AI Lawyer: A Literature Review of Retrieval-Augmented Legal Advice, Document Automation and Forensic Document Verification

Ankur Singh[1], Dr. M K Jayanthi Kannan[2]
[1]*PG Student, School of Computing Science Engineering and Artificial Intelligence (SCAI), VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh*
[2]*Professor, School of Computing Science Engineering and Artificial Intelligence (SCAI), VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh*

***Abstract*—Legal information is spread across long statutes, rules, and case material; routine legal documents are drafted by repeating standard text, and digital documents submitted as evidence may have been edited without notice. Large language models (LLMs) can answer legal questions fluently, but they can also produce confident, incorrect statements, which is serious in a legal setting. This paper reviews the literature behind *AI Lawyer*, a Python and FastAPI prototype that brings three functions behind one API: retrieval-augmented legal question answering, template-based generation of a non-disclosure agreement (NDA), and preliminary forensic screening of PDF and image files. Eight studies on retrieval-augmented generation, sentence embeddings, legal language models, legal hallucination, image forensics, and local language models are compared on objective, technology, methodology, efficiency, and open issues.**

The comparison shows that retrieval, domain adaptation and forensic heuristics are each well studied on their own, while their integration in a private, locally deployable legal workflow is not. The paper then describes the proposed system, its algorithms and parameters, and reports a small preliminary test of the Error Level Analysis (ELA) module on synthetic images: the default threshold of 35.0 flagged none of 60 images, and the score separated edited from authentic images only weakly (AUC 0.61). The legal knowledge base has not yet been populated, so answer quality is not claimed. The paper closes with the contribution, limitations, and a plan for evaluation.

***Index Terms*—Retrieval-Augmented Generation, Legal Informatics, Large Language Models, Sentence Embeddings, Document Generation, Digital Forensics, Error Level Analysis, FastAPI.**

I. INTRODUCTION

Legal work depends on reading source material accurately, drafting carefully, and judging documents as evidence. A person trying to understand a legal topic may have to search several long documents. A professional preparing recurring documents such as NDAs replaces the same few fields each time. Organizations that receive scanned or digital documents often cannot tell quickly whether a file has been edited. These tasks differ, but each can be helped by software that is built with care. Recent language models can summarize and explain text, but fluency does not guarantee correctness. Studies of legal queries report frequent hallucination by general-purpose models [5]. Retrieval-Augmented Generation (RAG) addresses part of this problem by giving the model passages retrieved from a document collection at answer time, so that the answer can be traced to a source [1]. The *AI Lawyer* prototype applies this pattern to a local collection of legal documents and runs the model on the user's own machine. The scientific rationale rests on three points. First, semantic retrieval with sentence embeddings finds passages by meaning rather than by exact words [2]. Second, a locally served model keeps queries and uploaded documents on the user's machine [8]. Third, standard documents and basic file checks are repeatable tasks that can be automated, provided the limits of each automated check are stated openly. The prototype exposes legal question answering, NDA generation, and forensic screening through one FastAPI service. The rest of the paper is organized as follows. Section

II reviews related work. Section III states the research gap and problem. Sections IV and V describe the proposed design and architecture, Section VI the algorithms, Section VII the implementation status, Section VIII a preliminary test, and Sections IX and X the contribution and conclusion. The user interface design with respect to users' viewing experience across static and dynamic websites. It highlights the importance of website presentation and content delivery in supporting usability and improving the overall user experience [30]. The automatic content creation and rearrangement mechanism improves cloud storage space utilization. Organizing digital content efficiently, with potential benefits for storage management and resource optimization in cloud-based environments [21].

