IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155280

Data-Driven Billing Reconciliation and Workforce Analytics via UiPath and Power BI

Ashutosh Mankar[1], Dr G.R Bamnote[2], Prof. S.P Akarte[3]

Student, Department of Computer Science and Engineering, PRMIT&R, Amravati, India[1]

Professor, Department of Computer Science and Engineering, PRMIT&R, Amravati, India[2]

Assistant Professor, Department of Computer Science and Engineering, PRMIT&R, Amravati, India[3]

Abstract: This research paper presented an Data driven powered robot for OCR driven billing reconciliation using UiPath. The study used an open source Accounts Receivable dataset exported from an Baan Application system, where customer balances were recalculated from invoices, payments, credits, and adjustments to identify integration errors. The project used three scripts for synthetic data generation, rule based reconciliation, and anomaly detection through Linear Regression. The UiPath based automation was developed with Outlook, Excel, System, and DataTable activities, along with reusable workflows for email reading, OCR extraction, file validation, data cleaning, matching, exception handling, and output generation. Extracted information from invoices and receipts was converted to structured Excel output to be reviewed and reported on. The results showed significant changes in operations including 87.57% less time spent processing invoices, 65% less time spent preparing orders, 66.67% less time spent reconciling inventory payments, 90% less time spent making mistakes when matching invoices and 20 to 25% less money spent on operations.

Keywords: AI ML, OCR, UiPath, Billing Reconciliation, Robotic Process Automation (RPA), Invoice Processing, Employee Analysis.

I. INTRODUCTION

Billing reconciliation was a critical finance operation because customer balances had to be checked correctly against invoices, payments, credits, and adjustments exported from an ERP system. Manual reconciliation often became slow and error prone when large records, invoice details, receipt information, exchange rates, rounding differences, and missing values had to be verified together. This research paper focused on an AI ML powered robot for OCR driven payment reconciliation using UiPath. The study used an open source Accounts Receivable dataset and created synthetic ERP records for 1,000 customers. Some rows were intentionally broken through missing data, wrong exchange rates, or rounding errors so that the system could detect reconciliation issues. The task involved Python based balance recalculation, Machine learning based anomaly detection, OCR extraction and UiPath automation. The robot collected details from invoices and receipts, cleaned and normalized the data, converted unstructured invoice information into Excel output, applied matching and validation logic, and provided reconciliation reports that were suitable for auditing. All of this was done to increase speed, accuracy, and exception handling.

The proposed framework not only reduces manual effort but also establishes a replicable model for organizations seeking to modernize financial operations through intelligent automation.

II. METHODOLOGY

The dataset used is an open source Accounts Receivable (AR) data exported from an Baan Application system. It checks whether customer balances were integrated correctly by recalculating them from invoices, payments, credits, and adjustments — just like the Baan does. The project has three Python scripts that work together:

1. generate_ar.py
Creates a synthetic Baan export with 1,0000 customers.
Each customer has invoice, payment, credit, and adjustment records.
About 10–12% of the rows are intentionally "broken" — missing data, wrong exchange rate, or rounding errors — so the system has something to detect.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155280

2. reconcile.py

Recalculates the balance for every customer using the same formula the ERP would use:
Balance = ((Invoice.total - Invoice.applied) * exchange_rate)- ((Payment.total - Payment.applied) * exchange_rate)
- ((Credit.total - Credit.applied) * exchange_rate)
+ ((Adjustment.total - Adjustment.applied) * exchange_rate)
Then it compares the recomputed balance with the reported ERP balance.
If there's a difference, it flags the row and gives a likely cause — for example:
"Invoice applied exceeds total"
"Check Payment / exchange rate / rounding"

3. anomalies.py

This step uses a simple machine-learning model (Linear Regression) to predict balances based on all numeric fields.
If a balance doesn't fit the overall data pattern, it's marked as an anomaly — even if the rule-based check didn't catch it.
This helps find subtle integration errors that break the expected relationships between invoices, payments, and credits.

