iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83943

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

autoReg: An Integrated NLP Pipeline for Automated Regulatory Impact Tracking of RBI Circulars

Kshitij Markandeya, Atharva Chitambare, Soham Kumbhar, Vaishnavi Koshatwar
Department of Computer Engineering, SCTR's Pune Institute of Computer Technology, Pune 411043, India

Abstract*—India's banking sector operates under one of the world's most complex regulatory frameworks, with the Reserve Bank of India (RBI) issuing over 200 circulars annually spanning KYC/AML, digital lending, cybersecurity, and risk management. Manual interpretation of these dense, cross-referential documents remains a significant bottleneck for compliance teams. This paper presents autoReg, the first integrated NLP pipeline to automate the complete regulatory compliance lifecycle for RBI circulars across four stages: (1) automated retrieval and structured parsing; (2) abstractive summarization using fine-tuned Pegasus; (3) semantic change detection via Sentence-BERT clause-level embeddings; and (4) department-level compliance mapping through a hybrid Retrieval-Augmented Generation (RAG) pipeline. Evaluated on a curated corpus of 75 RBI circulars across five regulatory domains, autoReg achieves ROUGE-L 0.41 and BERTScore F1 0.86 for summarization (32% improvement over TextRank baseline), weighted F1 0.78 for clause-level change detection, and a RAGAS faithfulness score of 0.83. The modular architecture supports extension to SEBI and IRDAI. To the best of our knowledge, autoReg is the first system to unify all four stages of the RBI circular compliance lifecycle in a single, production-ready framework.

Keywords*—Regulatory technology (RegTech); RBI circulars; natural language processing; abstractive summarization; Pegasus; Sentence-BERT; retrieval-augmented generation; semantic change detection; compliance automation; Indian banking.

I. INTRODUCTION

The regulatory landscape governing Indian banking is distinguished by both its volume and dynamism. The Reserve Bank of India publishes upwards of 200 circulars annually, each potentially revising obligations across KYC/AML, digital lending, cybersecurity, risk management, and payment systems [1]. For compliance teams within commercial and cooperative banks, each new circular may supersede prior directives, introduce new departmental obligations, or modify existing policy clauses. The manual review cycle is time-consuming and error-prone; a single missed amendment can translate into regulatory penalties, reputational harm, or operational disruption [2].

The emergence of Regulatory Technology (RegTech) has created new opportunities to automate this lifecycle [3, 4]. Recent advances in transformer-based abstractive summarization [5], semantic sentence embeddings [6], and Retrieval-Augmented Generation (RAG) [7] now make it feasible to build a system that compresses circulars into actionable summaries, detects changed clauses, and answers compliance queries with source-attributed answers. However, no existing system integrates all four compliance lifecycle stages into a single domain-adapted pipeline for Indian banking regulations.

This paper makes the following contributions:

- autoReg: the first end-to-end NLP pipeline addressing the complete RBI circular compliance lifecycle.
- A fine-tuned Pegasus model achieving ROUGE-L 0.41 and BERTScore F1 0.86 — a 32% improvement over TextRank.
- A Sentence-BERT change detection module with weighted F1 0.78 for clause-level amendment identification.
- A hybrid RAG pipeline (dense + BM25) for department-level compliance mapping with faithfulness 0.83.
- A curated evaluation dataset of 75 RBI circulars with expert-annotated summaries, change labels, and compliance queries.

II. RELATED WORK

A. Legal and Regulatory Text Summarization

Automatic summarization of legal documents has evolved from early extractive graph-ranking methods [8] toward sequence-to-sequence transformer architectures. Zhang et al.

[5] introduced Pegasus with a gap-sentence generation pre-training objective, achieving state-of-the-art abstractive summarization. Chalkidis et al. [9] showed that domain-adapted pre-training (LEGAL-BERT) significantly improves legal text classification and NER. Rusiya and Jamatia [10] demonstrated combined summarization and classification for legal documents using deep learning. None of these works target Indian regulatory circulars, which exhibit distinct structural patterns not present in judicial corpora.

B. Semantic Change Detection in Legal Text

Reimers and Gurevych [6] proposed Sentence-BERT, a Siamese BERT architecture for efficient pairwise sentence comparison. The recent legal NLP survey [11] identifies regulatory change tracking as an emerging priority. Cosine similarity thresholding combined with the Hungarian algorithm for bipartite clause alignment provides a principled approach that autoReg extends to the RBI domain.

