Mathematical Optimization of AI-Based Document Processing Workflows Using Markov Decision Processes

Ranadheer Reddy Charabuddi
Sr. Lead SAP Opentext VIM Consultant, Avventis Inc, Northlake, Texas, USA

Abstract

Conventional models tend to malfunction with stochastic task arrivals, indefinite processing times, and priority conflicts. The study presents a scalable and wise optimization model based on Markov Decision Process (MDP) along with Deep Reinforcement Learning (DRL) as Proximal Policy Optimization (PPO) to acquire optimal policies for document routing. The OCR Receipts Dataset from Kaggle (2023) is used that contains annotated receipt images. Major techniques involve feature vector encoding, filtering out noise, and balance in reward of latency, accuracy, and cost. Python implementation involves state-action modeling and multi-objective optimization to enhance task scheduling, usage of resources, and decision-making. There is improved accuracy, less processing latency, and adaptive performance compared to baseline models. The framework supports scalable automation of finance, law, and the public sector. On top of that, the PPO agent has strong learning abilities across varying workflow circumstances. The suggested system is up-scalable to other areas which need an intelligent, real-time processing of documents and routing tasks.

Key words: Markov Decision Process, Deep Reinforcement Learning, Document Workflow, PPO, OCR Receipts Dataset

1.Introduction

Over the past few years, artificial intelligence (AI) has turned and transformed the functioning of enterprises remarkably as it automated document-based tasks like verification of invoices, extraction of data out of contracts, digitization of forms, and classification of emails[1], [2], [3]. Through these smart systems, efficiency and the workload has been placed on hands[4]. Nevertheless, these developments raise concerns over the capabilities of AI-powered document processing pipelines in terms of dynamic loads, uncertain input format, and priorities in the tasks. Such aspects might cause resource congestions, higher delay and performance fluctuations[5]. Less flexible and not real-time adapting, traditional workflow management systems are based on the rule-based logic or heuristic-based approaches and, therefore, have little capability to adjust themselves to changing enterprise needs[6]. Consequently, they may tend to poorly perform in real life complex situations[5]. The lack of these has led to the current interest by researchers to incorporate the use of mathematical modeling, reinforcement learning, and sequential decision-making models[7]. Such strategies will produce adaptive and resilient document processing streamlines which are intelligent in both resource allocation and priorities and which will be able to deliver optimal throughput in dynamic enterprise environments[8].

Some of the latest findings discussed the application of powerful AI models in document automation[9]. As an example, Wang et al[10]. (2023) introduced a transformer-based model that makes a cost-effective use of unstructured data in invoices and beats rule-based systems in generalization and speed. In the same vein, Zhang and Liu[11] (2024) showed that reinforcement learning algorithms especially Deep Q-Networks (DQN) could be applied to optimize multi-priority document streams routing schemes dynamically[12]. Kim et al.[13] (2023) in another study found a hybrid approach to the use of OCR preprocessing and BiLSTM-CRF to classify the legal documents after performing OCR: their method presented high accuracy despite noisy data[14]. These surveys confirm the trend of the movement towards intelligent, data-driven solutions or away from the traditional static pipelines and point to the increasing success of such approaches like MDP or DRL in overcoming the issues of scalability, uncertainty, and adaptive controls in enterprise document management[15].

1.1 Research Motivation

Decision-making at real-time about task ordering, resource distribution and exception management should be supported by document processing systems. Such decisions are usually interdependent, and they have to be optimized in a wholesome manner over time, not as individual cases of rule-based thinking. Most of the approaches have not however been able to model long term dependencies and uncertainty. MDPs and their deep extension Reinforcement Learning (DRL) provide an encouraging basis to be able to model this sort of sequential and stochastic environments. They also have the capacity to allow systems to learn the best policies they can, by interacting with the work flow environment, and changing dynamically, due to changing work loads, trying out various types of documents, or the limitations of the system. This stimulates creation of a framework based on AI driven by MDPs that can be exploited to mathematically optimize document workflow decisions within the real world uncertainty.

1.2 Problem Statement

The existing frameworks of AI-assisted document processing cannot cope with stochastic arrivals of tasks, processing times that are imperfectly known, and conflicts in priorities[16]. In a dynamic environment static scheduling or rule-based handling of tasks is not enough to guarantee the best utilisation of resources and latency[17]. In addition, there are no smart control systems that can analyze past patterns of activities and use them to make proactive actions. Such restrictions require a powerful framework which can make the workflow of document processing a decision making problem, sequential in nature and optimize it through smart policies, which are learnt[18]. Thus, the aim of this study will be to build up the mathematical framework of optimization which implements an optimization framework based on the Markov Decision Process with Deep Reinforcement Learning mechanism designed to optimize complex workflows of AI-based document processing tasks.

1.3 Research Significance

The proposed solution is based on the framework that proposes the principled resolvable and scalable solution to a critical enterprise problem the dynamic document workflow optimization issue. In contrast to the rule-based approaches, the study involves the utilization of the math background of MDPs in describing the setting and DRL in learning flexible policies during interaction. Adding multi-objective optimization, such a system can also balance trade-offs in cost, latency, and accuracy, making it more appropriate to be deployed in a real enterprise. The findings of the research have important implications on industries like finance, legal, insurance and public administration where document processing would have immense use on efficiency of operations.