II. LITERATURE REVIEW OF EXISTING SYSTEMS

The reviewed work falls into four groups: retrieval and generation [1, 2, 8], legal-domain language models and their weaknesses [3, 4, 5], and image forensics [6, 7]. Each study below is described by its objectives, technology, methodology, efficiency, and challenges, and Table 1 summarizes all eight. Cybersecurity, authentication, cloud computing, and machine learning: Research on cybersecurity has examined threats and countermeasures across networked systems, cloud platforms, and user authentication mechanisms. The work on protecting confidential data over distributed cloud storage [13] addresses the importance of safeguarding sensitive information in distributed environments, while the book on network security attacks and countermeasures [14] provides a broader foundation for understanding attack models and defensive strategies. Image-based authentication using randomized selection [15] and the performance analysis of bag-of-password authentication across Python, Java, and PHP implementations [16] explore approaches to improving authentication security and implementation efficiency. Research on cybercrime and open-source software [17], strategic information-security planning using decentralized identity mechanisms [18], and machine-learning techniques for malware detection [19] further highlight the need for layered security, proactive threat detection, and user-oriented protection. The hybrid deep-learning approach for intrusion detection in AWS cloud environments [20] extends this direction by applying intelligent methods to cloud security, while the survey of confidential-data protection in distributed storage [13] emphasizes data privacy and secure storage as essential concerns. Collectively, these studies establish a foundation for combining authentication, intrusion detection, machine learning, and secure cloud architecture to address increasingly complex cyber threats.

2.1. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks [1]

The key objectives of the study include: combining the knowledge stored in a language model's parameters with an external, searchable document store, so that generation can use retrieved evidence and the knowledge base can be updated without retraining. The technology used in the study includes: A pre-trained sequence-to-sequence generator (BART) paired with a dense passage retriever over a vector index of Wikipedia, in two formulations called RAG-Sequence and RAG-Token. The methodology follows: The query is encoded, the top passages are found by maximum inner-product search, and the generator is conditioned on the query and the passages. The generator and query encoder are fine-tuned together. The study demonstrates efficiency with: State-of-the-art results were reported on several open-domain question-answering benchmarks, and the generated text was more specific and factual than that of a model without retrieval. However, key challenges include: The index was general-domain (Wikipedia), so there are no legal sources, no jurisdiction or date handling and no document tools. Answers can still contain unsupported statements when retrieval fails.

2.2. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks [2]

The key objectives of the study include: producing sentence embeddings that can be compared with cosine similarity, so that semantic search and clustering become fast enough for large collections. The technology used in the study includes: Siamese and triplet network structures built on pre-trained BERT and RoBERTa, with a pooling layer that outputs one fixed-length vector per sentence. The methodology follows: The networks are fine-tuned on sentence-pair data so that sentences with similar meaning lie close together in vector space; similarity is then a cheap

vector comparison. The study demonstrates efficiency with: The paper reports that finding the most similar pair among 10,000 sentences falls from about 65 hours with BERT to about 5 seconds, while accuracy on semantic-similarity tasks stays competitive. However, key challenges include: Training is general-domain, so statutory language, exact section numbers and rare legal terms may be represented poorly. The prototype's all-MiniLM-L6-v2 model belongs to this sentence-transformer family and has not been tested on legal queries.

2.3. LEGAL-BERT: The Muppets Straight Out of Law School [3]

The key objectives of the study include: finding out whether language models pre-trained on legal text work better on legal tasks than general BERT models, and supplying smaller models for limited hardware. The technology used in the study includes: BERT-base models in three settings: general BERT used as is, further pre-trained on legal text, and pre-trained from scratch on legal text, plus lightweight versions. The methodology follows: Models are pre-trained on English legal text (legislation, court cases and contracts) and then fine-tuned on legal classification and sequence-tagging tasks, with a search over hyperparameters. The study demonstrates efficiency with: Domain-specific pre-training generally gave better results than general BERT on the legal tasks tested, notably on the harder ones. However, key challenges include: The models are encoders for classification and tagging, not generators of answers. They are English-only and tied to the jurisdictions of the training text, and they include no retrieval or document tools.

2.4. How Does NLP Benefit the Legal System: A Summary of Legal Artificial Intelligence [4]

The key objectives of the study include: Summarizing how natural language processing and machine learning are applied to legal tasks, and identifying open problems. The technology used in the study includes: A review of embedding-based and symbol-based methods for legal text. The methodology follows: A survey of tasks such as legal judgment prediction, similar-case matching, and legal question answering, relating methods to tasks and available datasets. The study demonstrates efficiency with: It gives a structured map of the field and argues that embedding-based and symbolic methods should be combined. However, key challenges include: It is a review, not a system. Interpretability and ethics remain open, and it offers no guidance on building an integrated, deployable tool.

2.5. Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models [5]

The key objectives of the study include measuring how often general-purpose LLMs produce legal hallucinations, that is, answers that are fluent but wrong. The technology used in the study includes: Several general-purpose LLMs, tested without retrieval on verifiable questions, mainly about United States federal court cases. The methodology follows: Queries of varying complexity are posed, the answers are compared with known case information, and the errors are classified. The study demonstrates efficiency with: The study finds hallucinations to be common, with reported rates of roughly 69% to 88% on specific queries, and that models often fail to notice a false premise in the question. However, key challenges include: It motivates grounding and citations but proposes no system. It covers one jurisdiction and models without retrieval, so the effect of retrieval is not measured.

2.6. A Picture's Worth: Digital Image Analysis and Forensics [6]

The key objectives of the study include: giving investigators a quick, visual way to spot parts of an image that may have been edited. The technology used in the study includes Error Level Analysis (ELA), based on recompression of a JPEG image. The methodology follows: The image is saved again at a known quality, the difference from the original is computed and amplified, and regions whose error level differs from their surroundings may have a different compression history. The study demonstrates efficiency with: The method is simple and fast, needs no training data, and is useful for triage. However, key challenges include: It is a heuristic that needs human interpretation, depends on image content, resizing and earlier compression, and does not give a probability of tampering.

2.7. Image Forgery Detection [7]

The key objectives of the study include: Surveying the techniques for detecting tampering in digital images. The technology used in the study includes: Pixel-based, format-based (JPEG), camera-based, physics-

based, and geometry-based techniques. The methodology follows: Techniques are grouped by the statistical trace of tampering that each exploit, for example cloning, resampling, double compression, and inconsistent lighting. The study demonstrates efficiency with: It provides a taxonomy and shows that no single technique detects every kind of manipulation. However, key challenges include: The focus is on images, not PDFs or legal documents, and many techniques weaken after recompression.

2.8. Mistral 7B [8]

The key objectives of the study include: providing a compact open language model with a good balance of quality and efficiency. The technology used in the study includes: A 7-billion-parameter transformer with grouped-query attention for faster inference and sliding-window attention for longer inputs at lower cost. The methodology follows: The base model is trained as a general-purpose model, and an instruction-tuned variant is released for chat use. The study demonstrates efficiency: It is reported to outperform the larger Llama 2 13B model on the benchmarks evaluated, and its size allows it to run on local hardware. However, key challenges include: It is not specialized for law, can still hallucinate, and needs local compute and a running model service.

2.9. Comparison of the Reviewed Studies

Table 1 on the next page compares the eight studies. Taken together, they show that grounded generation [1], semantic search [2], legal adaptation [3], and image-forensics heuristics [6] each exist as separate tools, and that hallucination in legal answers is a measured problem [5]. None of the reviewed studies combines cited legal answering, document drafting, and file screening in one private, locally run service. Data analytics, intelligent applications, and emerging technologies: The survey of driver-monitoring systems using computer vision [17] demonstrates the potential of visual intelligence for improving road safety and identifying driver-related risks, while the textbook on Python for data analytics [31] supports the application of programming and analytical techniques to data-driven problem-solving. Machine-learning methods for malware detection [22] illustrate how predictive models can strengthen threat identification, and comparative implementations of password-authentication techniques [23] emphasize the importance of evaluating security mechanisms across programming environments. Research on quantum computing for supply logistics [24] highlights the potential of emerging computational paradigms to optimize complex logistics and supply-chain problems. Beyond cybersecurity, the survey of confidential-data protection in distributed cloud storage [25] reinforces the significance of privacy and data integrity, while soft clustering for gene-expression data [26] demonstrates the use of computational intelligence in biomedical analysis. Digital health and medical-tourism innovations [27] illustrate how digital technologies can support connected healthcare services, and research on ethics and regulations in AI-driven systems [28] underscores the importance of responsible AI deployment, governance, and patient protection.

Table 1: Comparison of the reviewed studies

Ref.	Objective	Technology	Methodology	Efficiency	Issues
[1] Lewis et al., 2020	Ground text generation in retrieved evidence; update knowledge without retraining	BART generator with dense retriever over Wikipedia	Retrieve top passages, condition generator on query and passages	Strong open-domain QA results; more factual output	General-domain index; no legal sources or tools
[2] Reimers and Gurevych, 2019	Comparable sentence embeddings for fast semantic search	Siamese and triplet BERT networks	Fine-tune on sentence pairs; compare vectors by cosine	About 65 hours reduced to about 5 seconds for 10,000 sentences	General-domain; not tuned to statutes
[3] Chalkidis et al., 2020	Test legal-domain pre-training of BERT	BERT variants pre-trained on legal text, plus small versions	Pre-train on legislation, cases, contracts; fine-tune on legal tasks	Generally better than general BERT, notably on hard tasks	Encoder only; English; jurisdiction-bound

Ref.	Objective	Technology	Methodology	Efficiency	Issues
[4] Zhong et al., 2020	Survey legal AI tasks and challenges	Embedding-based and symbol-based methods	Task-wise review of judgment prediction, case matching, QA	Structured map of the field	No system; interpretability and ethics open
[5] Dahl et al., 2024	Measure legal hallucination of LLMs	General-purpose LLMs without retrieval	Verifiable queries on federal cases; classify errors	Hallucination is common (roughly 69–88% reported)	One jurisdiction; no retrieval tested; no fix offered
[6] Krawetz, 2007	Quick visual detection of edited image regions	Error Level Analysis on JPEG	Recompress, subtract, amplify, inspect	Simple, fast, no training data	Heuristic; content and compression dependent
[7] Farid, 2009	Survey of image-forgery detection	Pixel, format, camera, physics, geometry techniques	Group methods by the trace each exploit	Clear taxonomy; no universal method	Images only; weak after recompression
[8] Jiang et al., 2023	Compact open model with good quality per size	7B transformer, grouped-query and sliding-window attention	General pre-training; instruction-tuned variant	Reported to beat Llama 2 13B on evaluated benchmarks	Not legal-specific; can hallucinate; needs local compute

III. RESEARCH GAP AND PROBLEM STATEMENT

The review points to four gaps that the prototype is built around. G1, fragmented tools: legal search, drafting and file checks are usually separate tools, so users move data by hand between them. G2, ungrounded answers: plain LLM output has no traceable source [5], while legal use needs the source to be visible. G3, privacy and cost: hosted LLM services send sensitive legal text off-site; local models are rarely packaged for this use [8]. G4, unvalidated forensic thresholds: ELA and metadata checks are often shown as demonstrations, and their thresholds are rarely calibrated on labelled documents [6, 7]. Problem statement.

To design and critically assess a modular, locally deployable prototype that offers (i) source-grounded legal question answering over a local document collection, (ii) automated generation of a standard NDA from structured inputs, and (iii) preliminary PDF-metadata and image-ELA screening through one API, while identifying the accuracy, reliability and security controls needed before responsible use.

Objectives.

O1. Build a RAG pipeline that answers legal queries from an indexed local law collection and returns the source of each answer.

O2. Automate NDA drafting by filling a Word template with party names and an effective date.

O3. Screen PDFs and images for likely tampering with a metadata check and ELA.

O4. Expose all three functions through one documented FastAPI service covered by tests.

O5. Identify reliability and security gaps and a plan to evaluate them.

IV. PROPOSED SYSTEM DESIGN

The scientific rationale is to answer legal questions from retrieved source text instead of model memory. The legal corpus is split into overlapping chunks, each chunk is converted into an embedding, and the embeddings are stored in a vector database. At query time, the question is embedded, the three closest chunks are retrieved and passed to a locally served Mistral model with an instruction to use the context and to say so when it does not know. The reply carries the answer and the source names of the retrieved chunks. The service is built with FastAPI [9], the vectors are

stored in Chroma [10], the model is served by Ollama [11], and PDF text and metadata are read with PyMuPDF [12]. The same service offers two further functions. The NDA generator fills a reusable Word template with two party names and a date. The forensic detector reads PDF metadata and flags a short list of editing tools, and computes an ELA score for JPG and PNG images. All three are exposed as HTTP endpoints, so any front end can use them. The outputs are informational: they support a human decision and do not replace legal advice or a forensic examiner.