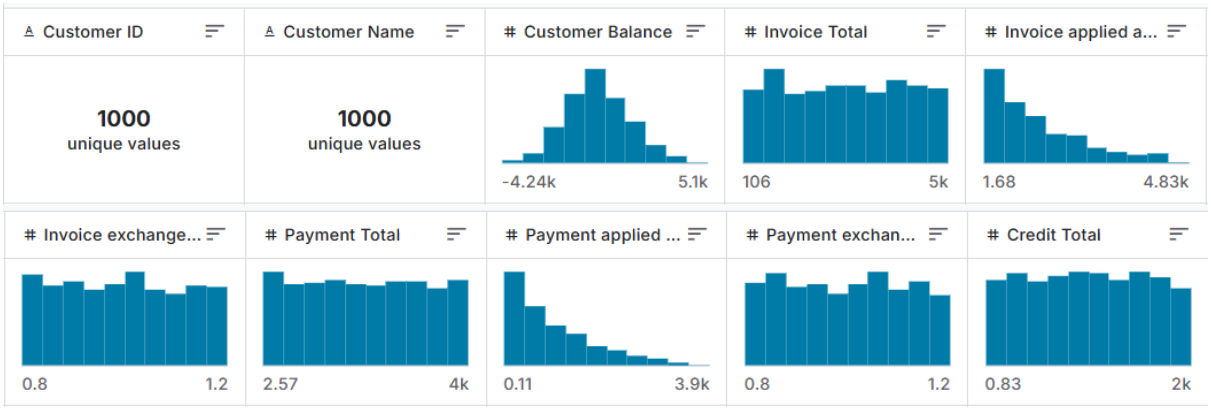


Figure 1: Payment reconciliation ERP dataset features

The system was developed using **UiPath Studio and Power Bi** with standard Outlook, Excel, System, and DataTable activities. The automation logic was structured through reusable workflows for email reading, attachment downloading, file validation, OCR extraction, Excel reading, data cleaning, Vlookups, matching, exception handling, and output generation. UiPath Orchestrator was used for unattended scheduling, queue management, asset storage, role based access, centralized logging, and execution monitoring. The synopsis also specified the use of REFramework, Orchestrator queues, assets, RBAC, centralized logs, and environment specific configurations for development, testing, and production deployment.

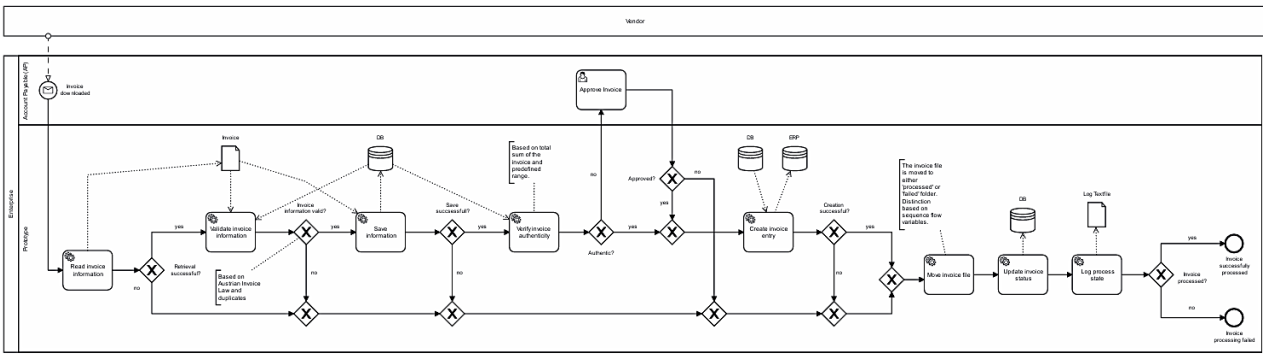


Figure 2: Automated Invoice processing

OCR and document extraction formed the intelligent processing layer of the study. Structured email bodies were parsed through rules and regular expressions, while scanned receipts, PDF receipts, and image based documents were processed through OCR or Intelligent Document Processing. The areas which were taken out were the name of the buyer, the order number, date of receipt, amount paid, currency, bank reference number and description of the payment.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155280

This step was helped by UiPath Document Understanding, as you were able to load documents, do some initial work, categorize, extract information from the documents with OCR, validate, and route to other systems. The resources also said UiPath Document Understanding combines OCR, ML, NLP and RPA to classify, extract and validate data from structured, semi-structured and unstructured documents.

The data was then cleaned up and populated in a standard reconciliation file after extraction. Standardizing invoice numbers meant removing extra spaces, changing the case of the text, removing special characters and fixing separators that didn't work right. Dates were converted to a common format and month and year fields were created for reporting. Values in currencies and payment amounts were converted to numbers and rules for rounding and tolerance limits were applied to account for small variations in amounts. Master files and validation tables were used to standardize payer names, partner names and client names. Duplicate rows were removed at the staging level using invoice number, payment amount, payment date, bank reference, and payer identifier as composite checking fields.

Ultimately, the integration of UiPath Document Understanding with machine learning created a robust foundation for anomaly detection, exception handling, and profitability analysis.

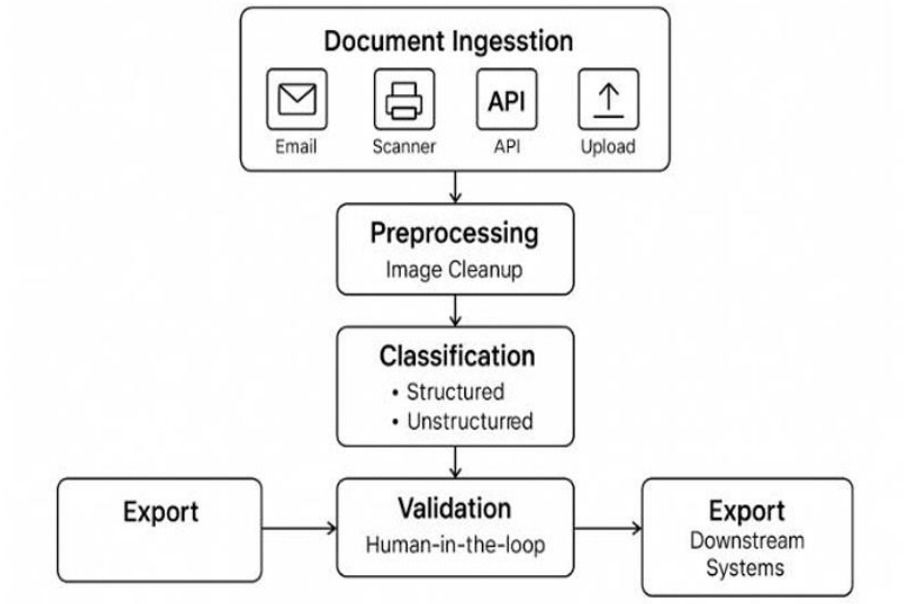


Figure 3: AI driven Document Processing framework

The Workflow Analyzer acts as the coordinator between the Project Parser and the XAML Parser to produce a complete project analysis. It is responsible for:

1. File discovery – Recursively search the project directory for all .xaml files, ignoring temporary and version control directories;
2. Coordinated parsing : invokes the Project Parser and XAML Parser for each file found, merges results into a single data structure;
3. Dependency graph construction: creates a directed graph that associates each workflow with the set of workflows it calls;
4. Execution order determination: From the main entry point workflow, the analyzer performs a depth-first traversal of the dependency graph to determine a plausible execution order;
5. REFramework analysis: If the project is identified as a REFramework implementation, the analyzer classifies workflows into their framework roles and identifies configuration files;
6. Statistics computation: calculates aggregate metrics such as the total number of workflows, variables, arguments, activities, and error handlers, as well as a frequency distribution of activity types.

Its output is a comprehensive JSON-serializable dictionary containing all extracted information, which serves as the input to the generation layer. Figure 4 shows the top-level structure of this output and the information contained in each section.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155280

Key	Content
`project_info`	Name, version, description, entry point, framework, expression language, dependencies
`workflows`	List of per-workflow analysis results (type, variables, arguments, activities, error handlers)
`dependency_graph`	Adjacency list mapping each workflow to its invoked sub-workflows
`execution_order`	Ordered list of workflows based on dependency traversal
`statistics`	Aggregate counts: total workflows, variables, arguments, activities, error handlers, activity frequency
`reframework_info`	(If applicable) Framework role assignments and configuration file references

Figure 4: Workflow Analyzer output dictionary

After recognizing a legitimate image file fi ∈ I, the system utilizes an optical character recognition engine Ok to extract textual data T from the picture. Both Tesseract (O1) and docTR (O2) may be used as our OCR engine. This is the result of extracting the text T:

$$\mathbf{T} = \mathcal{O}_k(f_i), \quad k \in \{1, 2\}$$

The OCR engine sends the raw text T to a Large Language Model (LLM) L, which then transforms it into structured JSON data J. Here is the representation of this process:

$$\mathbf{J} = \mathcal{L}(\mathbf{T})$$

The structured JSON makes that the data is in the right format for the reporting and database tools in the system. Information Collecting and Report Writing Using a mapping function, the structured JSON data is transformed into an Excel sheet E:

$$\mathcal{E} = \mathcal{M}(\mathbf{J})$$

The Excel sheet data is then used to generate a formatted Word document W:

$$\mathcal{W} = \mathcal{F}_{\text{doc}}(\mathcal{E})$$

With the help of the algorithm (refer to Algorithm 1), the system runs in a never-ending loop, processing new files as they come in:

Algorithm 1

Initialize monitoring of directory $\mathcal{D}$
while True **do**
  Check for new files in $\mathcal{D}$
  if new file f_i is detected **then**
    if $f_i \in \mathcal{I}$ **then** ▷ Check if f_i is an image
      Extract text data $\mathbf{T}$ using OCR engine $\mathcal{O}_k(f_i)$
      Send $\mathbf{T}$ to LLM $\mathcal{L}$ to create structured JSON $\mathbf{J}$
      Save $\mathbf{J}$ to the database DB
      Populate Excel sheet $\mathcal{E}$ and generate Word report $\mathcal{W}$
    else
      Ignore non-image file
    end if
  end if
end while

Figure 5: Proposed UiRPA Algorithm

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155280

The bot output was generated in the form of invoice wise Excel sheets and a consolidated reconciliation report. Each invoice wise sheet included payment details, invoice number, associate name, client name, billing status, receipt date, amount, currency, month, year, match status, and remarks. A summary report was also prepared with key reconciliation indicators such as total records processed, matched records, unmatched records, duplicate records, exception count, match percentage, processing time, and validation status. Exception registers were maintained separately for business exceptions and system exceptions. All input files, output files, logs, screenshots where required and exception reports were archived in predetermined retention folders for audit traceability.

The testing was done in steps. Each workflow was unit tested for the email parsing, OCR extraction, file validation, Excel joins, data normalization, LINQ filtering, matching rules and output generation. Integration testing was then performed to check the complete flow from input acquisition to final report generation. User Acceptance Testing was done by comparing the reconciliation outputs generated by the bot to reconciliation results prepared manually. The comparison was made to verify whether the robot has extracted, matched, validated and reported the payment records correctly.

The multi-stage testing strategy ensured that the reconciliation bot was not only technically sound but also aligned with business requirements, thereby establishing confidence in its deployment for enterprise finance operations.

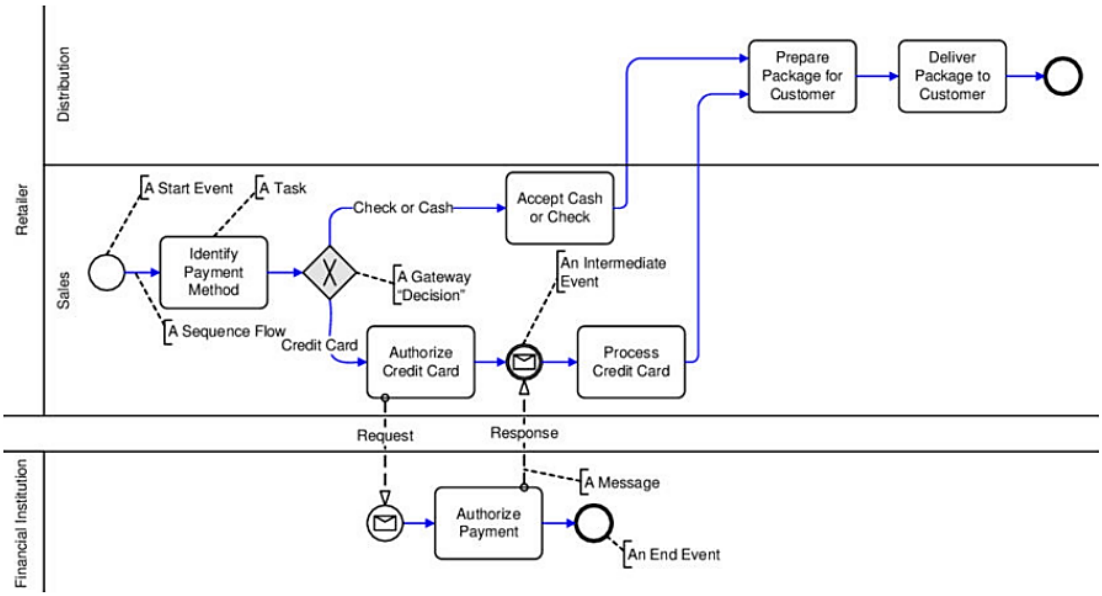


Figure 6: Proposed UiRPA Workflow

The proposed system was evaluated against both technical and business metrics. Technical metrics included OCR extraction accuracy, bot execution time, processing time per transaction, throughput, retry count, file processing success rate and bot error rate. Business metrics included match rate, allocation accuracy, exception rate, duplicate reduction, false positive rate, false negative rate, reconciliation cycle time, manual effort reduction, and audit trail completeness. Auto match rate, precision, recall, exception rate, mean time to reconcile, throughput and bot error rate were also used as evaluation indicators for AI RPA based payment reconciliation systems in prior reconciliation literature.

III. RESULTS AND DISCUSSION

The sample invoice is the type of document used to test the OCR driven payment reconciliation process. It contains important fields such as seller information, buyer information, invoice number, date, item description, quantity, unit price, subtotal, tax, and total due. These fields are shown in Figure 8, where the robot identified and extracted the invoice information and transposed it into structured Excel output for reconciliation. This result lends support to the paper's method, where OCR, UiPath activities, and validation logic converted invoice data into usable records for automated matching and reporting.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155280

AMAZON INVOICE
Seller: Aspiring Watches
Address: One Cambridge Centre,
Pune, 02014 INVOICE # 1
Phone: 8655464627 DATE: 21/02/2020

SHIP TO:
Atharva Pandit
B, Shiv Smruti Bldg, Shiv Shakti Comp, Next to Sarvoday
Mall, Kalyan (W)-421301
PHONE: 8452033685
EMAIL: atharvapandit98@gmail.com

SALESPERSON P.O. NUMBER REQUISITIONER SHIPPED VIA F.O.B. POINT TERMS

65464 - - Due on
R. Kotian receipt

QUANTITY DESCRIPTION UNIT PRICE TOTAL

1 Men's Watch 1200 1200

SUBTOTAL 1200

SALES TAX 800

SHIPPING & HANDLING 0

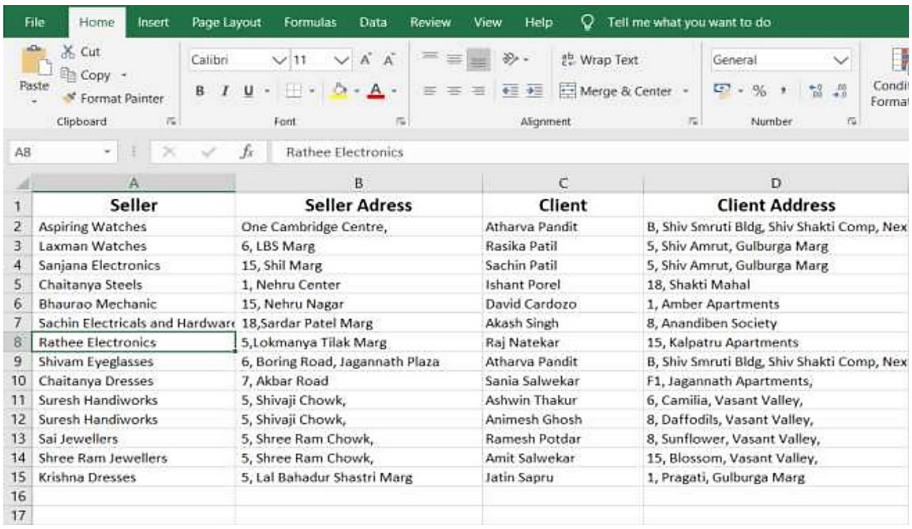
TOTAL DUE 2000

Make all checks payable to AMAZON

THANK YOU FOR YOUR BUSINESS!

Figure 8: Sample Invoice generated

The Excel output shows the details extracted from the invoice have been successfully converted into a structured tabular format for further reconciliation. Figure 9 shows key fields including seller, seller address, client and client address. It can be seen that the automation has structured the unstructured invoice information into usable columns in the spreadsheet. This result supports the OCR driven workflow as the data extracted was not left as raw text but was structured for validation, matching and reporting within the UiPath based reconciliation process.