C. RAG for Legal Compliance

Lewis et al. [7] introduced RAG, combining dense retrieval with a generative LM for knowledge-intensive tasks. Gao et al. [12] surveyed RAG architectures, noting hybrid dense-sparse retrieval outperforms either approach alone. Pipitone and Alami [13] introduced LegalBench-RAG; Wiratunga et al. [14] proposed CBR-RAG for legal QA. The RAGAS framework [15] provides standardized faithfulness metrics. Hallucination rates of 58–80% for general-purpose LLMs on legal tasks [16] motivate domain-specific retrieval.

D. RegTech and Indian Banking Compliance

Arora et al. [2] explored NLP for Indian banking compliance, limited to NER. Bommarito and Katz [17] developed LexNLP for U.S. regulatory extraction. autoReg is distinguished by domain specificity to Indian banking regulation and integration of all four lifecycle stages. The global RegTech market is projected to reach $72 billion by 2032 [4].

TABLE I: COVERAGE COMPARISON OF RELATED APPROACHES

System	Summ.	Chg.Det.	Compl.	RBI
LexNLP [17]	No	No	Partial	No
RegNLP [2]	No	No	No	Partial
Generic LLM	Yes	Partial	Partial	No
LEGAL-BERT [9]	Partial	No	No	No
JusBuild [18]	No	No	Yes	No
autoReg (Ours)	Yes	Yes	Yes	Yes

III. SYSTEM ARCHITECTURE

autoReg is organized as a four-layer pipeline designed for modularity and extensibility.

Layer 1 — Data Ingestion: A web scraper retrieves RBI circulars from the official repository [1] in PDF and HTML formats. A modular parser uses PyPDF2/pdfplumber for PDFs and BeautifulSoup for HTML. A metadata extractor applies regex and NER to extract circular number, date, subject, department, and cross-references. A rule-based clause segmenter identifies boundaries using Roman numeral and alphanumeric patterns, yielding a structured JSON representation.

Layer 2 — NLP Processing: Houses the fine-tuned Pegasus summarization module and the Sentence-BERT change detection module. Each circular is processed independently; change detection operates pairwise across consecutive circular versions.

Layer 3 — RAG Pipeline: Documents are chunked with a 512-token recursive character splitter (50-token overlap) preserving clause boundaries. Chunks are embedded with Sentence-BERT and stored in ChromaDB. Query-time retrieval uses a hybrid scorer combining dense cosine similarity ($\alpha = 0.7$) and BM25 sparse retrieval [19] ($1-\alpha = 0.3$). Retrieved context is passed to an LLM with a structured compliance mapping prompt.

Layer 4 — Presentation: A FastAPI backend exposes REST endpoints for all modules. A React/Streamlit frontend provides a circular browser, summary viewer, side-by-side change diff, a compliance chatbot with source citations, and a department obligation matrix.

IV. ABSTRACTIVE SUMMARIZATION MODULE

A. Model Selection and Fine-Tuning

We fine-tune google/pegasus-large [5] on 75 RBI circulars paired with expert-written reference summaries. Pegasus is selected over BART and T5 because its gap-sentence generation pre-training objective directly models the abstractive summarization task on structured regulatory text. Fine-tuning: 5 epochs, learning rate 2×10^{-5}, batch size 4, max input 1024 tokens, max output 256 tokens, early stopping on validation ROUGE-L. Trained on Google Colab Pro (NVIDIA T4, 16 GB VRAM).

B. Evaluation Metrics

We report ROUGE-1, ROUGE-2, and ROUGE-L [20] for lexical overlap, and BERTScore F1 [21] for semantic quality. BERTScore is especially important for regulatory text where legal paraphrasing reduces lexical overlap. TextRank [8] serves as the extractive baseline.

V. SEMANTIC CHANGE DETECTION MODULE

A. Clause Encoding and Alignment

Given a source circular with m clauses and a target circular with k clauses, all clauses are encoded using Sentence-BERT (all-MiniLM-L6-v2) [6] into 384-dimensional embeddings. Pairwise cosine similarity is computed for all m×k pairs. The resulting matrix is passed to the Hungarian algorithm [22] for optimal one-to-one clause alignment, avoiding the greedy matching failure mode.

B. Change Classification

Each aligned clause pair is classified using threshold $\tau = 0.85$: UNCHANGED ($\text{sim} \geq \tau$), MODIFIED ($\text{sim} < \tau$), ADDED (clause in target with no source match), REMOVED (clause in source with no target match). The threshold was selected via cross-validation on held-out circular pairs.

