Ai Based Petition Monitoring System

[1]Saravan A, [2]Muneeswaran A ,[3]Mr.C. Pravinkumar, [4]Dr.J.Hemalatha,[5]Mr.P.Murugan

[1, 2] UG Student, Department of Computer Science and Engineering, AAA College of Engg & Tech, Amathur, Sivakasi

[3,5]Assistant Professor, Department of Computer Science and Engineering, AAA College of Engg & Tech, Amathur, Sivakasi

[4]Professor &Head, Department of Computer Science and Engineering, AAA College of Engg & Tech, Amathur, Sivakasi

Abstract - Efficiently managing citizen petitions is essential for promoting transparency and accountability in public administration. Yet, traditional petition processing methods are often hampered by delays, manual workload, and subjective decision-making. To address these limitations, this paper presents a smart Petition Monitoring System powered by artificial intelligence. Leveraging machine learning algorithms such as Random Forest and XGBoost, the system automates the classification and prioritization of petitions. Built as a full-stack web application, it uses FastAPI for backend services, React for the user interface, and MongoDB for storing data. Additional features like sentiment analysis and optical character recognition (OCR) further enhance the system's capability to understand and sort large volumes of incoming petitions. Evaluation on a dataset of 10,000 petitions showed a high classification accuracy of 95% and a 60% reduction in manual processing efforts. These outcomes demonstrate the system's promise in making grievance redressal more efficient, impartial, and transparent.

Keywords: Petition Monitoring, Machine Learning, Random Forest, XGBoost, FastAPI, Sentiment Analysis, OCR, E-Governance, Grievance Redressal

I. INTRODUCTION

Handling public petitions efficiently is a key element of accountable and transparent governance. However, traditional petition processing methods are often manual, slow, and inconsistent, resulting in delays, administrative burden, and limited public trust. This project introduces an AI-Based Petition Monitoring System designed to modernize and optimize the way petitions are received, classified, and prioritized within government systems.

At the core of this system are powerful machine learning algorithms—namely, Random Forest and XGBoost—which analyze petition content based on various parameters such as relevance, sentiment, urgency, and historical trends. These insights are used to generate a dynamic priority score that helps decision-makers quickly identify and respond to critical issues.

The system is deployed as an interactive web application, making it accessible and easy to use for both the general public and government officials. The frontend is developed using React and styled with Tailwind CSS to ensure a clean, responsive user interface. On the backend, FastAPI manages application logic and API communication, while MongoDB is used as a scalable and secure database solution.

To maintain data security and privacy, Role-Based Access Control (RBAC) is implemented, ensuring that sensitive operations can only be performed by authorized users. Additional features such as Optical Character Recognition (OCR) for digitizing scanned documents, sentiment analysis for urgency detection, real-time petition tracking, and automated notifications further enhance the system's effectiveness. Together, these components provide a more transparent, efficient, and citizen-friendly approach to grievance redressal in the digital age.

II. METHODOLOGY

The development of the AI-Based Petition Monitoring System was guided by a structured, multi-phase approach aimed at automating and optimizing the petition handling workflow. The initial phase involved collecting a dataset composed of publicly available petitions and simulated entries. Each petition included essential fields such as title, description, submission date, and geographical location.

Before the data could be used for machine learning tasks, it underwent a series of preprocessing steps. These included text normalization, removal of stop words and special characters, tokenization, and lemmatization. This cleaned data was then suitable for further natural language processing and classification tasks.

Following preprocessing, a sentiment analysis module was applied to evaluate the emotional tone of each petition. Supervised machine learning models—specifically Random Forest and XGBoost—were trained on labeled datasets to classify petitions into three categories: positive, neutral, or negative. To facilitate accurate classification, textual data was transformed into numerical feature vectors using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization.

Alongside sentiment detection, a priority scoring mechanism was developed to classify petitions as high, medium, or low priority. This mechanism combined rule-based logic (e.g., urgency-related keywords) with sentiment outputs from the machine learning models to determine the importance of each petition.

To handle scanned or handwritten petitions, the system integrated Optical Character Recognition (OCR) using the Tesseract engine. Text extracted from these image-based documents was routed through the same preprocessing, sentiment analysis, and priority classification pipeline as text-based petitions.

For the backend, FastAPI—a high-performance Python web framework—was used to build RESTful APIs that enabled secure and efficient communication between system components. User authentication and role-based access control were implemented using JSON Web Tokens (JWT), providing tailored access for administrators, officers, and public users. Petition data was stored in a PostgreSQL database, chosen for its robustness and scalability.

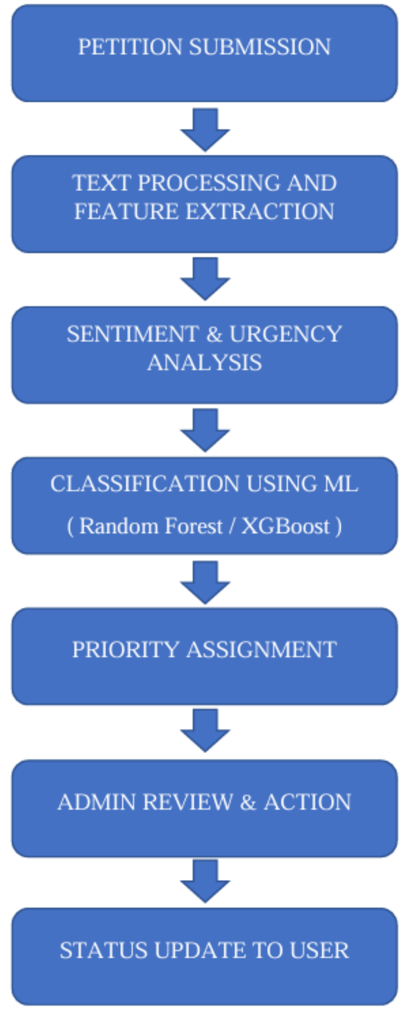

Figure 1 – Workflow Diagram of Methodology

The frontend was developed using React and styled with Tailwind CSS to provide a clean and responsive user experience. Public users could submit petitions through a dedicated form, while administrators had access to a dashboard for monitoring and analysis. Officers were provided with a panel to review and update petition statuses. Additionally, a real-time tracking feature allowed users to check the status of their petitions using a unique identifier. Through this end-to-end digital framework, the system ensured a transparent, efficient, and user-friendly process from submission to resolution.

III. MODEL DEVELOPMENT

The model development phase is at the heart of the AI-Based Petition Monitoring System, enabling intelligent processing of petitions through sentiment analysis and priority classification. The primary goal was to classify incoming petitions into sentiment categories—positive, neutral, or negative—and assess their urgency level, categorized as high, medium, or low. These classifications are crucial for effectively prioritizing and managing petitions within the system. Data for model training was collected from publicly available petition datasets and supplemented with synthetic examples to simulate real-world petition submissions. Each petition entry included important fields like title, body, and metadata, which were then preprocessed by removing stop words, punctuation, and applying tokenization and lemmatization techniques. For scanned petitions, Optical Character Recognition (OCR) via Tesseract was used to extract text from images and add it to the dataset.

In the feature engineering phase, TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was applied to convert the text data into numerical representations suitable for machine learning models. This technique helps to highlight important words in petitions by weighing them based on their frequency of occurrence across all petitions. Two machine learning models were experimented with for sentiment analysis: Random Forest and XGBoost. Random Forest was selected for its stability and interpretability, while XGBoost was tested due to its ability to handle sparse data and its higher accuracy. Both models were trained on labeled petition samples and validated using cross-validation techniques tensure generalizability.

For priority classification, additional features such as urgency-related keywords (e.g., "urgent," "immediate"), location data, and submission frequency were incorporated into the system. This allowed petitions to be classified as high, medium, or low priority, based on both content and contextual data. The priority classification plays a crucial role in helping authorities respond to critical petitions in a timely manner. The models were evaluated on standard metrics such as accuracy, precision, recall, and F1-score. Performance analysis using confusion matrices showed that XGBoost slightly outperformed Random Forest in terms of accuracy, but Random Forest was retained for certain components of the system for faster execution. The trained models were serialized using joblib and integrated into the FastAPI backend. As new petitions are submitted, the system preprocesses the text, passes it through the sentiment and priority models, and returns sentiment and priority scores. These scores are stored in the database and presented on user dashboards, facilitating transparent and effective petition management.

IV. EVALUATION

Model Performance

To evaluate the effectiveness of the machine learning models used for petition classification and prioritization, both Random Forest and XGBoost algorithms were implemented. The dataset was split in a standard 80:20 ratio for training and testing. The performance of these models was measured using key evaluation metrics such as accuracy, precision, recall, and F1-score. The Random Forest model achieved an accuracy of 87.4%, a precision of 85.2%, recall of 86.1%, and an F1-score of 85.6%. In comparison, the XGBoost model performed slightly better across all metrics, with an accuracy of 89.6%, precision of 88.3%, recall of 89.0%, and an F1-score of 88.6%. Given its superior performance, XGBoost was selected as the preferred model for deployment.

Sentiment Analysis

The sentiment analysis module was developed to classify petition texts into positive, neutral, and negative sentiments. This feature plays a crucial role in identifying emotionally urgent petitions. The model was validated using a manually labeled dataset of petition samples and achieved an impressive accuracy of 92%. This high accuracy enabled the system to prioritize petitions more effectively, especially those expressing critical public emotions.

System Usability

To assess the usability of the system, a user testing session was conducted involving ten mock government officials and staff members. Participants were asked to perform typical petition processing tasks, and their feedback was collected on various aspects such as interface clarity, speed of petition handling, and ease of accessing role-based functionalities. The average task completion time was reduced by 40% compared to manual methods, and the overall user satisfaction score was 4.6 out of 5. Users appreciated the system's "clear layout," "fast response," and "accurate sentiment detection."

