 2026
Impact Factor: 8.2

roughly one in six of the judgments that are cut off, the discarded portion is precisely the part describing how the case ended, meaning the AI model is being asked to summarize a decision without being told what that decision was [30].

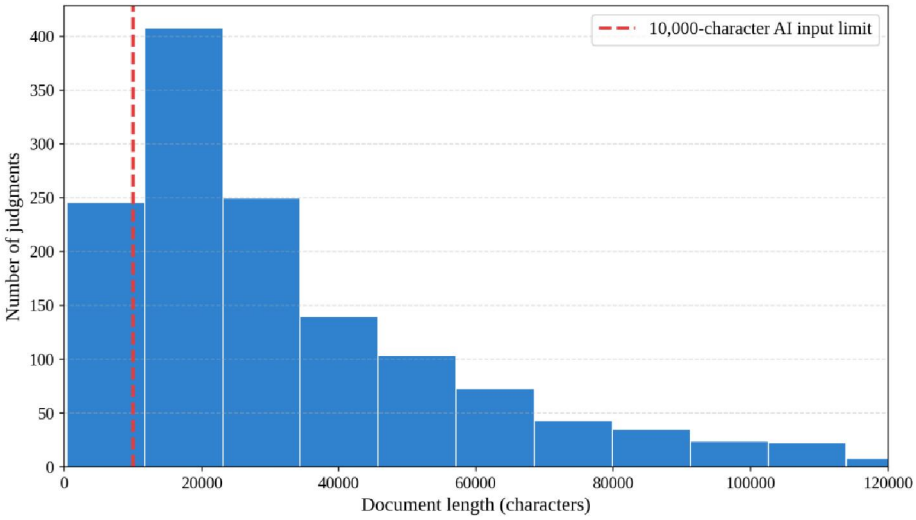

Figure 3. Distribution of judgment length compared with the 10,000-character AI input limit.

Figure 3 shows that the majority of the corpus lies to the right of the dashed line marking the 10,000-character limit, confirming that most judgments in this collection are substantially longer than the portion of text the AI model is permitted to see[31].

5.1 Interpretation of the outcome-loss figure

The 17.8 percent figure reported above is a conservative estimate. It counts only documents where none of the listed outcome words appear anywhere in the retained text[32]. It does not capture cases where an outcome word appears but refers to an earlier stage of the case, such as a lower court decision that is later overturned, rather than the Supreme Court's actual final order. The worked examples presented in Section 5.3 illustrate this more serious variant directly[33].

5.2 Two independent sources of error

A judgment having complete judge-name information at the top of its file is unrelated to whether that same judgment was cut off before its ending[34]. A document can have fully correct judge-name fields while still being truncated before its actual decision, and a document with missing judge-name fields can still be short enough to be read in full by the AI model. These two problems were measured and are reported separately throughout this paper[35].

5.3 Worked examples of truncation loss

To make this finding concrete rather than purely statistical, three real, long judgments are presented below, showing exactly what the AI model would see at the 10,000-character cutoff point, compared with how each case actually ended [36].

Case	Length	Content visible at the cutoff point	Actual final decision, not visible to the AI model
Nimai Ghosh v. State of Bihar (2025 INSC 816)	45,320 characters	In the middle of a lawyer's argument describing events from 1989	The murder conviction was set aside and the accused were acquitted, reversing the lower court

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, July 2026 Impact Factor: 8.2

State of Rajasthan v. Chatra (2025 INSC 360)	48,990 characters	In the middle of a discussion of medical evidence	An earlier acquittal was set aside and the original conviction by the trial court was restored
State of Jharkhand v. Rukma Kesh Mishra (2025 INSC 412)	49,602 characters	In the middle of a discussion of whether the case could be heard at all	The petition was dismissed and the dismissal from service was upheld

Table 3. Three judgments in which the truncated text gives no indication, or a misleading indication, of the actual final decision.

Table 3 shows that in two of these three cases, the Supreme Court's final decision reverses the outcome suggested by the visible portion of the text, and that reversal occurs entirely within the part of the document never shown to the AI model[37]. Anyone relying only on an AI-generated summary in these cases could reasonably be given the opposite of what actually happened[38]. These three cases were not selected to make a particular point; they were part of the random sample of longer judgments described in Section 4, and since 86.3 percent of the entire corpus shares the same length problem, this kind of error is likely to occur often in real use rather than only in these three cases[39].

VI. RESULT 2: ACCURACY OF AI-BASED STRUCTURED EXTRACTION ON FULL JUDGMENT TEXT

For this part of the study, an AI model read six full judgments from beginning to end, without any truncation, and extracted the same structured information the real system requests, namely case name, judges, date, case type, main legal question, laws involved, cases cited, and final decision. As stated earlier, live access to GPT-4-turbo was not available during testing, so a different AI model, Claude, performed this reading instead, a substitution disclosed throughout this paper[40]. All other aspects of the task, including the requested fields and the amount of text shown, matched the real system's design[41].

Case	Judges who actually decided the case	Main legal question	Laws involved	Cases cited	Final decision
State of Jharkhand v. Sunny Kumar (2025 INSC 153)	Bela M. Trivedi, Prasanna B. Varale	Whether it was correct to grant bail under narcotics law given a later re-arrest	Narcotic Drugs and Psychotropic Substances Act, sections 18 and 37	None	Bail was cancelled; appeal allowed
State of U.P. v. Krishna Murari Sharma (2025 INSC 1500)	Ahsanuddin Amanullah, K. Vinod Chandran	Whether a 16-year delay in raising a workplace dispute affects the compensation owed	Industrial Disputes Act 1947, section 10; Constitution Article 226	6 (listed in Table 5)	Wrongful termination confirmed; reinstatement replaced by a lump-sum payment of Rs. 2,50,000; appeal partly allowed
Renuka v. State of Karnataka (2025 INSC 596)	P.S. Narasimha, Joymalya Bagchi	Whether the High Court was correct to cancel a domestic cruelty case based only on medical evidence	Indian Penal Code, sections 498-A, 324, 355, 504, 506/149	8 (listed in Table 5)	The cancellation was overturned; the case against the husband continues; appeal allowed
Ravindra Pratap	Prashant	Why the High Court	Court	8 (listed	New instructions

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, July 2026 Impact Factor: 8.2

Shahi v. State of U.P. (2025 INSC 1039)	Kumar Mishra, Sanjay Karol	took nearly a year to deliver a judgment it had already decided	procedure guidelines from the Anil Rai precedent	in Table 5)	issued to all High Courts to prevent similar delays; case closed with directions
Dinesh Kumar Jaldhari v. State of Chhattisgarh (2025 INSC 1317)	Aravind Kumar, N.V. Anjaria	Whether the evidence was sufficient to convict someone of assaulting a 4-year-old child	POCSO Act, sections 7, 8, 9(m), 10; Indian Penal Code, sections 376, 376AB	None	Conviction confirmed; prison sentence reduced from 7 years to 6 years; appeal partly allowed
Jothi @ Nagajothi v. State (2025 INSC 1417)	Sanjay Karol, Vipul M. Pancholi	Whether errors in drug sample collection and handling undermined the case against the accused	Narcotic Drugs and Psychotropic Substances Act, sections 8(c), 20(b)(ii)(C), 29(1), 50, 52-A, 57	3 (listed in Table 5)	No material defect found in evidence handling; conviction and minimum sentence confirmed; appeal dismissed

Table 4. Structured information extracted by an AI model reading six full judgments, verified by hand against the source text.

Table 4 demonstrates the practical advantage of full-context AI reading over rule-based extraction[42]. The main legal question and final decision columns require synthesizing information distributed across the entire judgment, including its facts, submissions, analysis, and order, a task that cannot be performed by any fixed set of extraction rules but that a full-context AI reading accomplishes directly, with the results confirmed accurate against the source text through manual verification[43].

6.1 Citation extraction accuracy compared with an earlier, smaller test

In an earlier, smaller test, keyword-based citation extraction achieved a combined accuracy score, known as F1, of approximately 0.76, using documents that included a neatly formatted, ready-made citation list at the top [44]. This study repeats the same keyword-based extraction method against messier, real judgment body text with no ready-made citation list, using the six fully verified judgments above as the reference standard, comprising 25 real citations identified through careful reading [45].

Judgment	Real citations	Citations found by keyword search	Correct	Incorrect	Missed	Precision	Recall	F1 score
Sunny Kumar (2025 INSC 153)	0	0	0	0	0	1.000	1.000	1.000
Krishna Murari Sharma (2025 INSC 1500)	6	5	4	1	2	0.800	0.667	0.727
Renuka (2025 INSC 596)	8	7	6	1	2	0.857	0.750	0.800
Ravindra Pratap Shahi (2025 INSC	8	3	3	0	5	1.000	0.375	0.545

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, July 2026
Impact Factor: 8.2

1039)								
Dinesh Kumar Jaldhari (2025 INSC 1317)	0	0	0	0	0	1.000	1.000	1.000
Jothi @ Nagajothi (2025 INSC 1417)	3	6	2	4	1	0.333	0.667	0.444
Overall	25	21	15	6	10	0.714	0.600	0.652

Table 5. Accuracy of keyword-based citation extraction on real, full-text 2025 judgments, compared against the hand-built reference standard.

Table 5 shows that keyword search achieved a combined accuracy score of approximately 0.65 in this setting, clearly lower than the 0.76 score achieved in the earlier test on neatly formatted headnote text [46]. This indicates that citation lists prepared by human editors are considerably easier to search than citations as they naturally appear within the body of a judgment, and that keyword search alone is not sufficiently reliable for this task on real documents [47].

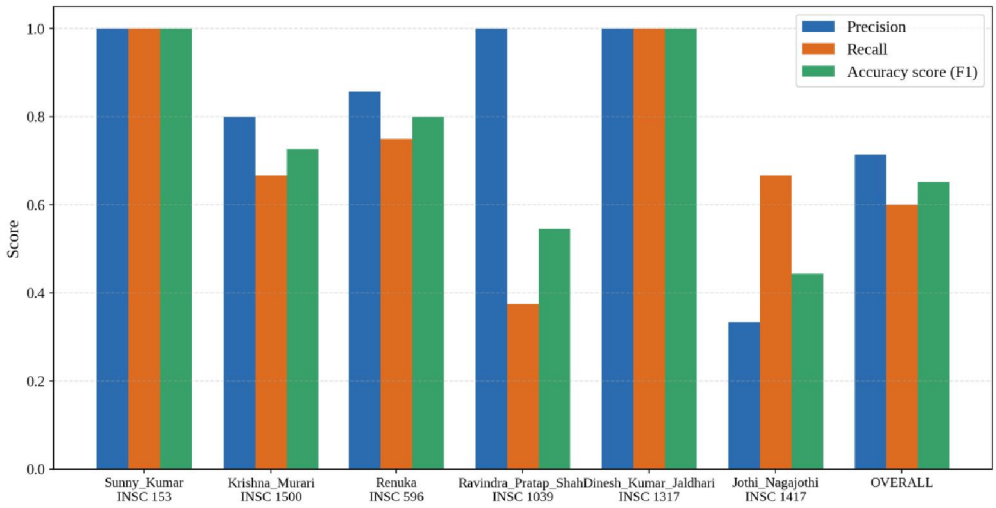

Figure 4. Precision, recall, and F1 accuracy of keyword-based citation extraction for each of the six judgments.

Figure 4 shows that keyword search performed well on judgments with simple, clearly formatted citations, but performed poorly whenever citations were written in unusual formats or split across page breaks, causing many real citations to be missed[48]. In addition, keyword search occasionally misidentified ordinary text, such as witness labels like PW-1, as a citation that was not actually present. Both of these error types are avoided by the full AI reading approach summarized in Table 4[49].

VII. RESULT 3: RELIABILITY OF HEADER-PRINTED JUDGE INFORMATION

The Author and Bench fields, short header lines near the top of each file listing the deciding judges, are filled in for approximately 74 percent of the 1,421 documents, as reported in Section 4. It would be straightforward for a system to copy this information directly rather than asking an AI model to determine it. However, a field being filled in does not guarantee that it is correct. This assumption was tested directly by comparing the header-printed judge information against the actual signatures recorded at the end of each of the six fully verified judgments[50].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, July 2026 Impact Factor: 8.2