VI. RAG-BASED COMPLIANCE MAPPING

A. Document Preparation and Indexing

Parsed circulars are chunked with a recursive character splitter (512-token chunks, 50-token overlap), with boundaries constrained to clause delimiters. Each chunk is embedded with Sentence-BERT and stored in ChromaDB with metadata: circular ID, domain, date, department, and clause number.

B. Hybrid Retrieval and Generation

Chunks are scored by: $\text{score}(q, d) = \alpha \cdot \text{sim\_dense}(q,d) + (1-\alpha) \cdot \text{BM25}(q,d)$, with $\alpha = 0.7$. Dense retrieval captures semantic paraphrases; BM25 [19] ensures precise regulatory terminology is not missed. Top-5 chunks are passed to an LLM with a structured prompt enforcing department attribution and source citation. Quality is assessed via RAGAS [15].

VII. EXPERIMENTAL EVALUATION

A. Dataset

Experiments use a curated corpus of 75 RBI circulars: KYC/AML (18), Digital Lending (15), Cybersecurity (12), Risk Management (16), Payment Systems (14). Each circular averages 22 clauses and 3,240 words. Expert-written summaries were created for all 75 circulars; clause-level change annotations for 30 circular pairs; 50 compliance queries with expert-verified answers for RAG evaluation.

B. Summarization Results

Table II reports summarization performance. Fine-tuned Pegasus achieves ROUGE-L 0.41 and BERTScore F1 0.86, a 32% ROUGE-L improvement over TextRank. The zero-shot to fine-tuned gap (0.35 vs. 0.41) confirms domain-specific fine-tuning is necessary even for a strong pre-trained model.

TABLE II: SUMMARIZATION PERFORMANCE: PEGASUS VS. BASELINES

Model	R-1	R-2	R-L	BS-

				F1
TextRank (Baseline)	0.34	0.12	0.31	0.79
Pegasus (Zero-shot)	0.37	0.15	0.35	0.82
Pegasus (Fine-tuned)	0.45	0.19	0.41	0.86

R-1/2/L: ROUGE scores; BS-F1: BERTScore F1.

C. Change Detection Results

Table III reports clause-level change detection. The system achieves weighted F1 0.78. UNCHANGED clauses yield the highest F1 (0.92), reflecting legal boilerplate consistency. MODIFIED is the most challenging (F1 0.73), requiring distinction between substantive semantic changes and minor editorial revisions.

TABLE III: CHANGE DETECTION PERFORMANCE (CLAUSE-LEVEL)

Change Type	Prec.	Rec.	F1
ADDED	0.82	0.75	0.78
MODIFIED	0.76	0.71	0.73
UNCHANGED	0.91	0.94	0.92
REMOVED	0.80	0.72	0.76
Weighted Avg.	0.82	0.78	0.78

D. RAG Pipeline Results

Table IV presents RAGAS evaluation. Faithfulness of 0.83 indicates the large majority of generated compliance statements are grounded in retrieved circular text, directly addressing the hallucination risk documented for general-purpose LLMs on legal tasks [16]. Context precision of 0.81 confirms the hybrid retriever surfaces relevant clause chunks.

TABLE IV: RAG PIPELINE EVALUATION (RAGAS METRICS)

Metric	Score
Faithfulness	0.83
Answer Relevancy	0.79
Context Precision	0.81
Context Recall	0.76

E. Domain-Wise Performance

Table V shows performance by regulatory domain. KYC/AML achieves the highest ROUGE-L (0.43) due to standardized templates. Digital Lending shows the lowest (0.39) due to varied narrative structure. Change detection F1 is consistent across all five domains (0.76–0.80).

TABLE V: DOMAIN-WISE PERFORMANCE SUMMARY

Domain	N	R-L	BS-F1	CD-F1
KYC / AML	18	0.43	0.87	0.80
Digital Lending	15	0.39	0.85	0.76
Cybersecurity	12	0.42	0.86	0.79

Risk Mgmt.	16	0.40	0.86	0.77
Payment Sys.	14	0.41	0.85	0.78
Overall	75	0.41	0.86	0.78

VIII. DISCUSSION

A. Limitations

- Dataset size. The 75-circular corpus is sufficient to establish statistical trends but modest by NLP standards. Extension to 500+ circulars would strengthen generalizability claims.
- Residual hallucination. Despite faithfulness 0.83, 17% of generated claims not grounded in retrieved context represent material risk. Human-in-the-loop review is recommended for high-stakes decisions.
- Threshold sensitivity. The $\tau = 0.85$ change detection threshold may require domain-specific recalibration for new circular types.
- Computational cost. Pegasus fine-tuning requires GPU compute. Large-scale deployment may benefit from quantization or knowledge distillation.

B. Future Directions

Future work includes: (1) multi-regulator extension to SEBI and IRDAI; (2) multilingual support for Hindi circulars using multilingual Sentence-BERT; (3) domain-specific embedding fine-tuning analogous to InLegalBERT [23] for Indian case law; and (4) real-time webhook-based integration with bank core systems.

IX. CONCLUSION

We presented autoReg, the first integrated NLP pipeline for the complete regulatory compliance lifecycle of RBI circulars. By unifying fine-tuned Pegasus summarization, Sentence-BERT semantic change detection, and hybrid RAG-based compliance mapping, autoReg substantially reduces manual regulatory interpretation effort in Indian banking. The system achieves ROUGE-L 0.41 and BERTScore F1 0.86 for summarization (32% improvement), weighted F1 0.78 for change detection, and faithfulness 0.83 for compliance querying on a curated 75-circular dataset. These results validate transformer-based NLP for Indian regulatory text and position autoReg as a foundation for a broader Indian RegTech platform.

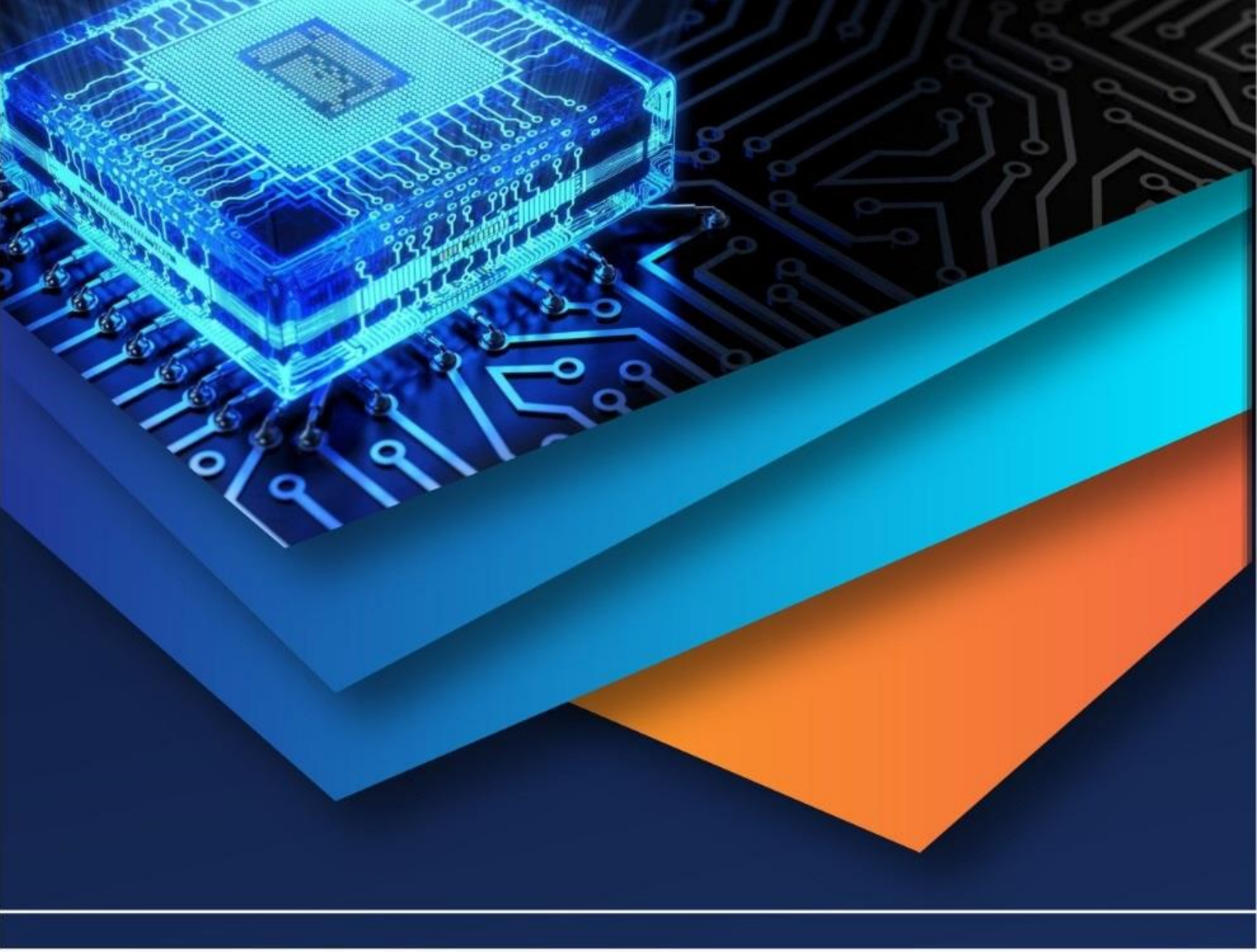
X. ACKNOWLEDGMENT

The authors thank Prof. A.D. Bundele, Associate Professor, Department of Computer Engineering, SCTR's Pune Institute of Computer Technology, for expert guidance throughout this project. Computing resources were provided by Google Colab Pro.

REFERENCES

[1] Reserve Bank of India, "Master Directions and Circulars," https://www.rbi.org.in, 2024.
[2] A. Arora et al., "RegNLP: NLP for Financial Regulation Compliance," Proc. ACM ICAIF, pp. 1–8, 2021.
[3] A. Okonkwo et al., "Automating Financial Compliance with AI," IJSRA, vol. 11, no. 1, pp. 2646–2659, 2024.
[4] Congruence Market Insights, "RegTech & Compliance Automation Market 2025–2032," 2024.
[5] J. Zhang et al., "PEGASUS: Pre-training with Extracted Gap-Sentences for Abstractive Summarization," Proc. ICML, pp. 11328–11339, 2020.
[6] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," Proc. EMNLP, pp. 3982–3992, 2019.