File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Cut Copy Format Painter Paste Clipboard | Calibri 11 Font | Wrap Text Merge & Center Alignment | General Number

A8 Rathee Electronics

	A	B	C	D
1	Seller	Seller Adress	Client	Client Address
2	Aspiring Watches	One Cambridge Centre,	Atharva Pandit	B, Shiv Smruti Bldg, Shiv Shakti Comp, Nex
3	Laxman Watches	6, LBS Marg	Rasika Patil	5, Shiv Amrut, Gulburga Marg
4	Sanjana Electronics	15, Shil Marg	Sachin Patil	5, Shiv Amrut, Gulburga Marg
5	Chaitanya Steels	1, Nehru Center	Ishant Porel	18, Shakti Mahal
6	Bhaurao Mechanic	15, Nehru Nagar	David Cardozo	1, Amber Apartments
7	Sachin Electricals and Hardware	18,Sardar Patel Marg	Akash Singh	8, Anandiben Society
8	Rathee Electronics	5,Lokmanya Tilak Marg	Raj Natekar	15, Kalpatru Apartments
9	Shivam Eyeglasses	6, Boring Road, Jagannath Plaza	Atharva Pandit	B, Shiv Smruti Bldg, Shiv Shakti Comp, Nex
10	Chaitanya Dresses	7, Akbar Road	Sania Salwekar	F1, Jagannath Apartments,
11	Suresh Handiworks	5, Shivaji Chowk,	Ashwin Thakur	6, Camilia, Vasant Valley,
12	Suresh Handiworks	5, Shivaji Chowk,	Animesh Ghosh	8, Daffodils, Vasant Valley,
13	Sai Jewellers	5, Shree Ram Chowk,	Ramesh Potdar	8, Sunflower, Vasant Valley,
14	Shree Ram Jewellers	5, Shree Ram Chowk,	Amit Salwekar	15, Blossom, Vasant Valley,
15	Krishna Dresses	5, Lal Bahadur Shastri Marg	Jatin Sapru	1, Pragati, Gulburga Marg
16				
17				

Figure 9: Excel file created.

The KPI comparison shows that there are clear operational benefits post UiRPA implementation in terms of finance, order processing, inventory reconciliation and invoice matching. Table 1 shows that the invoice processing time was reduced from 7 minutes to 1.5 minutes per invoice or by 78.57%. Order entry preparation time was reduced by 65%. Inventory Payment reconciliation improved from 6 days to 2 days per cycle – Faster closing of payment records. The invoice matching error rate was reduced significantly from 15% to 1.50% indicating better accuracy. The operational cost also reduced by 20 to 25% savings confirming that automation boosted speed, accuracy and cost efficiency.

TABLE I KEY PERFORMANCE INDICATORS (KPIS) IMPROVEMENT THROUGH UIRPA IMPLEMENTATION

KPI	Before RPA	After RPA	Improvement (%)
Invoice Processing Time	7 minutes/invoice	1.5 minutes/invoice	78.57% reduction
Order Entry preparation Time	8 minutes/order	2.8 minutes/order	65% reduction
Inventory payment Reconciliation	6 days/cycle	2 days/cycle	66.67% reduction
Error Rate	20%	2.00%	90% reduction
Operational Cost	Baseline cost	40-45% cost savings	40-45% reduction

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155280

The KPI progress graph shows visually that UiRPA made improvements that can be measured in all five performance fields. The bars in Figure 10 that display "before" and "after" show that the time it takes to process invoices, enter orders, balance inventory, and make mistakes when comparing invoices has gone down a lot. Biggest gain was a 90% reduction in number of errors, which shows tech improved matching accuracy and reduced human errors. Processing of invoices became a lot faster, by almost 78.57%. Improved order entry and inventory accounting by 66.67% and 65% respectively. The cost savings bar also shows that automation decreased running costs by 20 to 25 percent, which supports the idea that the proposed system would work well.

These measurable improvements validate the effectiveness of UiRPA in transforming financial operations, demonstrating that automation can simultaneously enhance accuracy, reduce costs, and accelerate cycle times, thereby offering a replicable model for enterprise adoption.

Figure 10: RPA KPI improvement chart showing before and after metrics across five categories

The performance graph shows the dataset was processed by the UiRPA system with good processing capacity while tracking key operational measures. Throughput in Figure 11 has the highest value, which is almost 40,000 records or actions. This indicates the robot can process a lot of reconciliation data. The other metrics such as auto match rate, exception rate, false positive rate, ERP posting latency and bot error rate are much lower on the scale, indicating controlled error behaviour and stable execution. This supports the value of UiRPA for high-volume reconciliation, where speed, consistency, exception monitoring and reduced manual intervention are key.

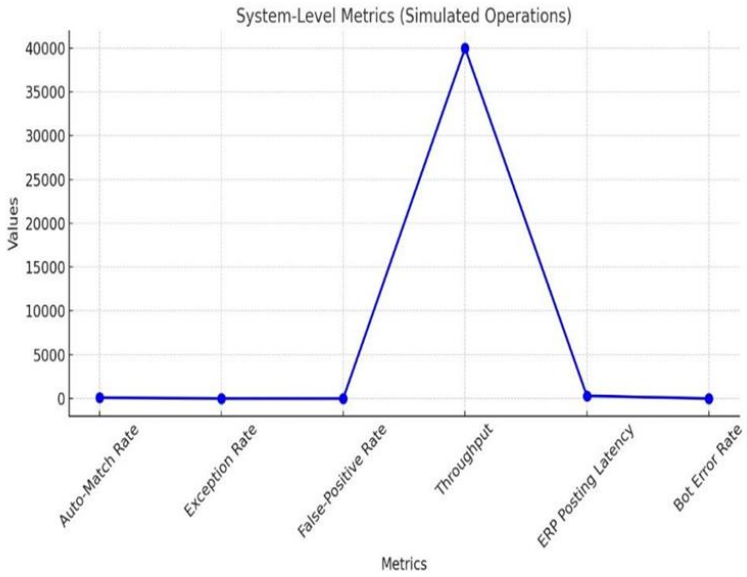


Figure 11: UiRPA performance with the dataset

DOI: 10.17148/IJARCCE.2026.155280

IV. CONCLUSION AND FUTURE SCOPE

The study revealed that the proposed UiRPA based payment reconciliation system improved the speed, accuracy and reliability of invoice and payment processing. The automation successfully transformed invoice data into structured Excel records and enabled validation, matching and reporting via OCR, UiPath workflows and machine learning-based anomaly detection. The synthetic ERP data with invoices, payments, credits, adjustments, missing values, wrong exchange rates and rounding errors was used to test the system against realistic reconciliation problems. The results confirmed that automation lowered invoice processing time from 7 minutes to 1.5 minutes per invoice and order entry preparation time from 8 minutes to 2.8 minutes per order. Inventory payment reconciliation efficiency increased from 6 to 2 days per cycle. Invoice matching errors reduced from 15% to 1.50%. The system resulted in operational cost savings of 20 to 25%. These findings showed that UiRPA can facilitate high volume reconciliation by enhancing throughput, minimizing manual intervention, managing exceptions and creating audit ready outputs for finance operations.

Future work can extend this framework by integrating real-time ERP connectors, advanced anomaly detection models, and predictive analytics to forecast payment discrepancies before they occur. Expanding the system to multi-currency, multi-lingual, and cross-border reconciliation scenarios would further validate its scalability. Additionally, embedding blockchain-based audit trails and explainable AI modules could strengthen compliance, transparency, and trust in automated financial operations.

REFERENCES

[1]. B.Bhuvaneswaran, Ajitha.B, & A, A. S. (2025). Agentic AI Driven - Automated Invoice Processing with Intelligent Document Understanding. *2025 4Th International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 1714–1720. https://doi.org/10.1109/icacrs67045.2025.11324121

[2]. Anghiuș, I. F., & Vălean, H. (2025). Intelligent Document Processing: Methods for Automated Data Extraction. *2025 29Th International Conference on System Theory, Control and Computing (ICSTCC)*, 37–42. https://doi.org/10.1109/icstcc66753.2025.11240355

[3]. Jia, C., Wulin, L., Run, Y., Yuehui, L., Xu, Y., & Qu, F. (2025). A UIPath-based Approach for Automated Invoice Data Capture:A Comparative Study of Google Cloud Vision, OmniPage, and Tesseract OCR. *2025 International Conference on Metaverse and Current Trends in Computing (ICMCTC)*, 1–9. https://doi.org/10.1109/icmctc62214.2025.11196129

[4]. Soni, K., Luthra, M., Soni, S., & Tiwary, G. (2026). Intelligent Document Workflows. *Natural Language Processing for Business and Organizations*, 146–162. https://doi.org/10.1201/9781032658056-10

[5]. Kothari, U. End-to-End Robotic Process Automation in Oracle On-Premise Environments: UiPath RPA for ERP, EBS, PeopleSoft, and HCM in Construction Materials Manufacturing. *IJLRP-International Journal of Leading Research Publication*, *5*(11).

[6]. Kavitha, T., Saraswathi, S., & Senbagavalli, G. (2024). Journey To Hyperautomation. *Hyperautomation for Next-Generation Industries*, 1–34. https://doi.org/10.1002/9781394186518.ch1

[7]. AlNaaji, H. M. M. (2022). *Automating unauthorized access attempts detection and handling using robotic process automation* (Master's thesis, Princess Sumaya University for Technology (Jordan)).

[8]. Shidaganti, G., Karthik, K. N., Anvith, & Kantikar, N. A. (2023). Integration of RPA and AI in industry 4.0. In *Smart innovation, systems and technologies* (pp. 267–288). https://doi.org/10.1007/978-981-19-8296-5_11

[9]. Varfa, R., Khare, U., Gupta, A., & Sharma, G. (2001). Robotic Process Automation: Chatbot integration for task automation (Banking Industry). *Grenze International Journal of Engineering & Technology (GIJET)*, *9*(2), 648. https://doi.org/,"

[10]. Ladeiras, J. P., & Martins, A. (2024). Robotic Process Automation. *Digital Transformation and Enterprise Information Systems*, 1–17. https://doi.org/10.1201/9781003461821-1

[11]. Ding, Y., Tang, Y., Liu, H., & Li, T. (2026). Overview of research on robotic process automation: application status, key technologies, and development trends. In *Lecture notes in networks and systems* (pp. 41–61). https://doi.org/10.1007/978-3-032-16794-1_4

[12]. Peng, J. (2026). Document design and AI-driven data extraction: an exploratory study. *International Journal of Services and Standards*, *15*(3), 147–159. https://doi.org/10.1504/ijss.2026.153419

[13]. Mohapatra, B., Mohapatra, S., & Mohapatra, S. (2023). Fundamentals of RPA. *Process Automation Strategy in Services, Manufacturing and Construction*, 123–166. https://doi.org/10.1108/978-1-80455-143-120231014

[14]. Al-E'mari, S., Sanjalawe, Y., & Al-E'mari, A. (2025). The Role of Artificial Intelligence in Enhancing financial Decision-Making and Administrative Efficiency: A Systematic review. *Al-Basaer Journal of Business Research*. https://doi.org/10.71202/paper21

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155280

[15]. Nayak, A., Satpathy, I., Patnaik, B. C. M., Gujrati, R., & Uygun, H. (2023). Simplified Hospital Management System. *Data-Centric AI* Solutions and Emerging Technologies in the Healthcare Ecosystem, 281–302. https://doi.org/10.1201/9781003356189-17

[16]. Ge, L. (2024). Enhancing financial audit efficiency through RPA Implementation: A Comparative analysis in manufacturing industry. Journal of Computing Innovations and Applications, 2(1), 62–73. https://doi.org/10.63575/cia.2024.20106

[17]. Akshay, B., Swamy, K. M., Prakash, J., & Khan, Z. A. (2025). Invoice Processing Using OCR Methodology in UiPath Robotic Process Automation. International Journal of Innovative Research in Technology, 12(5), 45–52.

[18]. Dineshkumar, V., & Shylaja, S. (2026). Automated Invoice Processing Using UiPath and SQLite. International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 15(3), 321–327. https://doi.org/10.17148/IJARCCE.2026.15321 (doi.org in Bing)

[19]. Mahmud, D., & Ikbal, M. Z. (2024). Power BI and Data Analytics in Financial Reporting: A Review of Real-Time Dashboarding and Predictive Business Intelligence Tools. International Journal of Scientific Interdisciplinary Research, 5(2), 125–157. https://doi.org/10.63125/yg9zxt61

[20]. AutomationEdge. (2025). Bank Reconciliation Automation: Transforming Financial Processes. AutomationEdge Whitepaper, 1–12.

[21]. IBM. (2025). Five Ways to Use RPA in Finance. IBM Research Insights, 1–10.