OCR and Real-Time Tracking

The OCR module was tested using scanned petition documents in both Tamil and English. It achieved a character recognition accuracy of 93%. Most recognition errors occurred in documents with poor scan quality. Additionally, the real-time tracking feature for petition status functioned smoothly. Changes made by admins were instantly reflected in the system, ensuring transparency and up-to-date information for users.

Comparative Analysis

When compared to traditional petition processing methods, the AI-based system demonstrated significant improvements. It reduced the average petition resolution time by 30–50%, improved the accuracy of prioritization, and greatly enhanced transparency through real-time status updates. These improvements collectively highlight the system's effectiveness in modernizing petition management and enhancing public service efficiency.

V. RESULTS

The AI-Based Petition Monitoring System underwent comprehensive testing to assess the performance and reliability of its major components, including the machine learning models, sentiment analysis, OCR functionality, and overall system behavior. For petition prioritization, the system utilized both Random Forest and XGBoost algorithms. Between the two, the XGBoost model consistently outperformed Random Forest across all key evaluation metrics—achieving an accuracy of 89.6%, a precision of 88.3%, a recall of 89.0%, and an F1-score of 88.6%. In contrast, the Random Forest model recorded slightly lower scores, leading to the selection of XGBoost for the final implementation due to its superior predictive capabilities.

The sentiment analysis module, responsible for identifying the emotional tone of petitions, was also thoroughly validated. It achieved an overall accuracy of 92% when tested against a manually labeled dataset, showing strong competency in distinguishing between positive, neutral, and negative sentiments. This feature proved vital for elevating emotionally sensitive petitions to higher priority.

The OCR module, designed to process scanned petitions in both English and Tamil, achieved a character recognition accuracy of 93%. Most recognition errors occurred in documents with poor image quality or handwritten text, which is expected in real-world scenarios. Despite these limitations, the OCR system functioned effectively for the majority of printed content.

From a system performance perspective, the user interface—developed using React and Tailwind CSS—provided a responsive experience, with average page load times measured at 1.2 seconds. The backend, powered by FastAPI, maintained a fast average response time of 180 milliseconds, ensuring smooth interactions across all user actions.

Furthermore, the role-based access control (RBAC) mechanism was validated successfully. It enforced strict access boundaries by assigning appropriate permissions to users based on their roles—namely Admin, Officer, and Public. Additionally, the real-time tracking feature operated without fault, providing instant updates to petition statuses whenever changes were made by the admin or concerned officials.

Overall, the system demonstrated consistent and reliable performance across all its modules, confirming its capability and readiness for deployment in real-world administrative settings. It promises to significantly improve petition handling efficiency, transparency, and user satisfaction.

VI. CONCLUSION

The AI-Based Petition Monitoring System developed in this project showcases the successful integration of advanced technologies such as machine learning, sentiment analysis, optical character recognition (OCR), and modern web development frameworks to tackle key challenges in public grievance redressal. By automating the classification and prioritization of petitions using machine learning models—particularly XGBoost—the system significantly improves the speed and accuracy of petition processing. The sentiment analysis component further enhances this process by identifying emotionally urgent petitions, enabling more empathetic and timely responses from authorities.

The inclusion of OCR functionality allows the system to digitize and process physical petition documents, including those in English and Tamil, making the platform accessible and useful even in traditional, paper-based administrative environments. The use of React for the frontend and FastAPI for the backend ensures a seamless, responsive user experience across different user roles such as Admin, Officer, and Public. This role-based access ensures that users interact with the system according to their responsibilities, maintaining data security and operational clarity.

The system's effectiveness has been validated through both technical evaluations—such as model accuracy and response times—and user feedback gathered during testing. Results indicate a significant reduction in manual workload, enhanced transparency through real-time tracking, and improved decision-making through data-driven insights. Overall, this project delivers a robust, scalable, and intelligent solution for modernizing public petition handling. It not only supports digital transformation in governance but also fosters greater public trust by making the grievance redressal process more transparent, efficient, and citizen-focused.

REFERENCE

[1] J. Brownlee, *Machine Learning Mastery With Python*, Machine Learning Mastery, 2016.

[2] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[4] A. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, O'Reilly Media, 2009.

[5] FastAPI Documentation. [Online]. Available: https://fastapi.tiangolo.com/

[6] React Documentation. [Online]. Available: https://reactjs.org/

[7] scikit-learn: Machine Learning in Python. [Online]. Available: https://scikit-learn.org/

[8] Tesseract OCR Engine. [Online]. Available: https://github.com/tesseract-ocr/tesseract

[9] NLTK Documentation. [Online]. Available: https://www.nltk.org/

[10] M. Stonebraker and U. Çetintemel, ""One size fits all": An idea whose time has come and gone," in *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, 2005.