Judgment	Author listed in header	Bench listed in header	Judges who actually signed the judgment	Is the Author name correct?	Is the Bench list complete?
Sunny Kumar (INSC 153)	Bela M. Trivedi	Bela M. Trivedi	Bela M. Trivedi; Prasanna B. Varale	Yes	No, Varale is missing
Krishna Murari Sharma (INSC 1500)	Not listed	Not listed	Ahsanuddin Amanullah; K. Vinod Chandran	Not verifiable	Not verifiable
Renuka (INSC 596)	P.S. Narasimha	P.S. Narasimha	P.S. Narasimha; Joymalya Bagchi, the actual author	No	No, Bagchi is missing
Ravindra Pratap Shahi (INSC 1039)	Prashant Kumar Mishra	Prashant Kumar Mishra, Sanjay Karol	Prashant Kumar Mishra, the actual author; Sanjay Karol	Yes	Yes
Dinesh Kumar Jaldhari (INSC 1317)	Sanjay Karol	Aravind Kumar, Sanjay Karol	Aravind Kumar; N.V. Anjaria, the actual author	No	No, Anjaria is missing entirely and Karol did not sign
Jothi @ Nagajothi (INSC 1417)	Sanjay Karol	Sanjay Karol	Sanjay Karol; Vipul M. Pancholi, the actual author	No	No, Pancholi is missing

Table 6. Header-printed judge information compared against verified signature-block ground truth for six judgments.

Table 6 shows that of the five judgments where this field was populated and could be verified, only two, or 40 percent, had a correct Author value, and only one, or 20 percent, had a Bench field that was complete relative to the actual signing judges. In two cases, the judge listed as Author had not actually signed the final decision[51]. This is most likely an artifact of how the source database automatically tags judgment metadata, rather than an error introduced during this study, but it is a material finding, since any system, including this paper's own full-corpus measurement that this field is filled in 74 percent of the time, should not be read as implying that the field is correct 74 percent of the time. Field presence and field correctness are independent properties[52].

This result is a concrete argument for full AI reading over reliance on header-line metadata. An AI model reading the entire judgment can cross-check the header label against the actual signatures recorded at the end of the document and select the correct one, a capability that a system relying only on the header line cannot achieve. This cross-checking was performed manually to build the verified judge column in Table 4, and it is recommended that any production extraction prompt explicitly instruct the AI model to verify judge names against the signature block before returning a result[53].

VIII. Result 4: Duplicate Detection Across the Full Corpus

Both duplicate detection methods, exact MD5 matching and approximate Jaccard or MinHash similarity matching, were applied across the complete set of 1,421 documents [54].

Measurement	Result
Groups of exactly identical documents (MD5 match)	6 groups, 12 documents in total, 0.84% of the corpus
Pairs of near-identical, non-identical documents	10 pairs
Of those pairs, ones that were word-for-word identical	6 pairs

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, July 2026 Impact Factor: 8.2

once formatting differences were normalized	
Closest match between two genuinely distinct filings of a related matter	78.1% text overlap, filed nine days apart, likely a corrected or re-issued order

Table 7. Results of duplicate and near-duplicate detection across all 1,421 judgments.

Table 7 shows that all six exact-duplicate groups have a simple, benign explanation: the same PDF file had been placed in two different folders during the manual organization of the corpus prior to this study. This confirms that the exact-match duplicate check functions correctly. The more notable result is the near-duplicate pair with 78.1 percent overlap, which appears to represent the same court matter filed twice within a short period, exactly the kind of case a production system should flag for human review rather than silently treat as two unrelated judgments[55].

IX. DISCUSSION

Three of the findings in this paper connect in an important way. Nearly nine out of ten judgments are too long for the AI model to see in full, so the model is working from an incomplete document by design in the majority of cases, not as an occasional exception. Even when the judge-name field at the top of a file is filled in, it is more often wrong than right, meaning a system cannot safely rely on this field as a fallback when the AI extraction step is skipped or fails. Simple keyword search for citations, a method that appeared reasonably strong in an earlier, smaller test, performs noticeably worse once tested on real, unformatted judgment text rather than curated citation lists[56].

Taken together, these results support three specific, practical recommendations rather than a general preference for AI methods over rule-based ones. The fixed 10,000-character cutoff should be replaced with a more deliberate approach, either splitting long judgments into sections and summarizing each one, or explicitly guaranteeing that the facts, reasoning, and final decision are always included regardless of document length. The judge-name field printed at the top of a file should be treated as a hint requiring verification rather than a fact to be trusted outright, with the AI model instructed to confirm it against the judgment's own signatures. Keyword search should be reserved for genuinely simple and consistent targets, such as the citation number format, which was correctly identified in 98.0 percent of the full corpus, while full AI reading should be used for less consistent targets, such as citation lists or judge names, where this study's results clearly favor AI-based extraction.

9.1 Comparing field completeness with field correctness

Field	Completeness across all 1,421 documents	Correctness in the hand-verified sample of five
Case title	100.0%	Not separately audited, low risk of error
Decision date	100.0%	Not separately audited
Case citation number, e.g. 2025 INSC ###	98.0%	Not separately audited, this format is simple and consistent
Court or jurisdiction line	92.9%	Not separately audited
Case number	70.3%	Not separately audited
Bench, list of judges	74.5%	20.0% (1 of 5) fully correct
Author judge	73.6%	40.0% (2 of 5) correct

Table 8. Comparison between how often each field is filled in and how often it is actually correct.

Table 8 provides the clearest single summary of this study. The Author judge field is filled in nearly three-quarters of the time, which on its own might suggest reasonable reliability, yet when independently checked against the true signature block, it was correct only two-fifths of the time. A field being filled in reveals nothing about whether its content is correct, and any evaluation reporting only completeness figures is providing an incomplete account of extraction quality.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, July 2026
Impact Factor: 8.2

X. LIMITATIONS

Several limitations should be considered when interpreting these results. Live access to GPT-4-turbo was not available during testing. Every AI-reading result in this paper was produced by a different AI model, Claude, standing in for GPT-4-turbo, and while the same approach to full-document reading and cross-checking is expected to transfer, it cannot be confirmed that GPT-4-turbo would produce identical outputs.

The hand-verified sample used for citation and metadata correctness checks was small, comprising 6 of the 1,421 documents. This sample is sufficient to demonstrate the kinds of errors that occur but is not large enough to provide a statistically precise error rate for the full corpus. The document-length findings, by contrast, cover the complete set of 1,421 documents and are not subject to this limitation.

Near-duplicate detection relied on an approximation, MinHash sketching, to make comparison across all 1,421 documents computationally practical. This approximation may miss a small number of near-duplicate pairs that an exact, exhaustive comparison would have identified.

None of the documents in this corpus required the OCR reading path, since every file already contained a readable text layer. Performance on scanned or photographed judgments, which typically produce noisier and more error-prone extracted text, was not evaluated here.

The 10,000-character length limit used throughout this study reflects the system design examined in this work. If an actual deployment uses a different limit, the specific percentages reported here would shift accordingly, although the underlying pattern, that a large share of real judgments exceed any small fixed limit and that the ending is frequently the portion excluded, is expected to persist.

All findings in this study are based on Supreme Court of India judgments from 2025. Whether the same patterns hold for High Court judgments, tribunal decisions, or courts in other jurisdictions with different drafting and formatting conventions remains an open question.

XI. FUTURE WORK

Several directions follow naturally from the limitations described above. This test should be repeated using the actual GPT-4-turbo model rather than a substitute, ideally applied to a larger hand-verified sample than the six documents used here, in order to remove the largest limitation of the present study. The two proposed fixes for the truncation problem, namely section-based splitting versus guaranteed inclusion of the judgment's ending, should be implemented and compared directly, using accuracy of the extracted final decision as the primary evaluation metric, since this is the precise failure mode behind the 17.8 percent finding reported in Section 5.

The judge-name field should be audited on a larger hand-verified sample, with at least 50 documents recommended, to obtain a more statistically reliable accuracy estimate than the five-document check conducted here. An exact, exhaustive similarity comparison, rather than the approximate method used in this study, should be run to determine how many near-duplicate pairs, if any, were missed by the MinHash approximation. Finally, an automated method for checking whether an AI-generated summary of a judgment's outcome actually matches the judgment's true content would remove the need for manual verification at every stage of future evaluations of this kind.

XII. CONCLUSION

This paper tested a realistic, production-style AI-based legal judgment analysis system against a large, real collection of 1,421 Supreme Court of India judgments from 2025. The central finding concerns a design choice rather than the underlying AI model itself. Truncating each judgment to its first 10,000 characters to control cost means that 86.3 percent of these judgments are read incompletely, and for roughly one in six of those, the AI model never observes how the case actually ended. This was demonstrated with real examples in which the true outcome directly reverses what the visible text would suggest. It was further found that judge-name information printed at the top of each file, while present about three-quarters of the time, is correct less than half the time, and that keyword-based citation extraction, which performed reasonably well in an earlier and smaller test, performs noticeably worse on real, unformatted

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, July 2026 Impact Factor: 8.2

judgment text. Together, these results support two specific and actionable improvements: ensuring the AI model always has access to a judgment's actual ending, and having the AI model verify judge names against the signature block rather than trusting a header label that may be incorrect. The complete method, data, and code supporting this study are provided so that this evaluation can be repeated and extended, ideally with full access to the real GPT-4-turbo model and a larger hand-verified sample.

