iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78010

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 III Mar 2026- Available at www.ijraset.com

Cleansera: A Context-Aware, Algorithm-Centric Data Cleaning System with RAG-Enhanced Intelligence

Kanchan Dhomse[1], Shubham Deshmukh[2], Vishal Daware[3], Shubham Rao[4], Om Bhise[5]
[1]*Professor, Department of Information Technology, MET's Institute of Engineering, Nashik, Maharashtra, India*
[2, 3, 4, 5]*Student, Department of Information Technology, MET's Institute of Engineering, Nashik, Maharashtra, India*

Abstract: ***The two most difficult tasks of data analytics are usually data preparation and data cleaning; according to the research presented in this article, these activities account for approximately 50 to 80 percent of total time spent on real-world data analytic initiatives [1]. Most existing data cleaning tools use static, rule-based approaches and do not accommodate the unique needs of a specific domain, nor do they provide clear visibility of the internal ways in which they arrive at their final data-cleaned outputs [1]. This study introduces Cleansera, a context-aware, AI based data cleaning solution that places a greater emphasis on algorithmic characterization and uses flowchart-driven methodologies to achieve execution. Cleansera offers automated context-detecting capability, Retrieval Augmented Generation (RAG), deterministic data cleaning workflows, version-controlled user interaction, and dual-checkpoint data cleaning QA capabilities [1][2]. The primary focus of this article is on the latest designed algorithms and validated workflows created during partial implementation of the Cleansera system. Through the creation of defined algorithms, execution paths, and verification checkpoints, Cleansera offers transparency, auditability, and repeatability for automated data cleaning [1]. Cleansera combines the elements of AI-driven flexibility with the principles of traditional algorithms to create a data cleaning methodology that may be adopted in both industry and academia [1][2].***

Keywords: ***Data Cleaning, Context-Aware Systems, Algorithm Design, Flowchart Methodologies, Retrieval-Augmented Generation, Data Quality Assurance, Explainable AI, Deterministic Processing, Semantic Analysis, Version Control.***

I. INTRODUCTION

All businesses and industries rely on data-driven systems to run their operations. From finance to healthcare to supply chain management to digital marketing, data-driven systems are the foundation for all industries [4]. Although advanced analytics and machine learning continue to evolve at an incredible pace, their effectiveness depends on the quality of the input data. In most cases, the dataset collected from an operational system is either incomplete, inconsistent, has duplicate entries, or has a semantic ambiguity [1]. As a result, data cleaning is necessary before any analysis can take place; however, data cleaning is one of the most time-consuming phases of the entire data lifecycle [1]. Data professionals provide little opportunity for extracting valuable insights while performing repetitive tasks associated with preprocessing data. Traditional data cleaning tools have attempted to solve this issue by using fixed validation rules and manual configurations, but these approaches typically do not work well with data that crosses multiple business domains [1].

This paper presents Cleansera, a context-aware data cleaning solution designed to model industry-specific semantics and specific business rules explicitly [1]. Unlike traditional systems that treat the cleaning of data as a black-box process, Cleansera is based on deterministic algorithms and flow charts [1]. The development philosophy used in Cleansera emphasizes the creation of a system that is transparent, predictable, and capable of automating the cleaning of data in a controlled manner. This paper will present details of the algorithms and flow charts used to implement the various functions of Cleansera, which represent the most current design results achieved to date through the partial implementation of the system.

II. LITERATURE REVIEW

Recent developments in data cleaning looked into rule-based approaches, statistical profiling methods, and AI supported pre-processing methods. Conventional solutions, like ETL pipelines and cleaning with spreadsheets, have been heavily reliant on guidelines prescribed in advance and on manual engagement, thus making them inefficient for structuring large and/or diverse datasets.

Research conducted into LLM assisted data cleansing revealed that one of their strengths lay in the ability to identify anomalies and suggest changes. Many of these approaches, however, still function as opaque "black boxes" and have a lack of explainability. AutoDCWorkflow and RetClean are two systems that provide the benefits of automation, but still require validation from a human, and do not provide guarantee of deterministic execution. Cleansera is an extension of prior work by including AI capability within the context of explicitly defined algorithmic logic. Unlike generative systems, the execution of all decisions in Cleansera is guided by formally established algorithms and can be validated by flowchart diagrams [1]. After this process, Cleansera's recorded processes will provide assurance of their reproducibility, auditability, and ability to be utilized for regulated industries such as BFSI and manufacturing [1].

III. DESIGN OBJECTIVES

The following goals define the design goals for Cleansera.

1) Context-Sensitive: Automatic identification or acceptance of user-defined industry context and application of industry specific cleaning rules.
2) Algorithmic Transparency: All core operations will be performed using explicit algorithms and processes.
3) Flowchart Driven execution: System behavior will be visually represented in the form of flow charts to facilitate better understanding and verification.
4) Quality Assurance: Quantitative and measurable determination of the effectiveness of cleaning operations using deterministic checkpoints.
5) Partial Automation with Control: Automated AI-assisted cleaning with human oversight and auditability now incorporated.

IV. SYSTEM ARCHITECTURE

Cleansera is built around seven core components that work together to clean and manage your data effectively. Let me walk you through what each one does:

A. Secure Authentication and Session Management

When you log into Cleansera, the system needs to know who you are and keep you securely logged in while you work. This component handles all of that - it verifies your credentials when you sign in, creates a secure session for you, and maintains that session throughout your work. It's like having a security guard who checks your ID at the door and then gives you a badge that proves you belong there. The system continuously validates your session to make sure unauthorized users can't access your data, and it manages when sessions should expire for security purposes.

B. Dataset Ingestion and Profiling

Think of this as the system's initial meeting with your data. When you upload a dataset - whether it's a CSV file, Excel spreadsheet, or database export - this module takes it in and gets to know it intimately. It examines every column, figures out what types of data you have (numbers, text, dates, etc.), counts how many missing values exist, identifies duplicates, and creates a comprehensive profile. It's similar to a doctor performing a complete physical examination before treatment - the system needs to understand what it's working with before it can clean anything.

C. Context Detection Engine

Here's where things get really interesting. This component is the brain that figures out what your data actually represents. It doesn't just see a column of numbers - it recognizes "oh, these are phone numbers" or "these look like product IDs." It identifies patterns, relationships between columns, and the semantic meaning behind your data. For instance, it can tell the difference between a zip code and a random five-digit number, or distinguish between a person's name and a product name. This contextual understanding is crucial because you can't properly clean data if you don't understand what it represents.

D. Context-Aware Cleaning Engine

Once the system understands your data's context, this engine performs the actual cleaning work. But it's not a one-size-fits-all approach - it applies different cleaning strategies based on what the data represents. Email addresses get validated differently than phone numbers. Names are handled with different rules than product codes. If you have missing values, it decides whether to fill them in, leave them blank, or flag them for review based on the column's importance and data type. It's like having a professional organizer who knows that books should be alphabetized but photos should be sorted by date - different content requires different approaches.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

E. Master Field Identification Module

In many datasets, you have columns that uniquely identify each record - like customer IDs, order numbers, or social security numbers. This module's job is to find those critical identifier fields. Why does this matter? Because these fields are extra sensitive - you never want to accidentally delete or modify them during cleaning. The system analyzes your data to spot which columns serve as unique identifiers or primary keys, then flags them as "master fields" that require special handling. It's like marking the load-bearing walls in a house before renovation - you need to know what's holding everything together.

F. Data Loss Detection and Validation Module

This is your safety net. Every time the cleaning process makes changes to your data, this module watches carefully to ensure nothing important gets lost or corrupted. It compares the before and after states, tracking exactly what changed, how many records were affected, and whether any critical information disappeared. If the cleaning delete too much data or make questionable changes, it raises red flags. It generates detailed reports showing you exactly what would happen before committing any changes, giving you the chance to review and approve. Think of it as having a meticulous accountant who reconciles every transaction to make sure the numbers still add up correctly.

G. Version Control and Audit Module

The final component maintains a complete history of everything that happens to your data. Every cleaning operation, every modification, every decision gets logged with timestamps and details about who did what. If you need to roll back changes, you can restore previous versions of your dataset. If you need to prove compliance with data regulations, you have a complete audit trail. It's similar to how Google Docs tracks changes and lets you see the revision history - except for data cleaning operations. This transparency and traceability is essential for both operational needs (fixing mistakes) and regulatory requirements (proving due diligence).

Each module operates with clearly defined inputs and outputs, enabling independent validation and deterministic behavior across executions.

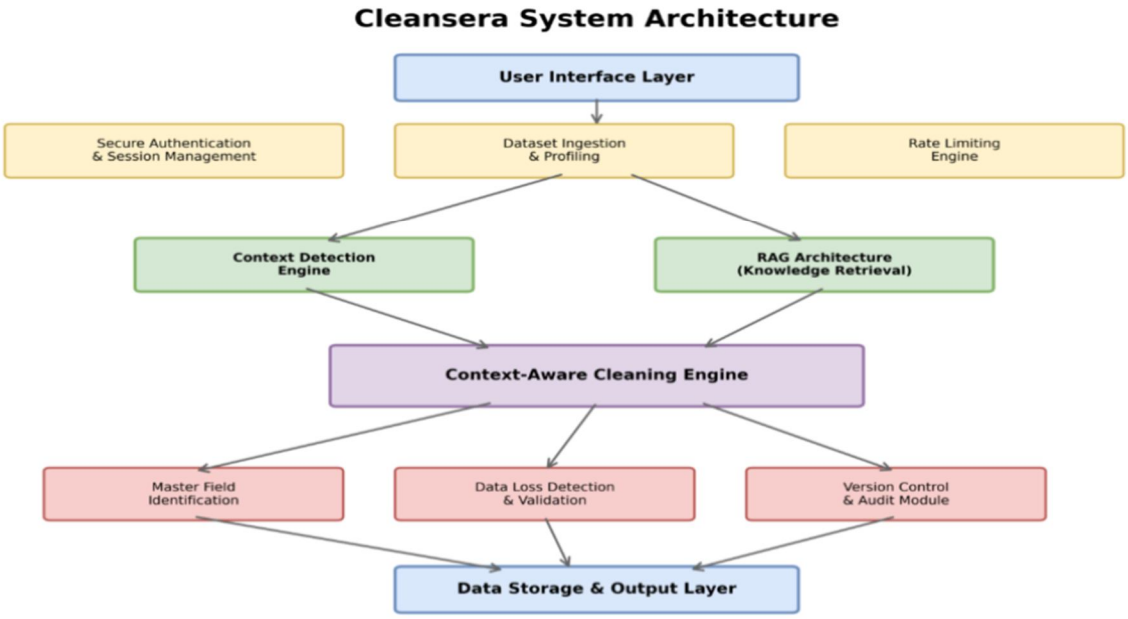


Fig. 1 Detailed Cleansera System Architecture

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

V. CORE ALGORITHMS AND IMPLEMENTATION

A. *Authentication and Session Management*

The authentication system uses bcrypt hashing (cost factor 12) and token-based session management. Upon credential verification:

1) Receive credentials
2) Hash password
3) Validate against stored hash
4) Generate device fingerprint
5) Issue access token (15 min) and refresh token (14 days)
6) Store hashed refresh token in Redis
7) Return secure httpOnly tokens
8) This ensures cryptographic security and token revocation capability.

B. *Rate Limiting Algorithm*

To prevent any system abuse and maintain system functionality, Cleansera uses a Redis-backed Rate limiting algorithm. A Redis-based system allows for a scalable solution by allowing each request to use a unique key based on the client's IP address and the endpoint of the request. Each request will increment the request counter atomically and set the list of valid request times with a Time-To-Live property. Once the request counter exceeds a specified threshold, the request will be rejected with a corresponding HTTP response.

C. *Dataset Profiling and Normalization*

All the datasets that are entered into the system (whether CSVs, Excels or JSON) are normalized into a tabular format. The profiling algorithm calculates various statistics about the data in each column including (but not limited to) the ratio of missing values, the percentage of unique values, and the inferred data type for each column. These profiles are used as input metrics to subsequently detect contexts and to clean up your data.

D. *Context Detection Algorithm*

Context detection is conducted by looking for metadata about your dataset including data characteristics, common use cases to the columns in the dataset, and any patterns in the data. Each potential context (other than the obvious ones) is given a weighted confidence score based on the number of semantic matches and the number of statistical indicators. The chosen context is the highest scoring weighted context, or the lowest scoring context exists, in which case the algorithm will prompt the user to confirm the context manually. This hybrid approach provides both an automated and manual way to maintain control of the data.

E. *Context-Aware Cleaning Engine*

Cleaning pipeline stages: Schema Validation, Duplicate Detection, Missing Value Treatment, Format Standardization, Outlier Handling, Semantic Validation, and Quality Checkpoint Evaluation. All transformations are logged for auditability.

F. *Master Field Identification*

Fields are ranked based on uniqueness score, semantic relevance, referential integrity, and type of consistency. Identified master fields are protected against destructive transformations.