The rest of this paper is structured as follows: Section 2 provides the literature review of the intelligent document processing, Markov Decision Processes and reinforcement learning based workflow optimization. The section 3 outlines the suggested methodology, namely the MDP formulation of the document workflow, the policy network design based on deep reinforcement learning, and the inclusion of the multi-objective optimization strategies. In section 4, details of the experimental environment, data sets, performance measures and comparison of the proposed model with the current methods is given. Lastly, Section 5 will end the paper by summarization of major findings, mentioning contribution, and providing ideas on potential future research and application to the real world.

2. Related Works

Yang et al.,[19] proposed the deep reinforcement learning powered intelligent scheduling system to solve the dynamic permutation flowshop scheduling problem (PFSP) on a real-time basis. The authors proposed a system structure that helps to reduce the total cost of tardiness with the help of applying the Advantage Actor-Critic (A2C) algorithm and making smart decisions. A mathematical model is made and such parameters as state characteristics, actions, and rewards functions were well-designed. The training process allowed the learning based agent to develop efficient scheduling responses that are made in an almost instant 2.16 ms on an average. Large-scale experiments showed better solutions quality and CPU time, used excessive and superior capabilities as compared to the other types of DRL models and meta-heuristics. The method can, however, be limited in large-industry applications such as the threat of system complexity and a prolonged training period affecting the scaling of the method.

Sheng et al.,[20] proposed a task scheduling algorithm based on deep reinforcement learning (DRL) specialized on edge computing (EC) to solve the problem of resource limitations and latency imposed to edge computing by IoT applications. Task scheduling problem is expressed as a Markov Decision Process (MDP) problem, that is, there are well-defined states, transitions, actions, and rewards to optimize long-term task satisfaction degree (LTSD). The policy-based REINFORCE algorithm is entailed with the use of a fully-connected neural network (FCN) to extract features. This approach takes care of not only the order of execution of tasks but also deployment of resources to the different virtual machines. The results of simulation show a better average task satisfaction and the success ratio than current methods. The considerable use of the REINFORCE algorithm can however restrict the rate and stability of convergence in more complex or highly dynamic EC situations.

Peng et al.,[21] introduced the intelligent task offloading strategy applied in this research is based on Multi-access Edge Computing (MEC) scenario, where the targeted applications contain interdependent applications. We formulate an optimization problem into Markov Decision Process (MDP) and the dependency of the tasks are modeled into Directed Acyclic Graphs (DAGs). Its authors proposed an adaptive offloading decision based on Deep Reinforcement Learning (DRL) strategy where they proposed a Deep Reinforcement Learning algorithm termed the Soft Actor Critic (SAC) algorithm to make such decisions in dynamic multi-user, multi-server settings. The centralized scheme is much more effective in reducing service delay and consumption of energy given limited resources. The experimental data prove better convergence and performance of overall solutions than the current solutions available. Nevertheless, the centrally-designed control can become a barrier to scalability and flexibility in the completely differentiated MEC systems.

(Zhang et al.,[22] developed a multi-agent production system based on Deep Reinforcement Learning (DRL) to improve dynamic responsiveness and self re-configuration in a personalized production system. The authors suggested a system in which equipment are autonomous agents with the aid of edge computing where it was coordinated using an enhanced Contract Net Protocol (CNP). An AI scheduler, consisting of a multi-layer perceptron, trained by the Proximal Policy Optimization (PPO) algorithm, makes task assignment an intelligent process because the scheduler acquires the workshop state at a given moment in time. Your most likely unforeseen events such as job insertions and machine failures are effectively dealt with using this system. Through experimentation, better performance of scheduling and resistance to disturbances caused by dynamics was proved. Nevertheless, the dependence on periodic training can cause the computational overhead during the real-time computing process.

(Dornheim et al.,[23] proposed the model-free deep reinforcement learning (DRL) approach in this study plans to optimize the processing path in material design based on modeling free inputs to attain target material structures leading to the development of desired properties. The approach specializes in learning the best transitions within the space of material structures with the help of structural properties and a spatial distance-based rewarding function toward the target. It can already do real-time adaptive pathfinding based not on pre-sampled information or prior information about a particular process. The strategy offers single-target and multi-target optimization through learning efficient attainable objectives. Simulation of results in metal forming proves the usefulness and flexibility of the approach. But model-free learning is expensive because of trail-and-error learning at the early stages of exploration.

Wang et al.,[24] proposed a new concept known as Temporal Error-based Adaptive Exploration (TEAE) to circumvent the drawbacks of conventional backward recursion algorithms to solve Markov Decision Processes (MDP). Proposed a reinforcement learning algorithm that can dynamical changing exploration probabilities and approximate optimal expected payoff solutions by deep convolutional neural networks. Generalised TEAE model to DQN-PER-TEAE and DDQN-PER-TEAE so as to show that it can be integrated with already available reinforcement learning methods. Simulated and case-study verified the effectiveness of the suggested model to multiple types of performance metrics. Obtained good decision-making efficiency and performance of complex MDPs. Nevertheless, there could be more algorithms involved in the implementation due to the involvement of multiplexation in the integration of the two networks and adaptation process.

Shao & Li,[25] proposed mobile edge computing (MEC) architecture based on SDN that can be used to aim at maximizing system utility through optimal offloading decisions and resource allocation. State-

optimized the dynamic modeled a Markov Decision Process (MDP) and proposed reinforcement learning (RL) algorithm to optimize dynamic utility. Does all the work on a deep reinforcement learning (DRL) based algorithm that can scale to large continuous state spaces, and has better performance in multiple measures of decision-making performance. Has produced great efficiency improvements with an increase in user and operator utility of 12.2 and 22.4 percent respectively. Proved the usefulness of intelligent learning-based approaches to adaptive design of MEC networks. The method however has a very high computing overhead as DRL is complicated.

Arcieri et al.,[4] proposed a deep reinforcement learning algorithm with combined it to address Partially Observable Markov decision-making processes (POMDPs) under uncertainty in transition and observation dynamics. Offered a way of jointly estimating these model parameters by Markov Chain Monte Carlo (MCMC) sampling of a hidden Markov model condited on actions. Created strong solutions by addressing the problem of parameter uncertainty using domain randomization that was instilled in the learning process. Comparisons against model-free methods where Transformers and LSTM have been used to compare a belief-input approach, where the inferred POMDP models are used as building blocks in belief space reasoning. Obtained enhanced performance of decision making in a real world railway maintenance planning problem, with a large saving of life-cycle costs compared to current policies. But as the framework used the MCMC and belief inference, it may lead to immense computational complexity and scalability constraint.

Zhou et al.,[26] proposed a complete deep reinforcement learning architecture to tackle Permutation Flow Shop Scheduling Problem (PFSP) with a view of minimizing makespan. Exploited the modeling of PFSP as a Markov Decision Process and proposed disjunctive graphs in order to provide a description of state information allowing the infusion of topology. Prepared a policy model based on graph isomorphism network and trained with proximal policy optimization, to know the good schedules. Shown to outperform six baselines methods on both randomly generated datasets as well as on the Taillard benchmark. Accomplished considerable savings of makespan (down to 188.4 hours) and computation time (down to 18.7 seconds). Nevertheless, the use of complex architectures based on graphs when training the model might slow real-time and industrial scalability and interpretability.

3 Proposed Method for Adaptive Document Workflow Optimization Using MDP and Deep Reinforcement Learning

A framework of intelligent document workflow optimization based on Markov Decision Process (MDP) and Deep Reinforcement Learning (DRL). Normalization, noise suppression, text region identification, as well as feature encoding are applied to the OCR Receipts Dataset to produce structured state representations. Such representations, which include document complexity, density and layout properties are adopted to model the workflow as an MDP, where actions, including assignment, escalation or reprocessing are taken on a case-by-case basis. A DRL agent is Proximal Policy Optimization (PPO) trained and learns to optimize action strategies maximizing long-term rewards exposed repeatedly to the environment. Reward function takes into account the objectives of the enterprise through a balanced accuracy, latency of processing, and cost of operations. This will result in effective routing of tasks and active decision-making as a response to document variability and uncertainty elements in the real world. The approach is highly flexible, interpretable, and robust, which allows the application even in the scalable and automated enterprise settings of documents processing.

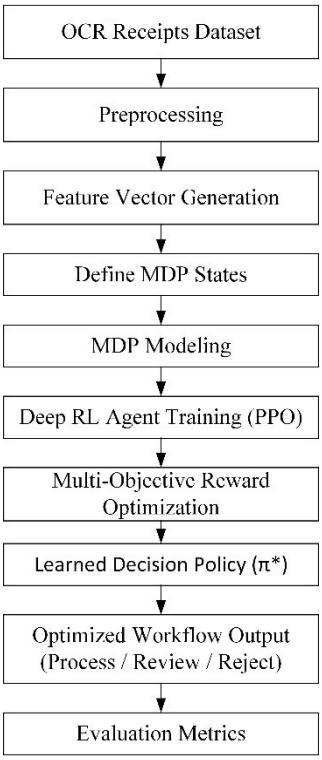

Fig 1: Adaptive Document Processing Workflow Using MDP and Deep Reinforcement Learning

This Fig 1 represents the work-flow at end-to-end of the proposed system beginning with OCR receipt data to MDP modelling, PPO-based, learning, policy learning, decision output and the final evaluation measures.

3.1 Data collection

The dataset that was used in this experiment is OCR Receipts Text Detection Dataset located at Kaggle (2023). It is a special collection created to support Optical Character Recognition (OCR) and text localization procedures. It includes many receipt images in the real world with rich annotations in JSON format that specify locations and text content of every part of text. The receipts are diverse in their layout, font, language, noise, and illumination, which contributes to realistic and difficult dataset to smart documents analysis. Such differences provide a perfect training data source to induce deep learning models to learn to better identify and recognize text in visual scenes. The structured format of the annotation of the dataset makes it compatible with any modern OCR engine and evaluation tool and comparatively accurate measures of detection performance can thus be obtained. Its intricacy and unpredictability favors it best when it comes to building and experimenting Markov Decision Process (MDP) and Deep Reinforcement Learning (DRL) applied solutions in the field of adaptive document processing workflows. This sample data underlies the process of modeling smart decisions on OCR systems.

3.2 Data preprocessing

In order to create a strong and flexible modeling of intelligent document workflows, most of the preprocessing pipelines have been implemented to the OCR Receipts Dataset. In this pipeline, image normalization is done to make the image look standard in terms of brightness and contrast, the Gaussian filter is applied to decrease the noise in the image to improve clarity of texts, and lastly the clearly identified text regions can be extracted using annotated bounding boxes. In addition, the documents are transformed to the structured feature vectors, which present layout complexity, amount of text, language metadata, spatiality dimensions. The processed features have critical roles in defining the state representations under Markov Decision Process (MDP) meaning that the system will take intelligent steps based on the uncertainty nature of the document structures and reality uncertainties.

1. **Image Normalization:**
   To stabilize brightness and contrast, all the images of receipts are resized into the fixed resolution and the pixel values are normalized with z-score normalization method.

$$I_{norm}(x,y) = \frac{I(x,y)-\mu}{\sigma} \quad (1)$$

   In Equation (1), $I(x,y)$ represents the original pixel intensity at position (x,y) in the image. μ denotes the mean pixel intensity across the entire image, and σ is the corresponding standard deviation. $I_{norm}(x,y)$ s the normalized pixel value, ensuring standardized brightness and contrast for improved model convergence.
2. **Noise Reduction:**
   Noise removing Gaussian filtering is done with the purpose of getting clear edges to enable OCR recognition. The overlay Anything in 2D kernel is convoluted across the normalized image.

$$I_{smooth}(x,y) = \sum_{i,j} G(i,j).I_{norm}(x+i,y+j) \quad (2)$$

   In Equation (2), $I_{smooth}(x,y)$ is the denoised pixel value at position (x,y). $G(i,j)$ represents the weight of the Gaussian kernel at offset (i,j) and $I_{norm}(x+i,y+j)$ is the normalized pixel intensity at a neighboring location. The summation performs Gaussian filtering to reduce noise.
3. **Text Region Extraction:**
   JSON annotations include bounding box coordinates of a given text segments. These areas are cropped using such boxes then analyzed.

$$B = \{(x_1,y_1),(x_2,y_2)\} \quad (3)$$

   In Equation (3), B denotes the bounding box used to localize a text region within a receipt image. The coordinates $(x_1,y_1) and (x_2,y_2)$ represent the top-left and bottom-right corners of the box, respectively, defining the rectangular area enclosing the detected text.
4. **Feature Vector Encoding:**
   All of the documents are encoded as a structured vector to measure layout complexities, block densities, the type of language, and dimensions to represent the state of MDPs.

$$F = [n_b, r_d, s_t, l_w, l_h] \quad (4)$$

   In Equation (4), F is the feature vector representing a document's structural attributes. n_b indicates the number of text blocks, r_d is the text region density, s_t denotes the language or script type, and l_w, l_h represent the layout's width and height, respectively.
5. **Label Normalization for Training:**
   Continuous variables such as OCR confidence scores are also min-max normalized to [0,1] so that they do not suffer during training as it relates to reward.

$$v' = \frac{v-v_{min}}{v_{max}-v_{min}} \quad (5)$$

   In Equation (5), V' is the normalized value scaled between 0 and 1.V represents the original raw value (e.g., OCR confidence score), while v_{min} and v_{max} are the minimum and maximum values of V in the dataset. This normalization ensures consistent input scaling.
6. **Dataset Partitioning:**
   The dataset is separated into training set-15 percent validation set-15 per cent and test set-70 per cent based on cross diversity in complexity and type of receipt categories.

$$D = D_{train} \cup D_{val} \cup D_{test},\ |D_{train} = 70\%| \quad (6)$$

   In Equation (6), D represents the complete dataset, which is partitioned into three subsets: D_{train} (training), D_{val} (validation), and D_{test} (testing). The size of D_{train} is set to 70% of the total data, ensuring sufficient data for model learning and generalization.

3.3 Markov Decision Process (MDP) Formulation

The name of the approach that should be proposed is Adaptive Document Workflow Optimization Using MDP and Deep Reinforcement Learning, which refers to the improvement of the efficiency and adaptability of document-centric processing pipelines. The task is formulated as sequential decision making over the document workflow in which the states have attributes of the document like document layout complexity, amount of text content, and document metadata containing language information etc. The system will wisely choose to perform some action: classification, extraction, or escalation relying on the actual state and the memorized policies. It manages loads and uncertainties dynamically (as opposed to fixed rules), by using a reinforcement learning agent whose performance has been learned over time as part of an iterative process, interacting with the environment. A reward system is also adapted in the method in which significant enterprise goals such as accuracy of processing, response time and cost are balanced. Through constant feedback about previous actions and results, the system will be able to optimize document workflows, bottlenecks as well as offering an optimal throughput in complex and real life situations in enterprise environments.