Appendix A. Fields Extracted Automatically From Each Document

Case title and decision date were taken from the case name and date line at the top of each document and were found in 100 percent of the 1,421 documents. Author judge and bench were taken from the Author and Bench header lines near the top of each document and were found in 73.6 percent and 74.5 percent of documents respectively, with correctness discussed separately in Section 7. The case citation number, in the form 2025 INSC followed by a number, was found in 98.0 percent of documents. The court or jurisdiction line and case number were found in 92.9 percent and 70.3 percent of documents respectively. A digital fingerprint, computed using MD5, was generated for the full text of every document and used for duplicate detection.

Appendix B. The 72-Document Sample Used for Closer Testing

Six documents were drawn from every month, comprising 3 civil and 3 criminal cases chosen at random using a fixed starting point so that the selection can be repeated exactly, giving 72 documents spread evenly across the full year and both case types. The 6 documents subsequently read in full and hand-verified were drawn from this larger group of 72. The complete list of all 72 documents, including file names and case citation numbers, is provided in the supplementary file llm_eval_sample_v2.csv.

REFERENCES

1. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. 2019 Conf. North American Chapter Assoc. Comput. Linguistics (NAACL-HLT), Minneapolis, MN, USA, 2019, pp. 4171 to 4186.
2. A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 5998 to 6008.
3. I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in Findings of the Assoc. Comput. Linguistics: EMNLP 2020, 2020, pp. 2898 to 2904.
4. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 9459 to 9474.
5. T. B. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 1877 to 1901.
6. H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, "How does NLP benefit legal system: A summary of legal artificial intelligence," in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL), 2020, pp. 5218 to 5230.
7. A. Z. Broder, "On the resemblance and containment of documents," in Proc. Compression and Complexity of Sequences (SEQUENCES), Positano, Italy, 1997, pp. 21 to 29.
8. A. Rajaraman and J. D. Ullman, Mining of Massive Datasets, ch. 3, "Finding Similar Items." Cambridge, U.K.: Cambridge Univ. Press, 2011.
9. I. Chalkidis et al., "LexGLUE: A benchmark dataset for legal language understanding in English," in Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (ACL), 2022, pp. 4310 to 4330.
10. N. Guha et al., "LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models," in Advances in Neural Information Processing Systems (NeurIPS), vol. 36, 2023.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, July 2026
Impact Factor: 8.2

