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 4, November 2026
Impact Factor: 8.2

A Framework for AI-Driven Legal Case Analysis and Grounded Decision Support Based on Integrated RAG and Agentic NLP

Prathmesh Dawkar[1], Ganesh Shejul[2], Sandesh Shinde[3], Dr. Pritesh Patil[4]
[1,2,3,4]Department of Information Technology,
All India Shri Shivaji Memorial Society Institute of Information Technology, Pune, India
[1]dawkarprathmesh251@gmail.com, [2]project.ganeshshejul@gmail.com,
[3]sandeshshinde2026@gmail.com, [4]pritesh.patil@aissmsioit.org

Abstract: *Massive amounts of textual and semi-structured data exist in legal domains in the form of court records, petitions, court judgments, depositions, and gazettes. However, the data suffer from incompleteness, heterogeneity, and fragmentation at different jurisdictions, with predominantly unstructured formats. Preprocessing such legal data for supporting judicious analysis and decision making (i.e., Legal Data Wrangling) is a tedious, time-consuming, and laborious task requiring expert knowledge and skills. Moreover, existing automated methods, including recent advances using general-purpose language models, are limited to performing specific tasks and thus unsuitable for judicious decision making in legal contexts. In this work, we propose a holistic idea and approach for an Automated Legal Case Analysis and decision support system called JuriumLegal++. This system employs an Agentic AI setup where an orchestrating AI agent, which is built upon a large language model (LLM) and a stateful graph (LangGraph), coordinates multiple specialized agents for profiling legal documents, cleaning legal statutes, standardizing legal terms and precedents through retrieval augmented generation (RAG), feature engineering for judicious decisions, validating evidence and generating explainable decision summaries. Document segmentation and similarity transformation were performed using deterministic database operations and vector calculations. If the quality of context alignment and accurate evidence is below a threshold, an in-built quality check forces the system to re-plan its workflow. The proposed methodology was designed and evaluated for an Indian Legal Document Corpus (ILDC) and High Court precedents and evaluated in 5 quality aspects and legal retrieval measures.*

Keywords: legal informatics, data wrangling, large language models, agentic AI, retrieval-augmented generation, citation grounding, decision support systems, explainable AI.

I. INTRODUCTION

Legal data is one of the most information-dense data domains in society. Legal cases, law firms, tribunals, and legal teams in corporations produce a large quantity of data including petitions, evidence, witness statements, court transcripts, and judgements. With appropriate processing, legal data can inform legal decision making, legal risk analysis, compliance, and precedent discovery. However, legal data is not usually easily analyzable. It often contains missing data, duplicate documents, inconsistencies in dates and case numbers, varying terminologies, regional variations, inconsistency in legislation references, and large quantities of messy text extracted via OCR. In particular, the data may have issues relating to: Incomplete data, Duplicate documents, Inconsistencies in date formats and case numbers, Inconsistencies in terminology, Regional differences, Inconsistencies in legislation references, Large amounts of messy text.

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 4, November 2026 Impact Factor: 8.2

Transforming this messy data into clean, standardized, and validated data to be used in machine learning and neural retrieval is commonly referred to as legal data wrangling. Steps in legal data wrangling include document profiling, text cleaning, legislation standardization, text transformation, multi-tenancy, feature extraction, and validation. Within legal practice, data wrangling is considered to be the most time-consuming part of any analytical project. It can involve very long hours of billable work by legal advocates and requires deep legal knowledge. Current data wrangling methods in the legal industry involve manually written regular expressions, fragile statistical scripts, and manual checks by paralegals. These methods are very expensive, are difficult to scale, and quickly become outdated as legal codes change (for example, a recent change in legislation in India from Indian Penal Code to Bharatiya Nyaya Sanhita).

Recent advances in Large Language Models (LLMs) allow for significant automation within data wrangling processes. Recent research has demonstrated that LLMs can be used to effectively automate data wrangling tasks including zero shot/few shot data transformation, data normalization, error detection and code generation [1] [2] [5], as well as legal entity extraction from judgements [4]. However, the majority of research in this field focus on individual data wrangling tasks, do not specifically apply to legal compliance, and are very vulnerable to hallucination by the LLM. As a result, fabricated legislation and legal precedent can be generated. Additionally, no justification for changes made during data wrangling can be provided [3] [6]. To solve these problems, this paper proposes a novel framework for an AI-enabled legal case analysis, data wrangling, and decision-making tool, called JuriumLegal++. Our contributions are as follows:

Agentic AI Framework. An orchestrator agent manages multiple agents performing specific data wrangling tasks, including document profiling, text cleaning, legislation and reference standardisation, feature extraction, validation, model recommendation, and explainable briefing.

Separation between planning and execution. While the LLM is responsible for planning data wrangling steps, reasoning, and providing justification in natural language, all actual transformations to text, chunking and similarity calculations are performed by a deterministic database and vector library. As a result, the impact of LLM hallucination is significantly reduced.

Legal Terminology and precedent standardisation using RAG. A RAG pipeline standardises messy, informal, and ambiguous Legal Terminology and regional references to legal cases by mapping them to standardised legal concepts using a verified vector database.

Iterative quality check with 5 data quality criteria. A quality check has been implemented to ensure high quality data after wrangling. Data is checked for Completeness, Uniqueness, Validity, Consistency and Accurate References. If the total quality score falls below a verified threshold, the process will automatically re-plan data wrangling.

Explainable and traceable decision making. Full lineage is recorded throughout data wrangling, and justification for all changes is provided. The methodology includes empirical evaluation on the Indian Legal Documents Corpus (ILDC) [8].

The rest of this paper is structured as follows. Related work is presented in section II highlighting research gaps. Problem statement and objective are outlined in section III. The proposed JuriumLegal++ framework and quality loop are presented in section IV. Details of the system design, architecture and implementation plan are given in section V. Evaluation method and expected results are given in section VI. Finally, conclusions and future research directions are described in section VII.

II. RELATED WORK

A. LLMs for General Data Wrangling

[1] Demonstrates how prompt-tuned LLMs can be used to perform various data wrangling operations, albeit the randomness in the LLMs generation process poses risks to reliability. [2] proposes AutoDW, which automates the generation of data transformation scripts using code generating LLMs. [3] investigates PEFT in order to reduce the computational cost of fine-tuning large models for data wrangling in enterprise settings. [5] uses code generating LLMs to automatically detect errors, infer types and impute values.

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 4, November 2026
Impact Factor: 8.2

B. LLMs for Data Preprocessing in Specialised Domains and Legal

[4] investigate fine-tuning of generative LLMs to extract clinical entities from text. [6] explore using LLMs to automate data cleaning and feature engineering within clinical data sets. While this shows the effectiveness of LLMs in this domain, it is not a complete system. It is therefore exploratory in nature. For structured data, [7] propose an open source rule based data wrangling pipeline tool, called FIDDLE, which transforms complex data into feature matrices suitable for ML. However, this does not make use of any LLMs in its processing. It cannot handle unstructured notes, and does not provide any explanations in natural language about why certain features were selected. Within the legal domain, several models have been developed, such as LEGAL-BERT [2] and Lawformer [9]. The models perform well in discriminative tasks, including NER and predicting verdicts. However, these models do not form part of a complete end-to-end agent workflow. They also have no built-in quality improvement loops and cannot dynamically integrate into legal practices.

C. Research Gap

A summary of related work in automated data wrangling, specialisation domain text preprocessing and AI aided analytics is provided in table I below. Most existing research performs one specific data wrangling task on general data, or performs very specific extraction tasks, or relies completely on rule-based heuristics. To the best of our knowledge, there has not been research which combines automated profiling, cleaning and standardising citations via RAG, argument engineering, verification of evidence and explainable decision making in legal documents, to create a complete self-improving agent system with quality loops.

TABLE I. COMPARISON OF RELATED WORK

Ref.	Focus	Main limitation
[1]	General wrangling via GPT prompting	Stochastic reliability; lacks legal domain knowledge
[2]	End-to-end code generation for wrangling	General-purpose; devoid of judicial context
[3]	Parameter-efficient fine-tuning for wrangling	Benchmark only; no end-to-end domain pipeline
[4]	Information extraction from unstructured text	Single task; no multi-agent orchestration
[5]	Code-generating imputation and error correction	Selected tasks; lacks feedback-driven quality gate
[6]	LLM-based cleaning and feature engineering	Evaluative study; not an integrated operational system
[7]	Rule-based feature transformation pipeline	Rigid rules; lacks LLM reasoning and explainability
This work	Agentic, legal-specific, explainable pipeline with quality gate loop	Concept and architecture stage; empirical evaluation planned

III. PROBLEM STATEMENT AND OBJECTIVES

Court case data in court dockets, portals, electronic filing systems, witness statements, and scanned affidavits can suffer from incomplete information, duplicated data, inconsistent legal references, wrong data types, anomalous values in settlements, inconsistent time formatting, and large amounts of unstructured text. The manual processing of legal datasets takes an inordinate amount of resources, increases cost in legal processes, and is difficult to scale at large volumes. Existing AI applications lack holistic design and cannot be guaranteed to cite accurate information.

Problem Description:

Given a raw legal case dataset/docket D containing various errors and unstructured data, output a processed legal dataset D' that can be applied to machine learning models to improve decision making in legal cases. Multi-dimensional quality scores must be calculated for both D and D'. An immutable record of transformation must be maintained. Finally, explainable justifications for all legal recommendations must be provided.

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 4, November 2026
Impact Factor: 8.2

Aim:

To create an agent driven workflow to perform legal case analysis using Large Language Models.Identify and correct missing information within legal cases, duplication of docket data, and chronological mismatches. Standardise legal terminology, court names and legal section numbers across both historic and modern legal codes.

Design an agent driven system with sub agents.

Automatically create features from legal data and confirm their relevance to the case. Create a quantified quality measure for legal data, including completeness, uniqueness, validity, consistency and accurate referencing. Provide explainable recommendations and automatically generate summarised case briefs using a reporting interface.

IV. METHODOLOGY

Overview:

As shown in Fig. 1, data from case dockets and legal databases via the web interface is interpreted by the AI Orchestrator (represented as a state machine in LangGraph). The entire workflow and sub agent execution is planned and interpreted by the AI Orchestrator. The planning, interpretation, and justification is performed by the LLM. Data manipulation is performed using deterministic modules (Pandas, NumPy, Scikit-learn, PostgreSQL pgvector, and Great Expectations). Feature validation of evidence and an iterative quality check ensures data is accepted or rejected.

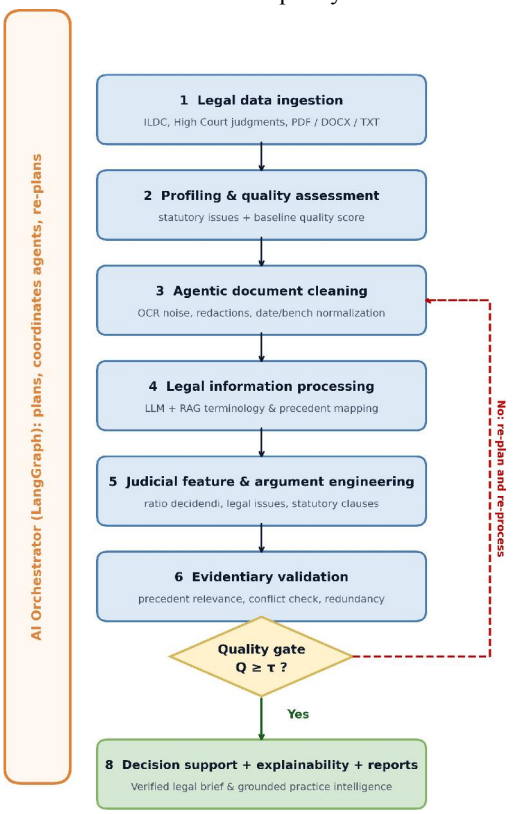


Fig. 1. Workflow of the proposed JuriumLegal++ framework.

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 4, November 2026
Impact Factor: 8.2

B. Agentic Architecture

Each agent is aware of the shared state: current schema, quality metrics, performed actions and iterations. Based on the identified problems, appropriate agents were selected from the following list: "Legal Ingestion Agent", "Legal Quality Agent", "Docket Mapping Agent", "Judicial Cleaner Agent", "Legal Knowledge & RAG Agent", "Argument Feature Engineering Agent", "Evidence Validation Agent", "Decision Analysis Agent", "Explainable AI Agent" and "Report Generation Agent". Each agent performs one specific task, can be tested independently and maintains a trace of the actions taken.

C. Profiling and Quality Assessment

For each column and text field, the "Legal Quality Agent" detects missing metadata, duplicates, invalid values, wrong data types, outliers, and inconsistencies in the legal schema. The five quality dimensions in Table II [10] [11] are computed. All dimensions have values within the range [0, 1] and the global quality metric Q is defined by computing the weighted average of all quality dimensions. By default, all weights are equal to 1.

TABLE II. DATA-QUALITY DIMENSIONS AND FORMULAS

Dimension	Definition used in this work
Completeness	1 − (missing cells / required metadata) / total required fields
Uniqueness	1 − (duplicate docket filings) / total filings
Validity	valid statutory references and date formats / non-missing values (type, format and range rules)
Consistency	values in standard legal nomenclature and unit / non-missing values
Accuracy	citations verified against canonical court repository / total extracted citations (ground truth verification)

$$Q = \Sigma\, w_i \cdot D_i \,,\ \Sigma\, w_i = 1 \quad (1)$$

where D_i are the five dimension scores. The data is accepted when $Q \geqslant \tau$, where τ is a configurable threshold (default 0.90).

D. Cleaning and Standardisation

Cleaning Tools. For each legal case, specific tools are used. Gaps in procedural information and court numbers are filled using sequential case information, or filled forward using related bench procedures. Duplicate procedural information is removed using a compound key based on court ID, Case Type, case number, and year. Outliers in claim, damage and penalty variables are identified in comparison to statutory maximums and using the IQR rule, with values outside the range (Q_1-1.5IQR, Q_3+1.5IQR) flagged as missing or, if extreme litigations were confirmed, flagged. References to sections in different versions of legal codes (IPC 302 to BNS 103) are deterministically mapped.

E. Legal RAG

Expressions from medical and legal domains, such as anticipatory bail (438), pre-arrest bail (CrPC) and plea under section 438 should be mapped to a single canonical term. Canonical statutes referenced in the cases (IPC, CrPC, CPC, BNS, BSA) and verified precedent cases, are stored within a vector DB (PostgreSQL with pg-vector plug-in, ChromaDB). For each term/proposition, the most similar terms are retrieved. If the best matching term does not appear semantically similar, the LLM returns only these terms. If some citations cannot be mapped to canonical sources within court repositories, they are marked as unmapped. This prevents hallucination of non-existent case citations and legal statutes.

F. Legal Features & Validation

Legal features, including exposure to legal statutes, litigation duration (time to disposal in months), density of precedent, adherence to the limitation period and category of cause of action, are engineered by the Argument Feature

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 4, November 2026 Impact Factor: 8.2

Engineering Agent. From this, the Evidentiary Validation Agent removes features of minimal variance. If two features have a correlation
> 0.9, one is removed. Mutual information between remaining features and legal outcome is calculated and features are prioritised accordingly. Great Expectations testing is enforced on all data, data ranges and null constraints.

G. Quality Gate

Algorithm 1 describes the iterative process of the framework allowing self-correction, with N being the maximum iterations.

Algorithm 1: Agentic legal data processing with quality gate Inputs: Data Set D, Threshold τ, Max Iterations N.
Q = score(D) i=0
while Q < τ and i < N:
plan = orchestrator.plan(issues(D)) #LLM for op in plan:
D = tool(op)(D) #Pandas/SQL/Regex D = validate(features(D))
Q = score(D) i += 1
if Q >= τ:
return D for machine learning else:
Flag data for legal review by Advocate.

H. Decision Analysis & Interpretability

Once data passes the quality gate, baseline prediction models (Logistic Regression, Random Forest, XGBoost) are trained using stratified k-fold cross validation. The Accuracy, Precision, Recall, F1-Score and AUROC is recorded. The interpretable AI agent gives an explanation of data cleaning, statute mapping, feature engineering and quality improvements in layman's terms. Additionally, SHAP is used to interpret the exact contribution of evidence features to legal outcome predictions.

V. SYSTEM DESIGN

There are four main components to this system. A NextJS 14 application with React and Typescript is used to create mobile and web applications. Using TailwindCSS and ReCharts, users can upload case dockets, view data quality reports, data wrangling processes, inspect features in data, and automatically export case summaries. The NodeJS backend written in Typescript, with Supabase JWT and Multi-Tenancy Row Level Security, manages user authentication, separation of dockets, user permissions and job management. Python FastAPI microservices contain the LLM orchestrator, dense embedding model, LangGraph agents, and RAG system. Data is stored in PostgreSQL (User Data, Metadata, Quality Reports, Lineage), pgvector (embedding of legal knowledge and precedent cases), Redis (fast query of sessions) and secured Cloud Storage (raw and processed legal documents).

This system design has been proposed to be tested on the Indian Legal Documents Corpus (ILDC) [8]. It contains thousands of judgements from the Supreme Court of India, precedent cases from High Courts, and user upload case dockets (.PDF, .DOCX, .TXT). Only public legal data and secure case files (isolated per tenant) are used. The system currently follows a human in the loop approach, and does not interface with active legal registries.

VI. EVALUATION PLAN

The framework will be evaluated through multiple experiments, using subsets of the ILDC data and introducing artificial errors into original case dockets in order to accurately evaluate the ability of the system to detect and correct issues in comparison with ground truth.

Accuracy of Detection and Correction: Errors in dockets (missing date of filing, duplication of motions, inconsistent citations, OCR errors) are introduced at varying percentages (5%-30%). Precision, recall and accuracy of correction is measured.

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 4, November 2026 Impact Factor: 8.2

Improvement in Data Quality: Five dimensional quality scores, and overall quality score (Q), are measured before and after agentic data wrangling for 1000 diverse case dockets.

ML Model Performance: The performance of models trained using minimally pre-processed raw case data is compared to models trained using data processed by the framework. Accuracy, Precision, Recall, F1-Score and AUROC is measured for predicting different legal outcomes.

Correctness of Terminology Mapping: Accuracy of mapping legal terms and citations is compared to a manually validated gold standard of 500 legal terms and canonical legal statutes.

Efficiency & Reliability: Time taken to prepare case dockets is compared to manual preparation by legal assistants. Outputs from LLMs which are invalid or not grounded are systematically filtered out.

Ablation Study: Two variations of the framework are tested. One without the quality-gate re-planning step, and one without RAG precedent mapping, to evaluate the impact of these components.

Significant time will be saved manually preparing case data, data quality in all 5 dimensions will be significantly improved, and the analysis process will be significantly sped up. As a result, machine learning models using this data should be more accurate. Additionally, legal professionals will be able to see detailed explanations regarding all decisions made by automated processes in an auditable manner. These are firm aims of this research, and will be validated by the experimental evaluations outlined.

Limitations

Only open source and simulated legal data was used. Complete digitisation of legal repositories and correct inference from LLMs is required for this research to function correctly. As ground truth for full trials may not be available, results are evaluated using legal plausibility. Finally, strategic decision making and legal advocacy in courts must be completed by qualified legal practitioners.

VII. CONCLUSION & FUTURE WORK

Within this research paper, the idea of creating an automated legal case analysis and data wrangling framework (JuriumLegal++) using AI is presented. By combining planning with an LLM orchestrator, standardising legal terminologies using RAG, deterministic execution, and a 5 dimensional quality gate with re-planning, unstructured legal case dockets may be converted into clean, standardised, validated and ready-to-train-for-machine learning datasets with very little manual effort. In future work, this framework will be fully implemented and the planned experiments will be performed on the ILDC and High Court data sets. Comprehensive results will be produced. Potential future work includes wrangling live courtroom transcript data, and self learning policies within agents to adapt to preferences of individual benches, cross-jurisdictional legal harmonization, and multimodal analysis of courtroom video and evidentiary exhibits.

REFERENCES

[1] G. Jaimovitch-López, C. Ferri, J. Hernández-Orallo, F. Martínez-Plumed, and M. J. Ramírez-Quintana, “Can language models automate data wrangling?,” Machine Learning, vol. 112, pp. 2053–2082, 2023.

[2] L. Liu, S. Hasegawa, S. K. Sampat, M. Xenochristou, W.-P. Chen, T. Kato, T. Kakibuchi, and T. Asai, “AutoDW: Automatic data wrangling leveraging large language models,” in Proc. 39th IEEE/ACM Int. Conf. Automated Software Engineering (ASE), 2024, pp. 2041–2052.

[3] Z. Zhang, P. Groth, I. Calixto, and S. Schelter, “Directions towards efficient and automated data wrangling with large language models,” in Proc. IEEE 40th Int. Conf. Data Engineering Workshops (ICDEW), 2024, pp. 301–304.

[4] D. Vithanage, C. Deng, L. Wang, M. Yin, M. Alkhalaf, Z. Zhang, Y. Zhu, and P. Yu, “Adapting generative large language models for information extraction from unstructured professional records,” J. Healthcare Informatics Research, vol. 9, pp. 191–219, 2025.

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 4, November 2026
Impact Factor: 8.2

[5] A. Akella and K. Narayanam, "Data wrangling task automation using code-generating language models," in Proc. AAAI Conf. Artificial Intelligence, vol. 39, no. 28, 2025, pp. 29616–29618.
[6] S. A. Babli, P. Gairola, C. E. Ogbu, F. A. Longa, A. Rautaray, O. S. Alamu, and S. F. H. Rizvi, "Evaluating large language models for automated data cleaning and feature engineering in specialized datasets," Cureus, vol. 18, no. 5, e108550, 2026.
[7] S. Tang, P. Davarmanesh, Y. Song, D. Koutra, M. W. Sjoding, and J. Wiens, "Democratizing domain analyses with FIDDLE: A flexible data-driven preprocessing pipeline for structured professional data," J. Amer. Med. Informatics Assoc., vol. 27, no. 12, pp. 1921–1934, 2020.
[8] V. Malik, R. Sanjay, S. K. Nigam, K. Ghosh, S. Guha, A. Bhattacharya, and P. Majumder, "ILDC for CJPE: Indian Legal Documents Corpus for court judgment prediction and explanation," in Proc. 59th Annu. Meeting Assoc. Computational Linguistics (ACL), 2021, pp. 4046–4062.
[9] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, and H. Wang, "Lawformer: A pre-trained language model for Chinese legal long documents," AI Open, vol. 2, pp. 79–84, 2021.
[10] R. Y. Wang and D. M. Strong, "Beyond accuracy: What data quality means to data consumers," J. Management Information Systems, vol. 12, no. 4, pp. 5–33, 1996.
[11] M. G. Kahn et al., "A harmonized data quality assessment terminology and framework for the secondary use of electronic record data," eGEMs, vol. 4, no. 1, art. 1244, 2016.
[12] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 9459–9474.
[13] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[14] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[15] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 4765–4774.