V. ARCHITECTURE DIAGRAM

Figure 1 shows the layered design. A client calls the FastAPI application in main.py, which dispatches the request to one of three modules. A separate ingestion script builds the vector store used by the legal module. Table 2 lists the endpoints. The review of NIST-recommended cryptographic schemes for the post-quantum era [29] addresses the need to prepare security infrastructures for potential quantum-computing threats.

Table 2: API endpoint summary

Method and route	Request fields	Response
GET /	none	JSON health message
POST /api/v1/legal-advice	form field prompt	JSON: query, advice, source names
POST /api/v1/generate-nda	party_a, party_b, effective_date	DOCX file
POST /api/v1/detect-fake-doc	uploaded file (PDF, JPG, JPEG, PNG)	JSON forensic report

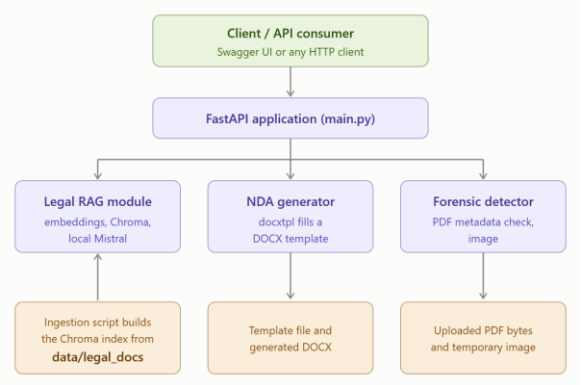


Figure 1: Architecture of the AI Lawyer and Legal Forensics System

Overall, these publications indicate that integrating AI, secure computing, data analytics, and emerging technologies can support more reliable, privacy-aware, and intelligent digital systems.

VI. METHODOLOGY AND ALGORITHMS USED

The system follows a staged workflow.

- Data collection: legal PDF, TXT or Markdown files are placed in data/legal_docs.
- Preprocessing: text is extracted page by page (PDF) or read as UTF-8, empty records are dropped, and the text is split into chunks of 1,000 characters with 150 characters of overlap.
- Embedding and storage: each chunk is embedded with all-MiniLM-L6-v2 and stored in Chroma with its source file name and page number.
- Retrieval and generation: the question is embedded, the top $k = 3$ chunks are retrieved, and Mistral on Ollama writes the answer from them.
- Drafting and screening: the NDA is rendered from a template, and uploaded files are checked by metadata (PDF) or ELA (image).

Retrieval and generation. With f the embedding model and C the set of stored chunks, the retrieved context and the answer are

$$D_k = \operatorname*{top-ksim}_{d \in C}\big(f(q), f(d)\big), \quad a = \mathrm{LLM}(q, D_k), \quad k = 3. \tag{1}$$

Error Level Analysis. For an input image I recompressed as $J = \mathrm{JPEG}_{90}(I)$ with $H \times W$ pixels and C channels,

$$\Delta(x, y, c) = |I(x, y, c) - J(x, y, c)|, \quad s = \frac{255}{\max \Delta} \quad (s = 255 \text{ if } \max \Delta = 0), \tag{2}$$

$$E = \frac{1}{HWC} \sum_{x,y,c} s\Delta(x, y, c), \quad \text{flag if } E > 35.0. \tag{3}$$

PDF metadata check. With T = {photoshop, canva, gimp, ilovepdf, pdfescape, paint}, a PDF is flagged when some $t \in T$ occurs in its lower-cased producer or creator string. This is a triage clue and not proof of editing.

Algorithm 1: Answer a legal query (RAG)	
1:	Receive question q from the API form field.
2:	$e \leftarrow f(q)$; retrieve the top three chunks D_3 of the Chroma store nearest to e.

3:	Build the prompt from the system instruction, the context D_3 and the question q.
4:	$a \leftarrow$ Mistral(prompt) through Ollama.
5:	$S \leftarrow$ the distinct source values of D_3.
6:	return $\{q, a, S\}$.

Algorithm 2: Forgery screening of an uploaded file	
1:	Read the uploaded bytes and the filename extension.
2:	if extension is .pdf then
3:	Read producer, creator, creation date and modification date.
4:	*flag* $\leftarrow$ some $t \in T$ occurs in producer or creator.
5:	else if extension is .jpg, .jpeg or .png then
6:	Compute E by ELA at JPEG quality 90; *flag* $\leftarrow$ ($E > 35.0$).
7:	else
8:	Reject the file with HTTP 400.
9:	end if
10:	return the forensic report.

Table 3: Important implementation parameters

Parameter	Current value	Purpose
Embedding model	sentence-transformers/all-MiniLM-L6-v2	same at ingestion and query time
Retriever k	3	chunks passed to the model
Chunk size and overlap	1,000 and 150 characters	passage length and boundary context
LLM	mistral via Ollama, localhost:11434	local text generation
ELA JPEG quality	90	recompression setting
ELA flag threshold	$E > 35.0$	heuristic flag, not a probability

VII. PROJECT FUNCTIONAL MODULES AND IMPLEMENTATION STATUS

Table 4 gives the status of each module as found in the supplied source archive. It separates what is implemented from what has been measured.

The source review also found points to harden before wider use: the ELA function writes to a fixed temporary file name, which can collide under concurrent requests; uploaded filenames are used in file paths;

Table 4: Functional modules and status

Module	What is implemented	Status and evidence
API (main.py)	four routes with basic validation	implemented
Legal RAG (legal_rag.py)	retriever $k = 3$, local Mistral, source names	implemented; answer quality not measured
Ingestion (ingest_laws.py)	PDF, TXT, MD loading, chunking, indexing	implemented; legal_docs folder is empty and the supplied Chroma store holds no embeddings
NDA generator (doc_generator.py)	template rendering with three fields	implemented; sample NDA produced; legal sufficiency not reviewed
Forensic detector (fake_detector.py)	PDF metadata flag and image ELA score	implemented; first synthetic test in Section VIII
Tests (tests/)	NDA field test, ELA return-type test	two smoke tests only

uploads are read fully into memory without a size limit; and error messages return raw exception text to the client.

VIII. PRELIMINARY RESULTS AND DISCUSSION

The legal question-answering module cannot be evaluated until a law corpus is ingested, so this section reports only the one test that could be run without it: a sanity test of the ELA module on synthetic images. It is a small first check, not a validation on real forgeries.

Setup. Thirty scan-like pages (700×900 pixels, with added scanner noise, saved as JPEG at quality 70) formed the authentic class.
For each, an edited copy was made by whiting out one text line and retyping it, then saving again at quality 70. The ELA logic of the module (Eq. 2–3, quality 90) was applied to all 60 images and the threshold of 35.0 was used as in the API. The script is supplied with this paper.

Table 5: ELA score on synthetic images (30 per class)

Metric	Authentic	Edited	Note
Mean score	9.94	10.21	
Standard deviation	0.54	0.91	
Range	9.03–11.00	7.86–11.96	
Flagged at E > 35.0	0 / 30	0 / 30	default threshold
AUC (edited > authentic)	0.61		0.5 means chance

Discussion. The default threshold of 35.0 flagged none of the 60 images, because scores in both classes sit near 10. The area under the ROC curve of 0.61 is only slightly above chance, so the mean score hardly separates edited from authentic images. A likely reason is that the score is the image-wide mean of a map scaled by its own maximum, which hides small local edits. Block-wise ELA with a heat-map, and a threshold calibrated on labelled genuine and forged scans, are the natural next steps. These results come from synthetic data, a single type of edit and 30 images per class, so they show a weakness of the current threshold and do not measure real-world accuracy.

IX. CONTRIBUTION AND FINDINGS

The contribution of this work is twofold. First, it reviews eight studies and shows that retrieval-grounded generation, semantic search, legal-domain adaptation and image-forensics heuristics are each established, while a private, locally run service that joins cited legal answering, NDA drafting and file screening is not covered. Second, it presents a modular prototype that does this, together with a status table that separates what is implemented from what is measured.

The main findings are as follows. (i) Hallucination in legal answers is a documented problem [5], which supports the design choice of answering from retrieved sources. (ii) The prototype's retrieval and generation pipeline is implemented, but its quality is unmeasured because the corpus is empty. (iii) The default ELA threshold did not flag any image in the first synthetic test and separated the classes only weakly. (iv) Several security and robustness points need work before use beyond a controlled demonstration.

X. CONCLUSION AND FUTURE SCOPE

AI Lawyer combines retrieval-augmented legal question answering, template-based NDA generation, and preliminary document forensics in one FastAPI service that runs locally. The literature supports each design choice but also shows the limits: retrieved context reduces but does not remove unsupported answers, templates need legal review, and forensic indicators are triage clues. The prototype is therefore a proof of concept that should assist qualified users and not replace legal counsel or a forensic examiner. Future scope. Ingest an official statute set with jurisdiction and effective-date tags, and evaluate retrieval (Recall@*k*, Mean Reciprocal Rank) and answer faithfulness with expert review. Return passage-level citations and quoted excerpts, and make the system say so when no relevant source is found. Combine dense retrieval with keyword search for exact section numbers and rare terms. Replace the image-wide ELA mean with block-wise heat maps and calibrate thresholds on labelled genuine and forged documents; report sensitivity and specificity. Harden the service: unique temporary files, sanitized filenames, upload size limits, generic error messages, authentication and logging. Add attorney-reviewed templates, input validation and a visible notice that outputs are not legal advice.

REFERENCES

[1] P. Lewis *et al*., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.

[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992.

[3] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of EMNLP*, 2020, pp. 2898–2904.

[4] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, "How does NLP benefit the legal system: A summary of legal artificial intelligence," in *Proc. ACL*, 2020, pp. 5218–5230.

[5] M. Dahl, V. Magesh, M. Suzgun, and D. E. Ho, "Large legal fictions: Profiling legal

hallucinations in large language models," Journal of Legal Analysis, vol. 16, no. 1, pp. 64–93, 2024.

[6] N. Krawetz, "A picture's worth: Digital image analysis and forensics," presented at Black Hat USA, 2007.

[7] H. Farid, "Image forgery detection," IEEE Signal Processing Magazine, vol. 26, no. 2, pp. 16–25, Mar. 2009.

[8] A. Q. Jiang et al., "Mistral 7B," arXiv preprint arXiv:2310.06825, 2023.

[9] FastAPI, "FastAPI documentation." [Online]. Available: https://fastapi.tiangolo.com/. Accessed: Oct. 12, 2026.

[10] Chroma, "Chroma documentation." [Online]. Available: https://docs.trychroma.com/. Accessed: Oct. 12, 2026.

[11] Ollama, "Ollama documentation." [Online]. Available: https://docs.ollama.com/. Accessed: Oct. 12, 2026.

[12] PyMuPDF, "PyMuPDF documentation." [Online]. Available: https://pymupdf.readthedocs.io/. Accessed: Oct. 12, 2026.

[13] G. D. K. Singh and M. K. Jayanthi, Eds., Network Security Attacks and Countermeasures. IGI Global, 2016, doi: 10.4018/978-1-4666-8761-5.

[14] R. M. Balajee, M. K. J. Kannan, and V. Murali Mohan, "Image-based authentication security improvement by randomized selection approach," in Inventive Computation and Information Technologies. Singapore: Springer, 2022, pp. 61–71.

[15] M. K. J. Kannan, N. Aaijaz, K. G. Mani, and V. Tewari, The Future of Innovation and Technology in Education: Trends and Opportunities. S&M Publications, 2025.

[16] B. R. M. and J. K. M. K., "Intrusion detection on AWS cloud through hybrid deep learning algorithm," Electronics, vol. 12, Art. no. 1423, 2023, doi: 10.3390/electronics12061423.

[17] K. L. S. Kumar and M. K. J. Kannan, "A survey on driver monitoring system using computer vision techniques," in Innovative Computing and Communications, A. E. Hassanien, S. Anand, A. Jaiswal, and P. Kumar, Eds., Lecture Notes in Networks and Systems, vol. 1021. Singapore: Springer, 2024, doi: 10.1007/978-981-97-3591-4_21.

[18] M. K. J. Kannan, "A bird's eye view of cybercrimes and free and open-source software's to detoxify cybercrime attacks—an end user perspective," in Proc. 2nd Int. Conf. Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 232–237, doi: 10.1109/Anti-Cybercrime.2017.7905297.

[19] M. K. J. Kannan, S. Kumar, P. T. Kalaivaani, and G. Tripathi, Artificial Intelligence and Blockchain Technology for Human Resource Management. Scientific International Publishing House, 2025.

[20] M. K. Jayanthi, "Strategic planning for information security—DID mechanism to befriend the cyber criminals to assure cyber freedom," in Proc. 2nd Int. Conf. Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 142–147, doi: 10.1109/Anti-Cybercrime.2017.7905280.

[21] B. R. M., M. K. J. Kannan, and V. Murali Mohan, "Automatic content creation mechanism and rearranging technique to improve cloud storage space," in Inventive Computation and Information Technologies, S. Smys, V. E. Balas, and R. Palanisamy, Eds., Lecture Notes in Networks and Systems, vol. 336. Singapore: Springer, 2022, doi: 10.1007/978-981-16-6723-7_7.

[22] P. Jain, I. Rajvaidya, K. K. Sah, and J. Kannan, "Machine learning techniques for malware detection—a research review," in Proc. IEEE Int. Students' Conf. Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2022, pp. 1–6, doi: 10.1109/SCEECS54111.2022.9740918.

[23] B. R. M., V. M. M., and M. K. J. Kannan, "Performance analysis of bag of password authentication using Python, Java and PHP implementation," in Proc. 6th Int. Conf. Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 1032–1039, doi: 10.1109/ICCES51350.2021.9489233.

[24] M. K. J. Kannan and T. R. Shree Nee, "Qubits unveiled: A deep dive into quantum computing and its revolutionary potential for supply logistics," in Qubits Unveiled: Quantum Computing Solutions for Efficient Supply Logistics, P. Gaba, A. Panwar, V. Jain, and R. Kannan, Eds. Nova Science Publishers, 2025, pp. 273–293, doi: 10.52305/WSXW8884.

[25] H. Naik and M. K. J. Kannan, "A survey on

protecting confidential data over distributed storage in cloud," SSRN, Dec. 1, 2020. [Online]. Available: https://ssrn.com/abstract=3740465.

[26] E. Kavitha, R. Tamilarasan, A. Baladhandapani, and M. K. J. Kannan, "A novel soft clustering approach for gene expression data," Computer Systems Science and Engineering, vol. 43, no. 3, pp. 871–886, 2022, doi: 10.32604/csse.2022.021215.

[27] T. R. Shree Nee, M. K. J. Kannan, and K. Mariyappan, "Digital health and medical tourism innovations for digitally enabled care for future medicine: The real-time project's success stories," in Navigating Innovations and Challenges in Travel Medicine and Digital Health. IGI Global Scientific Publishing, 2025, pp. 325–344, doi: 10.4018/979-8-3693-8774-0.ch016.

[28] M. K. J. Kannan, S. N. T. Ramesh, and K. Mariyappan, "Ethics and regulations in AI-driven ophthalmology," in Generative Artificial Intelligence in Ophthalmology, B. K. Mishra, A. Kumar, K. Mariyappan, V. Tiwari, P. S. Rathore, and G. H. Das, Eds., ch. 34. Scrivener Publishing, 2026.

[29] K. Mariyappan, B. K. Mishra, S. A. Harini, and M. K. J. Kannan, "A review on NIST-recommended cryptographic schemes for the post-quantum computer world era," in *Proc. 8th EAI Int. Conf. Big Data Innovation for Sustainable Cognitive Computing (BDCC 2025)*. Springer Nature, 2025.

[30] R. M. Balajee, M. K. J. Kannan, and V. Murali Mohan, "Web design focusing on users viewing experience with respect to static and dynamic nature of websites," in *Inventive Computation and Information Technologies*, S. Smys, V. E. Balas, and R. Palanisamy, Eds., Lecture Notes in Networks and Systems, vol. 336. Singapore: Springer, 2022, doi: 10.1007/978-981-16-6723-7_5.

[31] S. K. Shukla, U. Dwivedi, M. K. J. Kannan, and C. Sarvani, *Python for Data Analytics: Practical Techniques and Applications*. JSR Publications, 2024.