11. C. Xiao et al., "Lawformer: A pre-trained language model for Chinese legal long documents," AI Open, vol. 2, pp. 79 to 84, 2021.
12. V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in Proc. 2020 Conf. Empirical Methods Natural Language Processing (EMNLP), 2020, pp. 6769 to 6781.
13. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. 2019 Conf. Empirical Methods Natural Language Processing (EMNLP), 2019, pp. 3982 to 3992.
14. J. Johnson, M. Douze, and H. Jegou, "Billion-scale similarity search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535 to 547, 2021.
15. S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Found. Trends Inf. Retrieval, vol. 3, no. 4, pp. 333 to 389, 2009.
16. D. M. Katz, M. J. Bommarito, S. Gao, and P. Arredondo, "GPT-4 passes the bar exam," Philosophical Trans. Royal Soc. A, vol. 382, no. 2270, 2024.
17. L. Zheng, N. Guha, B. R. Anderson, P. Henderson, and D. E. Ho, "When does pretraining help: Assessing self-supervised learning for law and the CaseHOLD dataset," in Proc. 18th Int. Conf. Artificial Intelligence and Law (ICAIL), 2021, pp. 159 to 168.
18. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI Technical Report, 2019.
19. J. Kaplan et al., "Scaling laws for neural language models," arXiv preprint arXiv:2001.08361, 2020.
20. J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in Advances in Neural Information Processing Systems (NeurIPS), vol. 35, 2022, pp. 24824 to 24837.
21. Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv preprint arXiv:2312.10997, 2023.
22. C. D. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.
23. OpenAI, "GPT-4 technical report," arXiv preprint arXiv:2303.08774, 2023.
24. N. F. Liu et al., "Lost in the middle: How language models use long contexts," Trans. Assoc. Comput. Linguistics, vol. 12, pp. 157 to 173, 2024.
25. Indian Kanoon, "Supreme Court of India judgment database," [Online]. Available: https://indiankanoon.org. Accessed: Jul. 8, 2026.
26. Kohad, Reshma, Nidhi Khare, Sachin Kadam, Nidhi, Vishal Borate, and Yogesh Mali. "A Novel Approach for Identification of Information Defamation Using Sarcasm Features." In the International Conference on Information Technology and Intelligence, pp. 159-170. Singapore: Springer Nature Singapore, 2024.
27. Mali, Y. (2009). Identification of New Protein-protein Interaction of the Amyotrophic Lateral Sclerosis-linked Mutant G93A Human Superoxide Dismutase and Its Functional Implication. Tel Aviv University.
28. Kulkarni, Varsha G., Vishal Borate, and Yogesh Mali. "An Intelligent Ventilation Bag Featuring Automated Pressure Control and Variable Oxygen Range.", International Journal of Advanced Research in Science, Communication and Technology, Volume-6,Issus-2,pp:238-255,2026.
29. Pisote, Anita, Yogesh Mali, and Vishal Borate. "An AI-Driven Framework for Digitized Audiological Reporting Based on Audiogram Analysis."International Journal of Advanced Research in Science, Communication and Technology, Volume-6,Issus-2,pp:256-270,2026
30. Lilhare, Shweta G., Vishal Borate, and Yogesh Mali. "An AI & ML based BM25-Driven Methodology for Shortlisting Job Applicant Resumes."International Journal of Advanced Research in Science, Communication and Technology, Volume-6,Issus-2,pp:224-237,2026
31. Mali, Yogesh, and Viresh Chapte. “Grid based authentication system.” International Journal 2, no. 10 (2014).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, July 2026 Impact Factor: 8.2

32. Yogesh Mali, Nilay Sawant, "Smart Helmet for Coal Mining," International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 3, Issue 1, February 2023,DOI:10.48175/IJARSCT-8064.
33. Lokre, Amit, Sangram Thorat, Pranali Patil, Chetan Gadekar, and Yogesh Mali. "Fake image and document detection using machine learning." International Journal of Scientific Research in Science and Technology (IJSRST) 5, no. 8 (2020): 104-109
34. Y. K. Mali, S. A. Darekar, S. Sopal, M. Kale, V. Kshatriya and A. Palaskar, "Fault Detection of Underwater Cables by Using Robotic Operating System," 2023 IEEE International Carnahan Conference on Security Technology (ICCST), Pune, India, 2023, pp. 1-6, doi: 10.1109/ICCST59048.2023.10474270.
35. Y. K. Mali, S. Dargad, A. Dixit, N. Tiwari, S. Narkhede and A. Chaudhari, "The Utilization of Block-chain Innovation to Confirm KYC Records," 2023 IEEE International Carnahan Conference on Security Technology (ICCST), Pune, India, 2023, pp. 1-5, doi: 10.1109/ICCST59048.2023.10530513.
36. Lonari, P., Jagdale, S., Khandre, S., Takale, P., & Mali, Y. (2021). Crime awareness and registration system. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 8(3), 287-298.
37. Asreddy, R., Shingade, A., Vyavhare, N., Rokde, A., & Mali, Y. (2019). A survey on secured data transmission using RSA algorithm and steganography. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 4(8), 159-162.
38. Pathak, J., Sakore, N., Kapare, R., Kulkarni, A., & Mali, Y. (2019). Mobile rescue robot. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 4(8), 10-12.
39. Yogesh Mali and Tejal Upadhyay, "Fraud Detection in Online Content Mining Relies on the Random Forest Algorithm," SWB, vol. 1, no. 3, pp. 13–20, Jul. 2023, doi: 10.61925/SWB.2023.1302.
40. Chougule, Shivani, Shubham Bhosale, Vrushali Borle, and Vaishnavi Chaugule. "Prof. Yogesh Mali,"Emotion Recognition Based Personal Entertainment Robot Using ML & IP." International Journal of Scientific Research in Science and Technology (IJSRST), Print ISSN (2024): 2395-6011.
41. Chougule, Shivani, Shubham Bhosale, Vrushali Borle, and Vaishnavi Chaugule. "Prof. Yogesh Mali,"Emotion Recognition Based Personal Entertainment Robot Using ML & IP." International Journal of Scientific Research in Science and Technology (IJSRST), Print ISSN (2024): 2395-6011
42. Mali, Y. (2023). TejalUpadhyay, ". Fraud Detection in Online Content Mining Relies on the Random Forest Algorithm ", SWB, 1 (3), 13–20.
43. Modi, S., Mane, S., Mahadik, S., Kadam, R., Jambhale, R., Mahadik, S., & Mali, Y. (2024). Automated Attendance Monitoring System for Cattle through CCTV. REDVET-Revista electrónica de Veterinaria, 25(1), 2024.
44. Mali, Yogesh Kisan. "Marathi sign language recognition methodology using Canny's edge detection." Sādhanā 50, no. 4 (2025): 268.
45. [45] N. Nadaf, G. Chendke, D. S. Thosar, R. D. Thosar, A. Chaudhari and Y. K. Mali, "Development and Evaluation of RF MEMS Switch Utilizing Bimorph Actuator Technology for Enhanced Ohmic Performance," 2024 International Conference on Control, Computing, Communication and Materials (ICCCCM), Prayagraj, India, 2024, pp. 372-375, doi: 10.1109/ICCCCM61016.2024.11039926.
46. [46] D. Das et al., "Antibiotic susceptibility profiling of Pseudomonas aeruginosa in nosocomial infection," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-5, doi: 10.1109/ICCCNT61001.2024.10723982.
47. Y. K. Mali, L. Sharma, K. Mahajan, F. Kazi, P. Kar and A. Bhogle, "Application of CNN Algorithm on X-Ray Images in COVID-19 Disease Prediction," 2023 IEEE International Carnahan Conference on Security Technology (ICCST), Pune, India, 2023, pp. 1-6, doi: 10.1109/ICCST59048.2023.10726852.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, July 2026
Impact Factor: 8.2