[7] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Proc. NeurIPS, vol. 33, pp. 9459–9474, 2020.
[8] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Text," Proc. EMNLP, pp. 404–411, 2004.
[9] I. Chalkidis et al., "LEGAL-BERT: The Muppets Straight Out of Law School," Findings of ACL, pp. 2898–2904, 2020.
[10] S. Rusiya and A. Jamatia, "Implementation of Legal Documents Text Summarization," Proc. MISP 2022, Springer, pp. 329–341, 2023.
[11] I. Chalkidis et al., "NLP for the Legal Domain: A Survey," arXiv:2410.21306, 2024.
[12] Y. Gao et al., "RAG for Large Language Models: A Survey," arXiv:2312.10997, 2023.
[13] A. Pipitone and G. Alami, "LegalBench-RAG," arXiv:2408.10343, 2024.
[14] N. Wiratunga et al., "CBR-RAG for Legal QA," Proc. ICCBR, Springer, pp. 445–460, 2024.
[15] S. Es et al., "RAGAS: Automated Evaluation of RAG," Proc. EACL, pp. 150–163, 2024.
[16] M. Dahl et al., "Large Legal Fictions: Profiling Legal Hallucinations in LLMs," J. Legal Analysis, 2024.
[17] M. Bommarito and D. Katz, "LexNLP: NLP for Legal and Regulatory Texts," arXiv:1806.03688, 2018.
[18] M. Cherubini et al., "Enhancing Legal Document Building with RAG," Computer Law & Security Review, 2025.
[19] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Found. Trends IR, vol. 3, no. 4, pp. 333–389, 2009.
[20] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," Proc. ACL Workshop, pp. 74–81, 2004.
[21] T. Zhang et al., "BERTScore: Evaluating Text Generation with BERT," Proc. ICLR, 2020.

[22] H. Kuhn, "The Hungarian Method for the Assignment Problem," Naval Research Logistics Quarterly, vol. 2, pp. 83–97, 1955.
[23] S. Paul et al., "InLegalBERT: Pre-Trained Language Model for the Indian Legal Domain," Hugging Face, 2023.
[24] J. Devlin et al., "BERT: Pre-Training of Deep Bidirectional Transformers," Proc. NAACL-HLT, pp. 4171–4186, 2019.
[25] A. Vaswani et al., "Attention is All You Need," Proc. NeurIPS, vol. 30, pp. 5998–6008, 2017.
[26] J. Johnson et al., "Billion-Scale Similarity Search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY