Experimental Evaluation of an Explainable Multi-Condition Rule-Based Framework for Automated Email Attachment Extraction and Signature Validation

Ioan Florin Anghiuș
Automation Department
Faculty of Automation and Computer Science,
Technical University of Cluj-Napoca
Cluj Napoca, Romania
Ioan.Anghius@campus.utcluj.ro

Honoriu Vălean
Automation Department
Faculty of Automation and Computer Science,
Technical University of Cluj-Napoca
Cluj Napoca, Romania
honoriu.valean@aut.utcluj.ro

***Abstract* - Enterprise email processing often involves large volumes of heterogeneous messages and attachments that must be filtered, validated, and integrated into downstream business systems. A major challenge consists in accurately identifying only those documents that satisfy complex business conditions while maintaining full explainability, auditability, and operational reliability. This paper presents the experimental evaluation of an explainable multi-condition rule-based framework for automated email attachment extraction and signature validation developed using UiPath. The core contribution is a deterministic extraction engine that formalizes domain knowledge as regular expressions and conditional business rules to analyze email content, verify eligibility criteria, extract the correct attachments, and generate detailed diagnostic information for each processed transaction. Extracted documents are standardized and submitted to UiPath Action Center for human validation of signatures and other visual document characteristics. Approved documents are subsequently processed through a queue-based Dispatcher–Performer architecture and uploaded automatically to the target enterprise platform. The framework also incorporates a state-based folder organization and a multi-sheet audit report that provide complete end-to-end traceability. The experimental study is based on real production emails and document attachments covering multiple document categories and processing scenarios. Results demonstrate that the proposed approach significantly reduces manual effort while ensuring high extraction accuracy, robust exception handling, and full process transparency. In addition, the collected validation outcomes constitute a labeled dataset that can support future integration of machine learning and document understanding models aimed at progressively reducing human intervention.**

Keywords - UiPath, UiPath Action Center, UiPath Orchestrator, Robotic Process Automation, Document Understanding, Regular Expressions (Regex), Match Activity, Rule-Based Extraction, Explainable Automation, Multi-Condition Decision Logic, Human-in-the-Loop Validation, Email Attachment Extraction, Signature Validation, Queue-Based Processing, Dispatcher–Performer Architecture, Audit and Traceability, Machine Learning.

I. Introduction

Robotic Process Automation (RPA) has become a widely adopted technology for automating repetitive and rule-based activities in enterprise environments [1], [2], [3]. Platforms such as UiPath enable organizations to automate email processing, document handling, and integration with business applications while improving efficiency, accuracy, and auditability.

A significant challenge in enterprise document workflows is the accurate identification of relevant attachments from large volumes of heterogeneous email messages. In many practical scenarios, each email may contain multiple attachments, while only a subset satisfies the business conditions required for further processing. The automation system must therefore analyze the email content, verify multiple eligibility criteria, extract mandatory variables, and select only the documents that meet all predefined rules. At the same time, each decision must remain fully explainable and traceable [4].

This paper presents the experimental evaluation of an explainable multi-condition rule-based framework for automated email attachment extraction and signature validation developed using UiPath. The core contribution is an explainable extraction framework based on deterministic Regular Expressions (Regex) and conditional business rules. The engine analyzes email content, extracts critical variables such as transaction identifiers and candidate identifiers, and selects only the attachments that satisfy all eligibility conditions. The extracted documents are then submitted to UiPath Action Center for human-assisted validation and subsequently processed through a queue-based Dispatcher–Performer architecture[5]. Recent advances in intelligent document processing and UiPath Document Understanding provide opportunities for extending such frameworks with machine learning capabilities [6]. The experimental study is based on real production emails and document attachments covering multiple document categories and processing scenarios. Results show that the proposed approach significantly reduces manual effort while maintaining high extraction accuracy, robust exception handling, and full process transparency.

The main contributions of this paper are: (1) an explainable multi-condition rule-based extraction engine; (2) a human-assisted validation layer; (3) a state-based folder organization for modular orchestration; (4) a queue-based processing architecture; and (5) an experimental evaluation conducted on real-world industrial data. The remainder of this paper is organized as follows. Section II reviews related work. Section III presents the proposed framework. Section IV describes the experimental setup. Section V discusses the experimental results. Section VI outlines future research directions. Section VII concludes the paper.

II. Related Work

Robotic Process Automation (RPA) has become a widely adopted technology for automating repetitive and rule-based activities across enterprise environments [7]. Existing studies demonstrate that RPA can significantly improve execution speed, accuracy, compliance, and operational efficiency, particularly in processes involving structured digital interactions with enterprise applications.

Rule-based information extraction remains a highly effective approach when the target patterns are known and decision transparency is required. Regex enable domain knowledge to be formalized as deterministic pattern-matching rules, providing explicit decision transparency, structured diagnostic reporting, and complete processing traceability[8]. Such approaches are particularly suitable for document-centric workflows in which multiple business conditions must be verified before relevant information and attachments can be extracted.

In many practical scenarios, full automation is insufficient because certain decisions require visual inspection and contextual interpretation. Human-in-the-loop approaches address this limitation by combining automated processing with selective human validation. Within the UiPath ecosystem, Action Center enables unattended workflows to suspend execution, request human decisions, and resume processing after validation is completed. This mechanism is especially useful for document verification tasks involving signatures and other visual elements that cannot be reliably assessed through deterministic rules alone [9].

Recent advances in intelligent document processing and Document Understanding extend traditional RPA with machine learning methods capable of extracting and classifying information from unstructured documents [10]. Although these approaches offer significant potential, they require representative labeled datasets, model training, and continuous monitoring. In contrast, rule-based systems remain highly effective when business conditions are explicitly defined and complete explainability is mandatory.

The reviewed literature confirms the individual benefits of RPA, deterministic extraction, human-in-the-loop validation, and intelligent document processing. However, existing studies typically focus on isolated automation components, generic RPA workflows, or machine learning-based document analysis systems. Fewer works report experimental evaluations of integrated enterprise frameworks combining multi-condition Regex-based extraction, human-assisted signature validation, queue-based transaction processing, state-based orchestration, and comprehensive audit reporting using real industrial data.

This paper addresses this gap by presenting and experimentally evaluating an explainable multi-condition rule-based framework developed using UiPath. Unlike conventional RPA workflows that rely primarily on sequential automation logic, the proposed framework integrates deterministic extraction, selective human validation, scalable Dispatcher–Performer orchestration, and structured auditability into a unified and fully traceable enterprise document-processing architecture.

III. Proposed framework

A. Multi-Condition Rule-Based Extraction Engine

The multi-condition rule-based extraction engine constitutes the core contribution of the proposed framework and represents the principal source of the experimental results presented in this paper. Its objective is to automatically identify, classify, and extract only those document attachments that satisfy a predefined set of business conditions from large volumes of heterogeneous enterprise email messages.

The extraction process begins by reading incoming Office 365 emails and collecting the message subject, body, received date, and attachment metadata. The email body is subsequently normalized to eliminate unnecessary spaces and line breaks, thereby ensuring consistent and reliable pattern matching. A deterministic sequence of Regular Expressions (Regex) and conditional business rules is then applied to the normalized text in order to verify eligibility conditions and extract critical identifiers.

The eligibility logic evaluates both the presence and the expected values of key variables, including **Existing = True**, **TrainingRequired = NO**, **SuppDocsRequired = NO**, and **EvaluationResult = Successful**. Additional Regex rules are used to extract transaction-specific identifiers such as **ROEIndexID**, **CandidateID**, and **TaskIdentity**. The presence of at least one valid attachment is also mandatory. If any condition is not satisfied, the email is rejected and the specific reason is recorded in the audit report, ensuring full explainability and traceability of the decision. Only emails satisfying all eligibility conditions are forwarded to the attachment extraction module. Relevant documents are saved in the processing folder, converted to TIFF format, and prepared for human-assisted validation. Invalid or missing attachments are classified as business exceptions and recorded together with diagnostic information.

Figure 1 illustrates the simplified workflow of the proposed extraction engine, from incoming email analysis and rule-based validation to attachment extraction and document standardization.

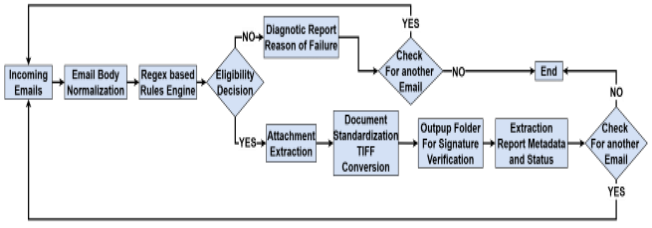

Fig. 1. Simplified Multi-Condition Extraction Workflow

The overall eligibility decision is formalized by Equation (1), where E denotes the eligibility of the email for automated extraction and $R_1, R_2, \ldots, R_8$ represent the individual validation rules.

$$E = R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5 \wedge R_6 \wedge R_7 \wedge R_8$$

Table I summarizes representative validation rules, their corresponding Regular Expression patterns, expected values, and diagnostic outputs generated when a condition is not satisfied.

Table I. Representative Extraction Rules, Regex Patterns, and Eligibility Conditions

Rule ID	Validation Rule	Representative Regex Pattern	Expected Value	Failure Output
R1	Existing	`Existing\s*=\s*(True	False)`	True
R2	TrainingRequired	`TrainingRequired\s*=\s*(YES	NO)`	NO
R3	SuppDocsRequired	`SuppDocsRequired\s*=\s*(YES	NO)`	NO
R4	EvaluationResult	EvaluationResult\s*=\s*([A-Za-z]+)	Successful	Evaluation not successful
R5	ROEIndexID	ROEIndexID\s*=\s*(\d+)	Valid numeric format	Missing or invalid ROEIndexID
R6	CandidateID	CandidateID\s*=\s*([A-Za-z0-9_-]+)	Exists	CandidateID missing
R7	TaskIdentity	TaskIdentity\s*=\s*([A-Za-z0-9_-]+)	Exists	TaskIdentity missing
R8	Attachments	`.*.(pdf	tif	tiff

Algorithm 1 describes the operational logic of the extraction engine. The algorithm first normalizes the email content, evaluates all validation rules, and either extracts the eligible attachments or records the corresponding rejection reason. This procedure ensures deterministic and fully explainable classification of both email messages and their associated attachments.

Algorithm 1. Multi-Condition Extraction Logic

Input: Email message M
Output: Extracted attachment or rejection reason
Normalize email body
Check attachment presence
Apply validation rules R1 ... R8
If all rules are satisfied then
  Extract eligible attachments
  Convert documents to TIFF
  Save files in validation input folder
  Update report as Successfully Extracted
Else
  Identify failed rule
  Update report with rejection reason
End If

B. Human-Assisted Document Validation Using UiPath Action Center

Documents successfully extracted by the rule-based engine are converted to TIFF format and submitted to UiPath Action Center for human-assisted validation. This stage is used to verify signatures and other visual document characteristics that cannot be reliably assessed through deterministic rules alone.

Prior to the implementation of UiPath Action Center, the validation of extracted documents was performed using a Microsoft Excel-based solution supported by Visual Basic for Applications (VBA). Although this approach provided a functional mechanism for manual verification, it required users to inspect documents outside the automation platform and offered limited integration with the unattended processing workflow.

The UiPath Action Center validation module was designed and configured specifically for the requirements of this process. Rather than relying on a generic validation interface, the solution was modeled from scratch to support the exact document types, verification criteria, decision options, and routing rules required by the business workflow. This custom design was developed as a modern replacement for the previous Excel and VBA-based solution, providing tighter integration with the automation platform, improved usability, and more reliable traceability of human decisions.

The adoption of UiPath Action Center significantly improved the validation process by providing a centralized and fully integrated human-in-the-loop environment. Reviewers can inspect each document, confirm whether the required signatures and visual elements are valid, and submit structured PASS or FAIL decisions directly within the automation platform.

Documents marked as PASS are routed to the approved folder and become eligible for queue-based processing. Documents marked as FAIL are redirected to exception folders and recorded in the audit report together with the corresponding validation outcome.

The validation decisions collected through UiPath Action Center also constitute a valuable labeled dataset. These historical outcomes may support future research involving Machine Learning and Document Understanding models capable of automatically classifying high-confidence cases and progressively reducing the level of human intervention. Figure 2 illustrates the customized human-assisted validation workflow implemented using UiPath Action Center.

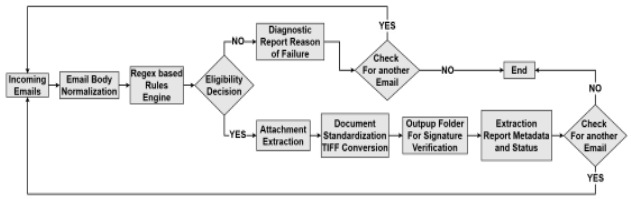

Fig. 2. Human-Assisted Document Validation Workflow Using UiPath Action Center

C. Queue Preparation and Dispatcher Process

The Dispatcher process represents the transition layer between human validation and automated business execution. It was specifically designed for the output produced by the extraction and Action Center validation modules, rather than being a generic queue-loading mechanism. Its role is to read the validated PASS/FAIL results, update the audit report, archive the processed files, isolate rejected cases, and create structured

queue items only for documents approved for further processing.

During the iterative development of the automation, several alternative designs were evaluated. In the initial versions, extraction, validation, and queue preparation were more tightly coupled, and part of the validation logic was supported by Excel and VBA. The final design separates the complete workflow into specialized modules: the rule-based extractor, the Action Center validation module, the Dispatcher, the Performer, and the reporting component. This separation improved maintainability, traceability, and process reliability.

After human validation, the Dispatcher reads the PASS and FAIL folders and associates each validated document with its corresponding metadata. For approved documents, the Dispatcher creates an Orchestrator Queue item containing the extracted identifiers, file path, validation status, email reference, and processing metadata. Failed documents are not added to the processing queue; instead, they are moved to exception paths and recorded in the audit report. This design ensures that only validated and business-eligible documents are forwarded to the Performer process.

The Dispatcher also performs housekeeping operations, such as moving TIFF files to archive folders, deleting temporary PDF files from PASS/FAIL folders, and moving emails related to failed files to dedicated exception locations. By modeling the Dispatcher as a dedicated process, the framework introduces a controlled transaction boundary between human decision-making and automated execution. Figure 3 illustrates the customized Dispatcher workflow used for queue preparation.

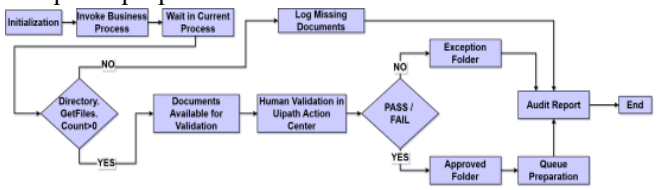

Fig. 3. Customized Dispatcher Workflow for Queue Preparation

A compact representation of the queue item generated by the Dispatcher is shown below:

QueueItem = {
Reference: ROEIndexID / Email Subject,
CandidateID: extracted candidate identifier,
TaskIdentity: extracted task identifier,
FilePath: approved document path,
ValidationStatus: PASS,
EmailMetadata: subject and received date,
ProcessingStatus: Pending
}

This Dispatcher design ensures that the Performer receives only clean, validated, and traceable transactions, reducing downstream exceptions and preserving a complete audit trail for every processed document.

D. Transaction Execution and Performer Process

The Performer process represents the final execution layer of the proposed framework. It was developed specifically to process the validated queue items generated by the Dispatcher and to upload the approved documentation into the target web-based enterprise platform. The process was modeled from scratch to ensure reliable interaction with the application interface, correct candidate identification, controlled document upload, and complete status reporting.

For each queue item, the Performer retrieves the transaction metadata, including the candidate identifier, task identity, document path, and validation status. The workflow then logs into the target web platform, searches for the corresponding candidate record, verifies candidate-related fields, and selects the record that matches the extracted metadata. This validation step is essential because an incorrect candidate selection would lead to an erroneous document upload and therefore represents a critical business risk.

To increase robustness, the Performer was designed with several defensive automation mechanisms. These include explicit field verification, candidate matching logic, page refresh handling, application closing routines, process cleanup activities, and exception management. These mechanisms reduce the risk of selector instability, page synchronization issues, application delays, and inconsistent web interface behavior during unattended execution.

After the correct candidate record is identified, the Performer uploads the approved document into the enterprise platform and updates the audit report with the final processing status. Successful transactions are marked as completed, while candidate mismatches, missing fields, upload errors, or application failures are recorded as business or system exceptions. The process can also send a final status notification and archive the corresponding report, ensuring that every transaction remains traceable from extraction to final upload.

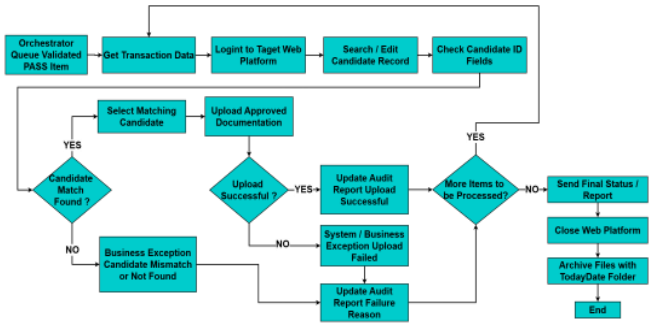

Fig. 4. Customized Performer Workflow for Web Application Document Upload

A compact representation of the Performer transaction logic is shown below:

Input: Queue item Q
Output: Uploaded document or processing exception
Read transaction metadata from Q
Login to target web platform
Search candidate record using extracted identifiers
Verify CandidateID and related fields
If matching candidate is found then
Select candidate record
Upload approved document
If upload succeeds then
Update report as Successfully Uploaded
Else
Update report with Upload Failure

End If

Else

Update report with Candidate Mismatch / Not Found

End If

Send final status notification

Close application and cleanup processes

This Performer design completes the end-to-end automation pipeline by transforming validated queue items into controlled business actions inside the target web application. Its contribution consists not only in executing the upload, but also in enforcing candidate verification, selector-resilient interaction, exception handling, and complete auditability for each processed transaction.

E. State-Based Folder Structure and Multi-Sheet Audit Reporting

The proposed framework consists of a multi-stage automation pipeline combining deterministic email extraction, human-assisted validation, queue-based transaction orchestration, and enterprise document processing. Each folder represents a distinct operational state and enables controlled communication between the extractor, Action Center validation module, Dispatcher, and Performer components.

This design prevents uncontrolled file movement between automation stages and supports exception isolation. Documents failing extraction or validation are routed to dedicated exception folders, while approved documents are transferred to the queue preparation stage. Temporary files are removed after processing, and final TIFF/PDF documents are archived according to their status, allowing the folder hierarchy to function as a lightweight state-management mechanism.

In parallel, the framework generates a multi-sheet audit report that records the evolution of each transaction from extraction to final upload. The report stores extraction results, validation outcomes, transaction identifiers, processing statuses, and upload results, thereby ensuring complete traceability across the automation pipeline.

The reporting mechanism provides an auditable data trail for each processed transaction and supports later analysis of extraction accuracy, validation outcomes, exception frequency, and upload success rate. Figure 5 illustrates the state-based folder and reporting structure used by the framework.

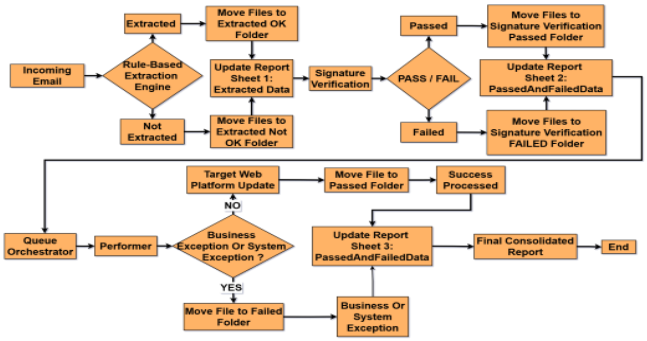

Fig. 5. State-Based Folder Structure and Multi-Sheet Audit Reporting.

Table II summarizes the structure of the three audit worksheets and the main categories of information recorded at each processing stage.

Table II. Multi-Sheet Audit Report Structure

Worksheet	Processing Stage	Main Information Recorded
Extracted Data	Extraction Engine	Email subject, received date, processed time, extraction status, diagnostic description, extracted file name
PassedAnd FailedData	Human Validation	File name, ROE index, Candidate ID, Task Identity, PASS/FAIL validation status
Performed Data	Performer Process	Processed time, file name, ROE index, Candidate ID, upload status, final status description

The combination of deterministic extraction, human-assisted validation, Dispatcher–Performer orchestration, state-based folder organization, and multi-sheet audit reporting provides complete end-to-end traceability of every processed document. Each transaction can be followed from the original email through extraction, validation, queue preparation, and final upload into the target enterprise platform. This integrated architecture supports operational monitoring, exception recovery, experimental evaluation, and future data-driven enhancements.

IV. Experimental Setup

The experimental evaluation was conducted using 25 production reports collected throughout January 2024, comprising 7,469 enterprise emails and their associated document attachments. The dataset included heterogeneous emails containing multiple attachments, of which only a subset satisfied the business conditions required for automated processing. The documents covered a wide range of processing scenarios, including successful extractions, missing mandatory variables, invalid attachment formats, failed signature validation, business exceptions, and successful uploads to the target web platform.

The proposed framework was implemented using UiPath and executed under the control of UiPath Orchestrator. Human-assisted validation was performed through UiPath Action Center, while audit data were recorded in a multi-sheet Microsoft Excel workbook. The automation was deployed in an unattended environment and integrated with a web-based enterprise application used for final document upload.

The experimental methodology followed the complete end-to-end workflow described in Section III. Incoming emails were first processed by the multi-condition rule-based extraction engine. Successfully extracted documents were submitted to Action Center for signature validation. Documents approved during human validation were then added to an Orchestrator Queue by the Dispatcher and processed transaction by transaction by the Performer. At each stage, detailed operational data were recorded in the audit report. The evaluation focused on both functional correctness and operational performance.

The principal metrics considered in this study were:

- extraction success rate;
- number of business exceptions detected during extraction;

- number of documents classified as PASS and FAIL during human validation;
- queue creation success rate;
- upload success rate;
- end-to-end processing traceability.

In addition to quantitative metrics, the experimental analysis examined the explainability of the framework by assessing the ability of the system to provide explicit diagnostic messages and structured status information for each processed transaction. The resulting dataset, consisting of extraction outcomes, validation decisions, and final processing statuses, also provides a structured basis for future Machine Learning and Document Understanding research aimed at reducing the level of human intervention.

V. Experimental Results And Discussion

The experimental evaluation confirmed the effectiveness of the proposed framework under real industrial operating conditions. Table III summarizes the principal performance indicators obtained from 25 production reports collected throughout January 2024.

Table III. Experimental Performance Summary

Metric	Value
Production Reports Analyzed	25
Total Emails Processed	7,469
Successful Extractions	7,193
Extraction Success Rate	96.30%
Extraction Exceptions	276
Documents Validated	7,190
PASSED Documents	6,532
FAILED Documents	658
Signature Validation Pass Rate	90.85%
Queue Items Executed	6,083
Successful Uploads	5,550
Upload Success Rate	91.24%
Performer Exceptions	533

The framework achieved an extraction success rate of 96.30%, a signature validation PASS rate of 90.85%, and an upload success rate of 91.24%. These results demonstrate that the proposed architecture provides reliable and scalable automation while maintaining explainability, auditability, and selective human-assisted verification.

VI. Future Research Directions

The experimental results demonstrate that the proposed rule-based framework provides an efficient and highly reliable solution for automated email attachment extraction and signature validation. The combination of deterministic extraction, customized human-assisted validation, queue-based processing, and comprehensive audit reporting produced results that exceeded initial expectations and confirmed the practical effectiveness of the approach in real industrial conditions.

A promising direction for future research is the integration of Machine Learning and Document Understanding techniques, such as a custom ML Extractor, to automatically classify signatures and other visually ambiguous document characteristics. The PASS and FAIL decisions collected through UiPath Action Center, together with the associated documents and audit information, constitute a structured labeled dataset suitable for supervised model training.

In a hybrid architecture, high-confidence cases could be processed automatically, while only uncertain documents would be routed to human validation. New validation outcomes could be incorporated into the training dataset and used to periodically retrain the model, enabling continuous improvement and progressively reducing the level of human intervention while preserving the explainability, auditability, and operational robustness demonstrated in this study.

VII. Conclusion

This paper presented an explainable multi-condition rule-based framework for automated email attachment extraction and signature validation developed using UiPath. The solution integrates a deterministic extraction engine, a customized UiPath Action Center validation module, dedicated Dispatcher and Performer processes, and a state-based folder and reporting system providing complete end-to-end traceability.

Experimental results obtained from real production emails demonstrate that the proposed framework significantly reduces manual effort while maintaining high reliability, explainability, and auditability. The resulting extraction outcomes, PASS/FAIL decisions, and final processing statuses also provide a structured labeled dataset for future Machine Learning and Document Understanding research aimed at progressively reducing human intervention.

The proposed architecture additionally demonstrates that deterministic rule-based automation can remain highly effective in enterprise environments where business rules are clearly defined and full decision traceability is required.

References

[1] M. Lacity and L. Willcocks, "Robotic Process Automation and Cognitive Automation: The Next Phase," Journal of Information Technology Teaching Cases, vol. 8, no. 2, pp. 86-95, 2018.

[2] A. Hofmann, M. Samp, and N. Urbach, "Robotic Process Automation," Electronic Markets, vol. 30, no. 1, pp. 99-106, 2020.

[3] S. Madakam, R. M. Holmukhe, and D. K. Jaiswal, "The Future Digital Workforce: Robotic Process Automation (RPA)," Journal of Information Systems and Technology Management, vol. 16, 2019.

[4] D. Gunning, "Explainable Artificial Intelligence (XAI)," Defense Advanced Research Projects Agency (DARPA), 2017.

[5] UiPath, "Action Center Documentation," [Online]. Available: https://docs.uipath.com/action-center/.

[6] UiPath, "Document Understanding Guide," [Online]. Available: https://docs.uipath.com/document-understanding/.

[7] W. M. P. van der Aalst, "Robotic Process Automation," in Intelligent Systems in Accounting, Finance and Management, 2018.

[8] R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. J. Leemans, C. Ouyang, A. H. M. ter Hofstede, I. van de Weerd, M. T. Wynn, and H. A. Reijers, "Robotic Process Automation: Contemporary Themes and Challenges," Computers in Industry, vol. 115, 2020.

[9] H. K. Dam, T. Tran, A. Ghose, and A. K. Kalia, "Human-in-the-Loop Artificial Intelligence", Annals of Operations Research, vol. 308, 2022.

[10] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2021.