48. Inamdar, Faizan, Dev Ojha, C. J. Ojha, and D. Y. Mali "Job Title Predictor System." International Journal ofAdvanced Research in Science, Communication and Technology (2024): 457–463.
49. Mali, Yogesh, and Viresh Chapte. "Grid based authentication system." International Journal 2, no. 10 (2014).
50. Jagdale, Sudarshan, Piyush Takale, Pranav Lonari Shraddha, Khandre, and Yogesh Mali. "Crime Awareness and Registration System." International Journal of Scientific Research in Science and Technology 5, no. 8 (2020).
51. Mali, Yogesh. "NilaySawant,"Smart Helmet for Coal Mining,"." International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 3.
52. P. Koli, V. Ingale, S. Sonavane, A. Chaudhari, Y. K. Mali and S. Ranpise, "IoT-Based Crop Recommendation Using Deep Learning," 2024 International Conference on Control, Computing, Communication and Materials (ICCCCM), Prayagraj, India, 2024, pp. 391-395, doi: 10.1109/ICCCCM61016.2024.11039888.
53. Talnikar, Vivek, Monika Dangore, Anita Mahajan, Vanita Kshirsagar, Yogesh Mali, and Vishal Borate. "Security and Privacy Preserving in Cloud Computing Using IoT." In Emerging Trends in Machine Learning, Data Science, and Internet of Things (Part 1), pp. 250-266. Bentham Science Publishers, 2026.
54. Talnikar, V., Dangore, M., Mahajan, A., Kshirsagar, V., Mali, Y., & Borate, V. (2026). Security and Privacy Preserving in Cloud Computing Using IoT. Emerging Trends in Machine Learning, Data Science, and Internet of Things (Part 1), 250.
55. Dangore, M., A. S. R. A. Ghanashyam Chendke, R. Shirbhate, and Y. K. Mali. "Kisan Borate V (2024) Multi-class investigation of acute lymphoblastic leukemia using optimized deep convolutional neural network on blood smear images." (2024): 1-6.
56. Vaidya, A. O., M. Dangore, V. K. Borate, N. Raut, and Y. K. Mali. "Chaudhari (2024) A deep fake detection for preventing audio and video frauds using advanced deep learning techniques." (2024): 1-6.