3.4 State Representation and Action Space

A new incoming document is processed into the structural feature vector that conveys important properties of the document like the amount of text lines, text density, type of script, or language, and the size of the document in terms of width and height. This is a feature vector that is used as the input state by reinforcement learning agent. According to the existing state, the agent can choose among a prescribed array of actions, which could include assigning the worker, reprocessing the document, escalation of the document to manual review or drop the document off the pipeline. The system switches states based on the specified action and the results of prior processing activities, which makes such a system follow the changing workflow conveniently.

3.5 Deep Reinforcement Learning Policy Design

An agent of Deep Reinforcement Learning (DRL) is trained with Proximal Policy Optimization (PPO) as a popular stable policy-gradient algorithm, in order to find the best policy to govern document workflows. The agent explores a neural network that approximates the policy of the decision-making and the expectation of every state. The agent learns to formulate actions that yield maximum long-term performance through the trade-off (or summation) between the various end-results of the workflow activities through numerous interactions with the environment. The flexibility of PPOs that allow trading off stability in learning and exploration makes PPOs applicable in the setting of complex and dynamic document processing processes in which conditions commonly change and instability is the norm.

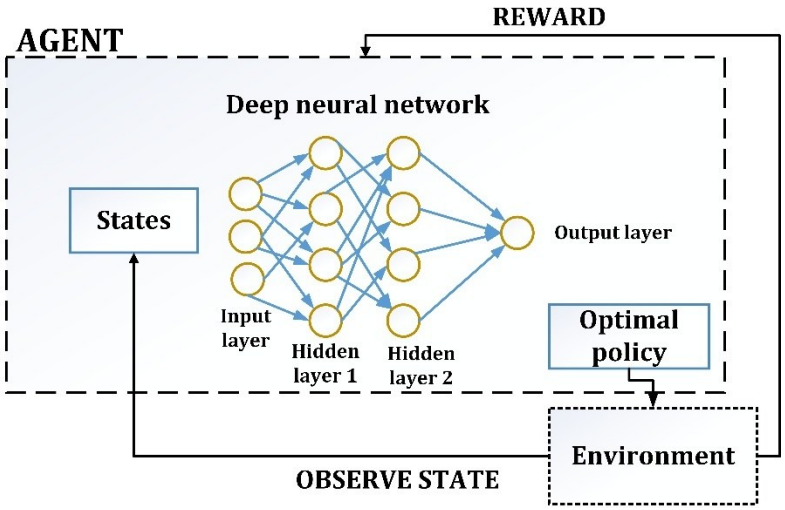

Fig 2 Deep Reinforcement Learning Architecture

This Fig 2 is of a reinforcement learning architecture in which an agent consisting of a deep neural network monitors states of the environment, generates a choice of action through an efficient policy, and optimizes its decisions based on rewards.

3.6 Multi-Objective Reward Design

To obtain a trade-off at enterprise level a multi-objective reward function that balances between latency, cost, and accuracy is applied:

$$R(s_t, a_t) = \propto . R_{accuracy} + \beta . R_{latency} + \delta . R_{cost} \quad (7)$$

The state variables would involve the number of text blocks, density of texts, the type of language that proceeds in the document, width of layout, and the height of the layout. Such characteristics indicate the complexity and format as well as the content of individual documents to be used in decision making, which is in (7).

Algorithm 1: Adaptive Document Workflow Optimization Using MDP and Deep Reinforcement Learning

---

Input:

- OCR Receipts Dataset (scanned documents and annotations)

Output:

- Optimized workflow action: Accept, Rerun, Escalate, Drop

- Learned policy π*

- Evaluation metrics

Step 1: Data Preprocessing

Resize all input images to a fixed resolution.

Normalize pixel values to the range [0, 1].

Apply Gaussian smoothing for noise reduction.

Extract text region patches using bounding box annotations.

Step 2: Feature Extraction

Encode each document as a feature vector:

F = [number of blocks, text density, language type, width, height]

Step 3: MDP Modeling

Define each feature vector as a state in the MDP.

Define action space: assign worker, reprocess, escalate, drop.

Model state transitions based on outcomes of previous steps.

Design a reward function balancing accuracy, latency, and cost.

Step 4: Deep Reinforcement Learning

Initialize policy network using PPO algorithm.

Train the DRL agent using state-action-reward trajectories.

Update policy iteratively based on feedback from environment.

Step 5: Workflow Optimization

Use the learned policy π* to make real-time decisions.

Select the best action dynamically based on document state.

Optimize workflow throughput while minimizing bottlenecks and cost.

The given algorithm 1 is a proposal that will help optimize the document workflows, which are being viewed as a Markov Decision Process. It preprocesses OCR receipts, extracts features from them and trains Deep Reinforcement Learning agent PPO learns dynamic action policies in real-time. The system enhances precision in the processing of documents, delay is minimized, and the system can adjust well to the changing complexities of documents.

4. Result and Discussion

4.1 Document Complexity and Adaptive Workflow Analysis

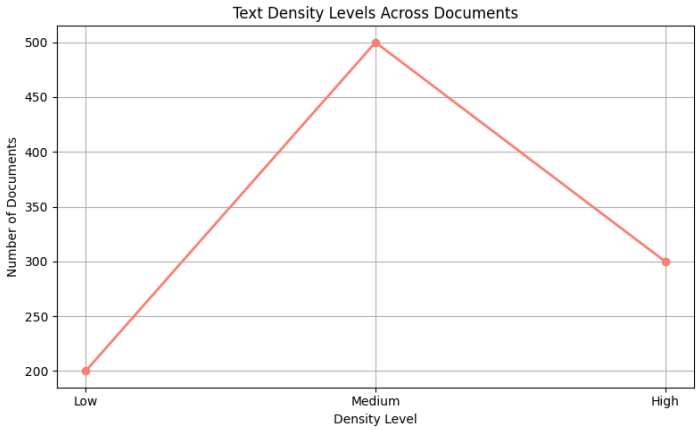

Fig 3: Distribution of Text Density Levels Across Documents

In this line graph the data is presented in terms of text density levels Low, Medium, and High. It enforces the fact that the dataset is heavily biased towards medium-density documents, with high and low-density documentations as the minority. This deviation affects document complexity that affects state representation and decision-making process in the adaptive MDP-based workflow system as depicted in Fig 3.

This bar graph represents the arrangement of the document layout complexity in the data set. It indicates that moderately complex layouts are the most popular type of layouts, and there are equal amounts of simple and complex layouts. Such variations play a pivotal role in defining MDP states and having adaptive decision-making in optimizing workflow as in Fig 4.

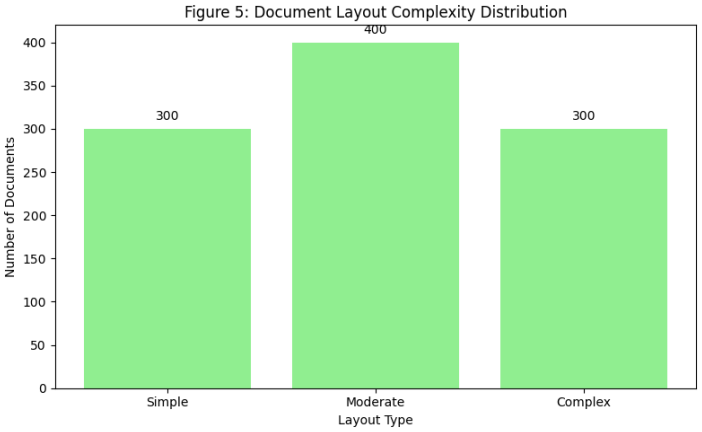

Fig 4: Document Layout Complexity Distribution

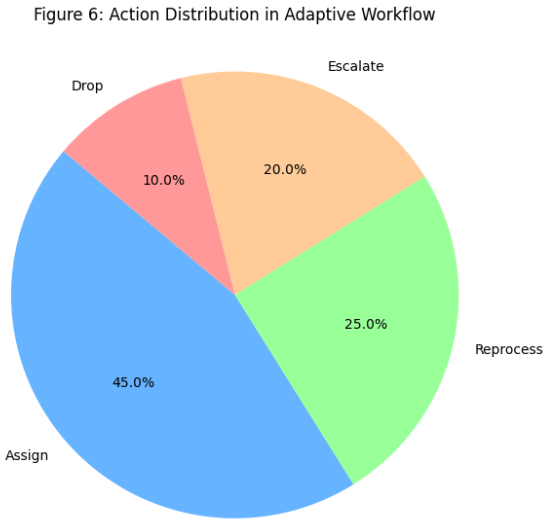

Fig 5: Action Distribution in Adaptive Workflow

The distribution of system actions in adaptive document processing is shown by this pie chart in Fig 5. Most of the documents are directly assigned and a good percentage face reprocessing or escalation. A lesser proportion is dropped off. Such dynamic decision behavior represents the policy of the reinforcement learning agent in the effective management of document routing.

4.2 Performance Metrics

4.2.1 Accuracy

Measures how frequently the PPO agent chooses the correct or best action against a ground truth or expert-labeled workflow decision.

$$Accuracy = \frac{N_{correct}}{N_{total}} \times 100 \tag{8}$$

In Equation (8), $N_{correct}$ represents the number of correct decisions made by the agent, and N_{total} is the total decisions attempted, expressing policy accuracy as a percentage.

4.2.2 Latency

Average time to process one document workflow cycle — encompasses OCR, preprocessing, policy decision, and task execution.

$$Latency = \frac{1}{n}\sum_{i=1}^{N} T_i \tag{9}$$

In Equation (9), T_i denotes the processing time for the i^{th} document, and N is the total number of documents. The equation calculates average document processing latency.

4.2.3 Operational Cost Score

Average cost incurred per document based on workflow actions assigned (e.g., escalation = high cost, auto-process = low cost).

$$Cost_{avg} = \frac{1}{N}\sum_{I=1}^{N} C(a_i) \tag{10}$$

In Equation (10), $C(a_i)$ represents the operational cost of the action (a_i) taken for the i^{th} document, and N is the total number of documents processed.

4.2.4 Cumulative Reward

Cumulative reward obtained by the agent during a single training/testing run, reflecting how well it trades off several goals.

$$R_{cumulative} = \sum_{t=1}^{T} r_t \tag{11}$$

In Equation (11), r_t denotes the reward received at time step t, and T is the total number of steps in an episode. It computes the cumulative reward earned.

Table 1. Performance metrics

Metric	Value
Accuracy	96.8
Average Latency	1.34
Operational Cost Score	2.15
Cumulative Reward	187.6

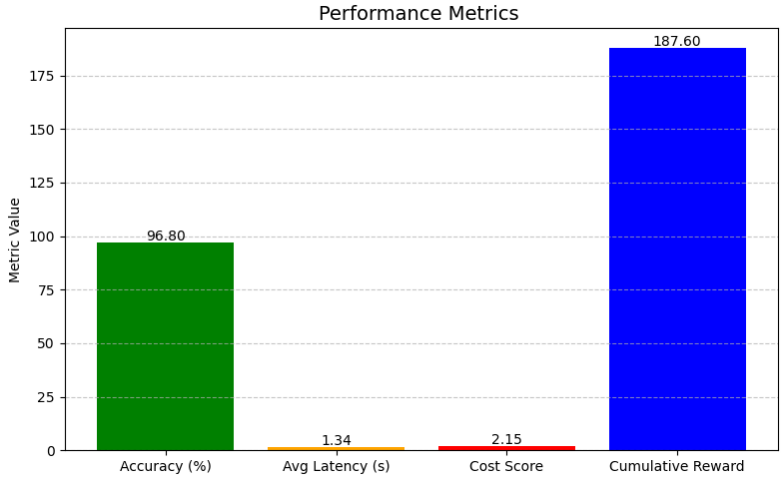

Fig 6. Performance metrics

Fig 6 shows the performance parameters of the PPO-based workflow system demonstrated by Table 1 are Accuracy, Average Latency, Operational Cost Score, and Cumulative Reward. The agent is sticky in accuracy and reward, meaning ability to make excellent decisions and sticky in latency and cost, meaning efficiency and effectiveness of handling OCR-based document workflows.

4.3 Discussion

The suggested DRL-based system solves the dynamic complexity of document processing systems by acquiring knowledge about features of documents and past decisions. This model contrasts with rule-based systems that are inflexible and do not respond to any uncertainty by adapting to the diverse document layouts, densities and priorities via structure state representation. The action distribution analysis indicates that the model mostly chooses the best actions such as auto-assignment and strategic escalation and thereby decreases bottlenecks. When the complexities of the layouts are visualized and the densities of the text are overseen, it becomes evident that the model is highly sensitive to the structural dimensions of the document which is highly imperative in real-time decision-making process. The performance measures, such as policy accuracy, processing latency, cost of operations, and cumulative reward, are used to ensure that performance (learning efficiency of the framework) is much better than the conventional strategies. Based strictly on Python, powered by neural policy networks and PPO, the system demonstrates real-time adaptation. These results confirmed the appropriateness of DRL in intelligent automation pipelines to allow enterprises to handle large volume material with better reliability, throughput and cost effectively.

5 Conclusion and Future Works

The proposed solution to this research is a flexible and responsive document workflow optimization framework based on MDP and Deep Reinforcement Learning. The model is capable of document routing

in an improved performance since the unstructured receipts are converted into proper representations as states of the various policies before PPO is used to the learning of policies. The findings indicate significant decision accuracies, workflow and cost trade-offs resulting to the feasibility of the framework to real life in document intensive markets. An added value to its utility is given by the ability to integrate multi-objective optimization. The solution entirely performed in Python is built on modular and scalable design, which can be deployed to any enterprise. To extend this toward future work, the framework may be extended to support a more general usage with multimodal data (e.g. handwriting, audio notes) as well as external task queues. Moreover, the options of lightweight model compression and federated DRL may be explored to make edge deployment possible within resource limiting environments. Additional enhancements such as the incorporation of interpretability aspects, as a way of explaining policy actions, will also contribute to increasing trust and adoption to the regulated sector such as finance or healthcare.

References

[1] S. V. Mahadevkar, S. Patil, K. Kotecha, L. W. Soong, and T. Choudhury, "Exploring AI-driven approaches for unstructured document analysis and future horizons," *J. Big Data*, vol. 11, no. 1, p. 92, 2024.

[2] A. Premkumar, "AI-based Information Retrieval from Structured Text Documents," PhD Thesis, Technische Universität Kaiserslautern.

[3] B. N. Jørgensen and Z. G. Ma, "Impact of EU Regulations on AI Adoption in Smart City Solutions: A Review of Regulatory Barriers, Technological Challenges, and Societal Benefits," *Information*, vol. 16, no. 7, p. 568, 2025.

[4] G. Arcieri, C. Hoelzl, O. Schwery, D. Straub, K. G. Papakonstantinou, and E. Chatzi, "POMDP inference and robust solution via deep reinforcement learning: An application to railway optimal maintenance," *Mach. Learn.*, vol. 113, no. 10, pp. 7967–7995, 2024.

[5] T. Myllynen, E. Kamau, S. D. Mustapha, G. O. Babatunde, and A. Collins, "Review of advances in AI-powered monitoring and diagnostics for CI/CD pipelines," *Int. J. Multidiscip. Res. Growth Eval.*, vol. 5, no. 1, pp. 1119–1130, 2024.

[6] S. Zhang, Z. Zhao, C. Liu, and S. Qin, "Data-intensive workflow scheduling strategy based on deep reinforcement learning in multi-clouds," *J. Cloud Comput.*, vol. 12, no. 1, p. 125, 2023.

[7] W. Zhang, A. Valencia, and N.-B. Chang, "Synergistic integration between machine learning and agent-based modeling: A multidisciplinary review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 5, pp. 2170–2190, 2021.

[8] S. Chandrasiri and D. Meedeniya, "Energy-efficient dynamic workflow scheduling in cloud environments using deep learning," *Sensors*, vol. 25, no. 5, p. 1428, 2025.

[9] K. Alexopoulos, P. Mavrothalassitis, E. Bakopoulos, N. Nikolakis, and D. Mourtzis, "Deep Reinforcement Learning for Selection of Dispatch Rules for Scheduling of Production Systems," *Appl. Sci.*, vol. 15, no. 1, p. 232, 2024.

[10] A. Amari *et al.*, "An Efficient Deep Learning-Based Approach to Automating Invoice Document Validation," in *2024 IEEE/ACS 21st International Conference on Computer Systems and Applications (AICCSA)*, IEEE, 2024, pp. 1–8.

[11] F. Krieger, P. Drews, and B. Funk, "Automated invoice processing: Machine learning-based information extraction for long tail suppliers," *Intell. Syst. Appl.*, vol. 20, p. 200285, 2023.

[12] Y. Wang, M. Othman, W. O. Choo, R. Liu, and X. Wang, "DFRDRL: a dynamic fuzzy routing algorithm based on deep reinforcement learning with guaranteed latency and bandwidth for software-defined networks," *J. Big Data*, vol. 11, no. 1, p. 150, 2024.

[13] D. Yan, Q. Guan, B. Ou, B. Yan, Z. Zhu, and H. Cao, "A Deep Reinforcement Learning-Based Decision-Making Approach for Routing Problems," *Appl. Sci.*, vol. 15, no. 9, p. 4951, 2025.

[14] Y. Zhang *et al.*, "Multi-path routing algorithm based on deep reinforcement learning for SDN," *Appl. Sci.*, vol. 13, no. 22, p. 12520, 2023.

[15] B. Suh, I. Akobir, J. Kim, Y. Park, and K.-I. Kim, "A Resilient Routing Protocol to Reduce Update Cost by Unsupervised Learning and Deep Reinforcement Learning in Mobile Ad Hoc Networks," *Electronics*, vol. 14, no. 1, p. 166, 2025.

[16] F. Grumbach, A. Müller, P. Reusch, and S. Trojahn, "Robust-stable scheduling in dynamic flow shops based on deep reinforcement learning," *J. Intell. Manuf.*, vol. 35, no. 2, pp. 667–686, 2024.

[17] S. Zhang, Z. Zhao, C. Liu, and S. Qin, "Data-intensive workflow scheduling strategy based on deep reinforcement learning in multi-clouds," *J. Cloud Comput.*, vol. 12, no. 1, p. 125, 2023.

[18] T. Zheng, J. Wan, J. Zhang, and C. Jiang, "Deep reinforcement learning-based workload scheduling for edge computing," *J. Cloud Comput.*, vol. 11, no. 1, p. 3, 2022.

[19] S. Yang, Z. Xu, and J. Wang, "Intelligent decision-making of scheduling for dynamic permutation flowshop via deep reinforcement learning," *Sensors*, vol. 21, no. 3, p. 1019, 2021.
[20] S. Sheng, P. Chen, Z. Chen, L. Wu, and Y. Yao, "Deep reinforcement learning-based task scheduling in iot edge computing," *Sensors*, vol. 21, no. 5, p. 1666, 2021.
[21] B. Peng, T. Li, and Y. Chen, "DRL-based dependent task offloading strategies with multi-server collaboration in multi-access edge computing," *Appl. Sci.*, vol. 13, no. 1, p. 191, 2022.
[22] Y. Zhang, H. Zhu, D. Tang, T. Zhou, and Y. Gui, "Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems," *Robot. Comput.-Integr. Manuf.*, vol. 78, p. 102412, 2022.
[23] J. Dornheim, L. Morand, S. Zeitvogel, T. Iraki, N. Link, and D. Helm, "Deep reinforcement learning methods for structure-guided processing path optimization," *J. Intell. Manuf.*, vol. 33, no. 1, pp. 333–352, 2022.
[24] X. Wang, Z. Yang, G. Chen, and Y. Liu, "A reinforcement learning method of solving Markov decision processes: an adaptive exploration model based on temporal difference error," *Electronics*, vol. 12, no. 19, p. 4176, 2023.
[25] J. Shao and Y. Li, "Optimizing the Long-Term Efficiency of Users and Operators in Mobile Edge Computing Using Reinforcement Learning," *Electronics*, vol. 14, no. 8, p. 1689, 2025.
[26] T. Zhou, L. Luo, S. Ji, and Y. He, "A reinforcement learning approach to robust scheduling of permutation flow shop," *Biomimetics*, vol. 8, no. 6, p. 478, 2023.