G. *Data Loss Detection and Validation*

The dual checkpoint quality assurance process built into Cleansera includes a data loss detection algorithm that compares the raw data and cleansed data after cleansing is complete [1]. The results from the data loss detection algorithm will assist the user in determining how much record change and attribute change occurred during the cleansing process. In addition, the user will also have access to the record-level and attribute-level quantification of the clean raw data records and the clean cleansed data records. The user will receive a computed measure of the deletion rate and a computed measure of the modification rate [1].

VI. FLOWCHART-DRIVEN EXECUTION MODEL

Cleansera's design principle is that flowcharts should be treated as the main tool used when designing a system (as opposed to only being used as a reference).

Each major algorithm will have a flowchart which represents visually where an algorithm will execute in a time sequence; where decisions (i.e., branches) will be made; how an algorithm will verify its input is valid; and what actions will occur when an algorithm fails to execute properly.

Within the authentication and session management process, there are multiple steps including receiving the user's credentials, verifying those credentials against their password/username, issuing tokens/creating sessions, and verifying devices. In addition, the flowchart has placed conditional branches to process invalid credentials, expired sessions, or tokens that have been revoked. This guarantees that predictable results will occur when security fails.

Within the context detection workflow, the flowchart is used to illustrate the flow of processing metadata (via the extraction of item level information), assigning a rating or score based on the semantics of the metadata, determining if the context is acceptable based on a confidence threshold, and whether a manual override for the context is available. By explicitly modeling low-confidence situations, the Cleansera system will no longer allow for incorrect identification of the type of context that exists for most industries by automated tools.

Cleansera's data cleaning flowchart is modeled after a sequential pipeline and includes steps such as normalizing a dataset to be cleaned; selecting the rule(s) that will be applied to the normal data; applying the determined rule(s) to the clean data set; protecting the master data fields; and logging the results of each transformation. Each step in the transformation process has been blocked from passing errors to the subsequent stages in the transformation process. The final step of the data cleaning flowchart transitions into the quality assurance flowchart to ensure that the process of operating has been validated.

Using flowcharts as a primary method for execution in an AI-assisted decision-making process will allow for all decisions made by AI in a deterministic manner that shall remain within reviewable workflows. This design creates a large improvement in explaining how the AI makes decisions and thus, it will be appropriate for being validated academically and/or working in regulated industries.

VII. CONCLUSION

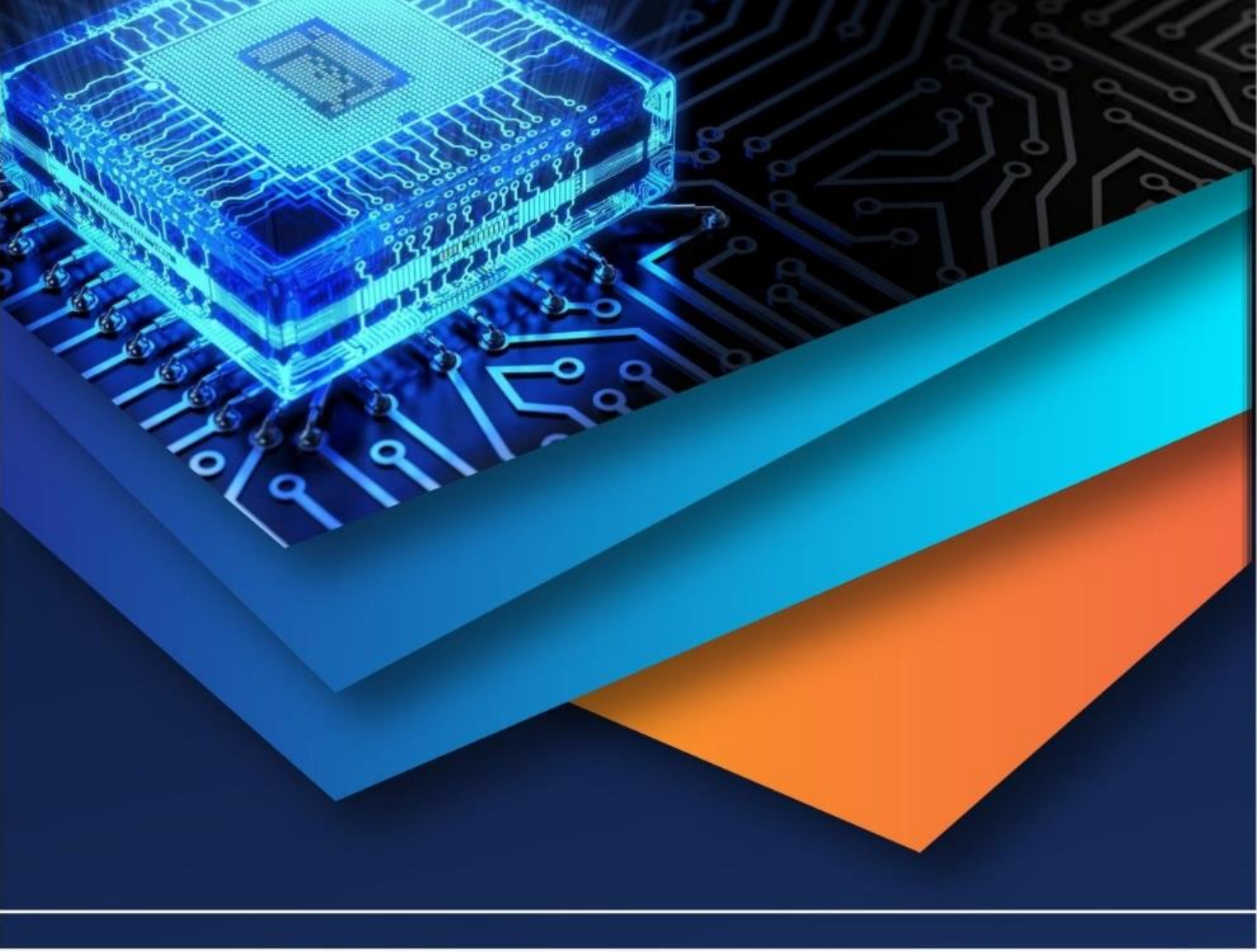
The goal of this paper was to provide a more comprehensive and expanded view of Cleansera's algorithmic foundation. Specifically, through a focus on transparency, flowchart-based execution, and deterministic quality assurance, Cleansera has been designed to overcome some of the major challenges associated with currently available data cleaning systems [1]. Future work on Cleansera will focus on developing full operational implementations, evaluating Cleansera's performance, and determining Cleansera's empirical efficacy across a variety of industry data sets.

VIII. ACKNOWLEDGMENT

We express sincere gratitude to Dr. Kanchan Dhomse and Prof. Kishor Mahale for their invaluable guidance and support. We also thank the Department of Information Technology at MET’s Institute of Engineering, Nashik, for providing resources and infrastructure for this research.

REFERENCES

[1] S. Deshmukh, O. Bhise, S. Rao, and V. Daware, "Cleansera: An intelligent desktop application for domain-specific data cleaning," International Research Journal of Engineering and Technology (IRJET), vol. 12, no. 11, 2025.
[2] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020.
[3] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023.
[4] L. Li et al., "AutoDCWorkflow: LLM-based data cleaning workflow auto-generation and benchmark," arXiv, 2025.
[5] M. Naeem et al., "RetClean: Retrieval-based data cleaning using LLMs and data lakes," arXiv, 2024.
[6] E. Meguellati et al., "Are LLMs good data preprocessors?" arXiv, 2025.
[7] S. Zhang, Z. Huang, and E. Wu, "Data cleaning using large language models," arXiv, 2024.
[8] L. Biester et al., "LLMClean: Context-aware tabular data cleaning via LLM generated OFDs," in Proc. VLDB, 2024.
[9] F. Ahmadi, Y. Mandirali, and Z. Abedjan, "Accelerating the data cleaning systems Raha and Baran through task and data parallelism," in Proc. VLDB Workshop, 2024.
[10] J. Choi et al., "Multi-News+: Cost-efficient dataset cleansing via LLM-based data annotation," in Proc. EMNLP, 2024.
[11] S. Zhang, Z. Huang, and E. Wu, "Cocoon: Data cleaning using LLMs," arXiv, 2024.
[12] W. Ni et al., "IterClean: Iterative data cleaning with LLMs," in Proc. SIGMOD, 2024.
[13] P. Martins et al., "Performance and scalability of data cleaning tools," MDPI Data, 2025.
[14] T. Brown et al., "Language models are few-shot learners," in Proc. NeurIPS, 2020.
[15] A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, 2017.
[16] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers," in Proc. NAACL-HLT, 2019.
[17] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," IEEE Data Engineering Bulletin, vol. 23, no. 4, pp. 3–13, 2000.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY