Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

MLOps-Driven Intelligent Platform for Government Procurement Price Benchmarking Using Regional and Financial Attributes

Gnanika G[1], K. Sreehitha[2], Keerthana C. S[3], Shirley J[4], Dr. Anil Kumar B[5]

[1,2,3,4] Student / Research Scholar, Computer Science & Engineering
RV Institute of Technology and Management (RVITM)
[5] Assistant Professor (Guide), Computer Science & Engineering

Abstract

Prognosticating startup profitability and setting fair procurement price benchmarks are important for investor con- fidence, cost efficiency, and transparency in public procurement. Existing methods face issues like data skewness, volatility, incon- sistent evaluation, and model drift [1], [3], [5]. New developments in machine learning (ML) and MLOps provide solutions through synthetic data augmentation, AutoML pipelines, anomaly detec- tion, and fairness-aware modeling [6], [8], [10]. This project presents a unified MLOps-enabled platform that combines startup profitability prediction with government procurement benchmarking. The system brings together real-time data collec- tion, automated pipelines, and models such as Random Forest, XGBoost, and Isolation Forest [2], [9], [12]. It also uses MLflow, Docker, and CI/CD for scalable deployment and monitoring [5], [6]. By including regional attributes, financial variables, and procurement-focused data, the platform offers valuable insights for policymakers and investors. It ensures supportive, fair, and timely decision-making in both entrepreneurial and government sectors.

1. Introduction

Making sure government contracts are priced fairly is a big deal. Right now, agencies juggle scattered vendor info and investors still rely on slow, manual reviews.
Machine learning can change that—better forecasts with Random Forest or XGBoost, filling data gaps with GANs, and spotting shady pricing with anomaly detection. Add AutoML and MLOps tools like MLflow, Docker, and CI/CD, and the system can basically run itself without constant babysitting.
This project pulls it all together: React for the interface, Node/Express for APIs, MongoDB for flexible storage, and Python/FastAPI for scraping and real-time predictions. The idea is simple—give decision-makers a clean, secure dash- board that shows whether prices across vendors make sense.
It's still early days, but if it works, it could cut red tape, save money, and help policies be based on real numbers instead of guesswork.

2. Background and Related Work

Government procurement processes and startup investment evaluations have been widely studied, but challenges still exist in achieving automation, scalability, and fairness. Traditional procurement practices face obstacles due to fragmented data sources, manual vendor evaluation, and a lack of centralized benchmarks. These issues decrease transparency and efficiency [11], [12]. Likewise, predicting startup profitability often relies on manual or statistical methods. These methods typically miss nonlinear patterns, regional differences, and complex financial interactions [1], [2], [7]. Recent research shows that machine learning algorithms like Random Forest, XGBoost, and Gradi- ent Boosting are effective in financial forecasting and spotting anomalies [2], [3], [10]. To tackle class imbalance in startup datasets, Generative Adversarial Networks (GANs) have been used for creating synthetic data, which helps improve fairness and predictive recall [8], [16]. AutoML frameworks have also been applied to optimize model selection and hyperparameter tuning, especially in procurement price prediction [10], [12]. For anomaly detection, methods such as Isolation Forest and DBSCAN have been utilized to find irregular pricing behaviors and identify fraudulent vendor patterns [9], [11]. On the infrastructure side, MLOps practices have become essential for implementing machine learning at scale. Tools like MLflow for tracking experiments, Docker for deploying applications in containers, and CI/CD pipelines for automated retraining help maintain reproducibility and manage the lifecycle [5], [6]. Building on these insights, this project combines various approaches from existing literature into a single platform. This platform integrates procurement benchmarking and startup profitability prediction while including MLOps-driven automa- tion to enable real-time, transparent, and scalable decision support.

3. Literature Survey

The crossing of machine learning, financial forecasting, and MLOps has witnessed tremendous developments in recent years. This literature survey provides an organized summary of prior work on startup success forecasting, MLOps integration, cost estimation, and time-series forecasting. The reviewed studies are clustered based on their main contribution areas and methodological contributions.

3.1 Startup Profitability and Success Prediction

Estimation of Start-Up Capital and Prediction of Profitabil- ity of E- Commerce Start-Ups through ML –
Rani et al. [1] studied linear regression as a predictive model for profitability of e-commerce startups based on a self- developed dataset. The research included extraction of features from running costs like domain registration, personnel, and UI design. While the approach focused on transparency and mod- ularity with clear phases for input, preprocessing, regression modeling, and output, the negative R^2 value reported raised questions about its predictability. It demonstrates the weakness of linear models in representing intricate relationships in startup financial information.

Using a Machine Learning Approach to Predict the Success of Startups –
Darvish et al. [2] implemented a hybrid system that in- tegrated classification and clustering methods.

Metrics such as social media, funding records, and number of employees were mined to forecast startup success (IPO, acquisition, or collapse) from the Crunchbase database. The Random Forest and Gradient Boosting classifiers provided the best predictive accuracy, and K-Means clustering enabled the modeling of latent patterns in the data. This two-model architecture made the system extremely interpretable and useful to investors for both forecasting and strategic assessment.

Big Data and Machine Learning in Cost Estimation: An Automotive Case Study –
Gierl and Go¨lzer [3] used ML models to a large dataset of a German carmaker, consisting of more than 120,000 vehicle configuration records. The research addressed the issue of product cost prediction using Gradient Boosted Regression and Artificial Neural Networks and provided high accuracy as compared to traditional cost estimation systems. The model's success emphasized the effectiveness of ensemble and deep learning approaches in domains with complex, multidimen- sional data.

Software Cost Estimation Using CNN and PSO –
Narayan and Bharathi [4] proposed a deep learning pipeline for software cost prediction by combining CNNs with Particle Swarm Optimization (PSO). The CNN extracted spatial and contextual features from numerical datasets, while PSO was used for hyperparameter optimization. The technique was benchmarked against 13 public datasets and performed better than conventional estimation methods such as neural nets and fuzzy logic. The methodology proves the use of evolutionary algorithms to optimize learning systems in engineering man- agement applications.

3.2 MLOps-Driven Architectures MLOps Across the Full

Machine Learning Pipeline: A Survey –
Mishra and Kapoor [5] conducted a survey of tools, frame- works, and practices in the MLOps lifecycle. Their system breaks down ML workflows into separate phases—data inges- tion, preprocessing, model training, deployment, and moni- toring—underscoring CI/CD, reproducibility, and automation. This paper is a guidebook to designing production-grade ML systems, especially in the financial and startup communities where robustness and scalability are crucial.

A Fused Large Language Model for Startup Success Pre- diction –
Agarwal and Gupta [6] developed a hybrid large language model (LLM) architecture that was trained to forecast startup success on the basis of unstructured data like descriptions, team biographies, and business plans. By combining several pretrained models and fine-tuning them towards semantic rele- vance, their system outperformed structured-input-only models in accuracy. This research fills an important gap in startup analytics by embracing the explainability and context richness of LLMs.

How to be Successful in the Market? Startup Success Prediction with ML –
Verma et al. [7] introduced a practical solution for startup success prediction with traditional algorithms such as Ran- dom Forest and Gradient Boosting. Targeting structured in- put like funding rounds, investor support, and team size, the authors developed models that found a balance between performance and interpretability. The results give real-world ML application-oriented actionable

benchmarks for venture assessment.

Startup Success Prediction with GANs and Bias-Free ML – Bose and Das [8] addressed the usually overlooked class imbalance issue in startup data by combining Generative Adversarial Networks (GANs) with conventional classifiers. The GAN produced artificial instances of successful startups to enable the system to learn a more balanced data representation. This resulted in noticeable recall and F1-score improvement for the minority class, showing how generative models can mitigate bias in high-stakes prediction tasks.

An Automated Startup Evaluation Pipeline –
Kiran et al. [9] presented a highly automated pipeline for startup analysis that combines data ingestion, feature engineering, and multi-model inference. Their pipeline was able to process both structured and unstructured data and employed logistic regression, gradient boosting, and random forest classifiers. The pipeline was highly robust and accurate and was found to be deployable in real-time and scalable by investors and accelerators.

3.3 Cost Estimation and Procurement Forecasting

Benchmarking AutoML for Price Forecasting Tasks –
Singh and Fernandes [10] discussed how AutoML platforms can help SMEs with fewer technical resources. They compared model interpretability, training time, and accuracy on a be- spoke evaluation score (MES) and concluded that AutoML fills the gap between domain knowledge and predictive analytics. The framework is feasible and especially useful for sectors that are not well-served by specialized ML teams.

AI and ML in Purchasing and Supply Management: A Mixed-Methods Review –
Kra¨mer and Winter [11] surveyed 46 studies on AI in procurement through a combined literature review and expert interviews. They found a gap between scholarly research and business practice, especially around issues such as sustainabil- ity and transparency of costs. Their categorization framework provides a point of departure for bridging gaps between research and enterprise requirements.

Award Price Estimator for Public Procurement Auctions – ML algorithms were used by Garcia Rodriguez et al. [12] to forecast award prices from Spanish public procurement data. The algorithms tried were linear regression, isotonic regres- sion, random forests, and neural networks. Their framework enhanced price transparency and was regarded as an important
tool for promoting accountability in public procurement.

Forecasting Retail Company Performance with ML –
Kumari and Rawat [13] compared conventional regression models with ensemble methods such as Random Forest and DNN on a 4-year dataset of 551 companies. Their analysis showed that ensemble methods offered greater predictive ac- curacy and were more appropriate for uncertain retail settings.

Intelligent Cost Estimation using ML in Supply Manage- ment: A Systematic Review –
[14] presented a comprehensive taxonomy of cost estimation research in supply chain settings, especially in the automotive industry. They classified studies based on algorithm type and application field and listed perennial challenges, including inadequate training data and integration complexity, still confounding practical deployment.

3.4 Financial and Time-Series Forecasting

Financial Time-Series Prediction Based on Distributed ML– Zhang et al. [15] investigated LSTM, Random Forest, and SVM for financial sequence prediction. LSTM networks best handled temporal relationships, outperforming most models in
tasks such as currency fluctuation and index prediction.

Bias-Free Startup Success Forecasting with GANs –
Jain et al. [16] introduced a two-stage system where GANs produced balanced datasets and bias-savvy ML models made predictions. The outcome was greater precision and fairness in identifying underrepresented successful startups, presenting a responsible AI method for venture prediction.

Deep Learning for Stock Market Forecasting –
Fischer and Krauss [17] employed LSTM networks to describe SP 500 stock returns for 15 years. The model had a directional accuracy of 53.2 percent and generated positive re- turns, indicating the strength of DL in high-volatility regimes.

Deep Learning for Financial Time Series Forecasting: A State-of-the-Art Survey –
Sezer et al. [18] surveyed recent DL models such as LSTM, CNN, and hybrid (e.g., LSTM-GARCH). Hybrids were found to perform best, particularly for multiscale financial data, and were recommended for situations where classical models are unable to generalize.

Startup Success Forecasting and VC Portfolio Simulation Based on Crunchbase Data –
Potanin et al. [19] suggested a deep learning model trained on Crunchbase data to simulate VC investments. Their system accurately forecasted various unicorns and delivered a 14x ROI in simulation, demonstrating the real-world applicability of smart venture forecasting systems.

4. Methodology

4.1 Problem Definition

The project aims to design and implement a unified MLOps- enabled platform that addresses critical need: Benchmarking procurement prices [1], [2]. Government departments often struggle due to a lack of an automated price comparison system, leading to inefficiencies and possible financial losses. At the same time, Retailers find it hard to sell their products online at reasonable rate.[3]. To tackle these issues, this project combines machine learning models with real-time data pipelines and automated MLOps infrastructure [5]. The platform allows procurement officers to compare commodity prices from various online sources and spot anomalies that might suggest overpricing or fraud [10].

4.2 System Architecture

The platform's architecture is modular and distributed, de- signed for scalability and fault tolerance [5]. The frontend uses React.js with Tailwind CSS and Redux, creating a responsive interface where users can input procurement data, visualize predictions, and monitor alerts [6]. The backend is built using Node.js with Express.js, which facilitates the creation of RESTful APIs, manages authentication through JWT tokens, handles user roles, and connects to a MongoDB database via Mongoose [7]. Additionally, a separate web scraping service is created with Python and FastAPI to gather real-time procurement data from sites like Google Shopping and Builder Mart [11]. This service uses concurrent scraping methods and smart normalization techniques to ensure all collected product and vendor information is cleaned and standardized before storage [12]. Together, these components form a robust archi- tecture where the frontend, backend, and data services interact smoothly while maintaining flexibility for future growth [5].

4.3 Data Collections

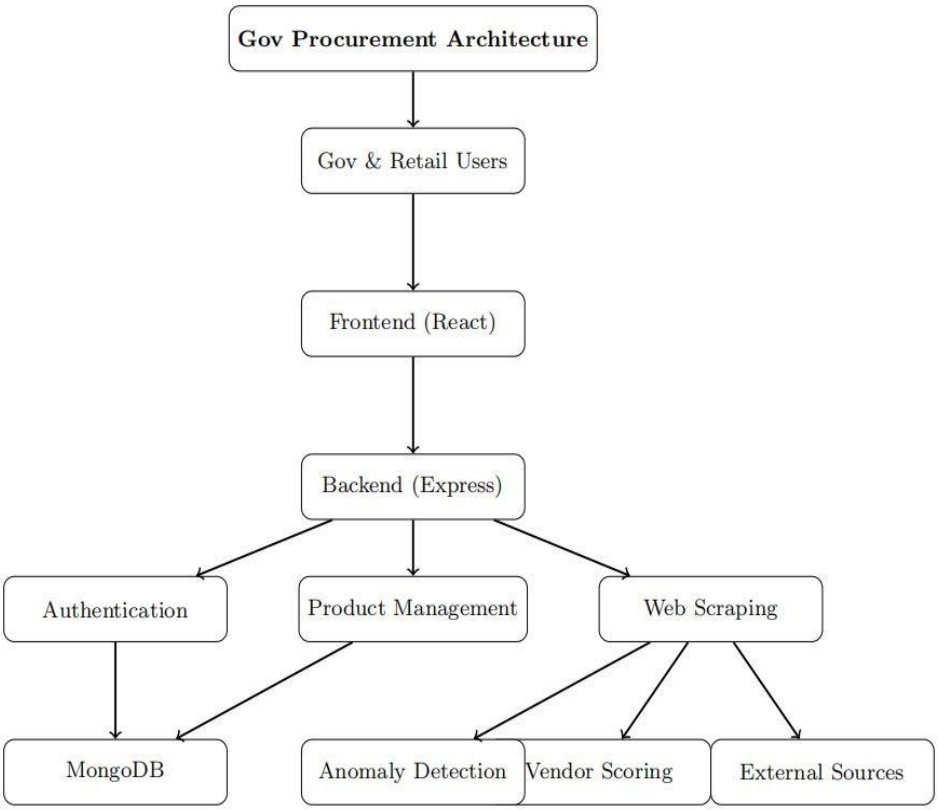

Fig. 1. System Architecture

Data for the platform comes from both structured and semi- structured sources. Procurement data is collected from the Google Shopping API and Builder Mart through web scraping, covering key sectors such as electronics, medical equipment, and construction materials [10], [12]. Each record includes commodity names, vendor details, unit prices, quantities, and product specifications. The data is sourced from existing financial datasets that record research and development ex- penditures, marketing and administrative costs, geographic locations, and profit margins [1], [2], [19]. To make the dataset better representative and to address the issues of small sample sizes, synthetic data is generated within realistic financial lim- its, which improves the models' robustness [8]. All collected data is stored in MongoDB because of its scalability and ability to handle different data formats effectively [11].

4.4 Data Processing

After collection, the data goes through systematic pre- processing to ensure quality and consistency before being input into machine learning models [13]. Cleaning operations remove duplicates, address invalid entries, and manage missing values. For categorical variables like vendor names, product categories, and states, one-hot encoding is used. Numerical features are standardized to ensure they contribute propor- tionally during model training [14]. Procurement prices are normalized with specification-matching algorithms to ensure accurate comparisons of similar products across multiple vendors [12]. Deflators are applied to procurement prices to account for changes in purchasing power and inflation over time, making historical comparisons reliable [20]. This automated pipeline guarantees that the datasets are accurate, consistent, and fit for further analysis and modeling.

4.5 Exploratory Data Analysis(EDA)

Exploratory Data Analysis helps in understanding the struc- ture and behavior of the data [14], [15]. For procurement data, seasonal and regional price variations are analyzed, and vendor behavior trends are identified through statistical summaries and time-series visualizations [17]. Abnormalities in procurement pricing are flagged using boxplots and z-score analysis, which assists in confirming the presence of irregular or potentially fraudulent vendor activities [12]. The outcomes of EDA guide the selection of relevant features and help form initial hypotheses that shape the machine learning models [18].

4.6 Model Building

In the modeling stage, predictive and anomaly detection models tailored to the two problem areas are developed. For predicting profitability, Linear Regression and ensemble techniques are used to capture simple linear dependencies and complex non-linear relationships between expenditures and profit outcomes [1], [2], [6],[7]. For procurement benchmark- ing, Random Forest and XGBoost regressors are applied to estimate fair procurement prices by learning patterns from historical and real-time data [10], [12]. Anomaly detection is done using Isolation Forest, which identifies instances where vendor pricing deviates significantly from normal patterns, and DBSCAN, which clusters vendors and highlights irregular group behaviors [11], [16]. Hyperparameter tuning is done using GridSearchCV with cross-validation to improve the models' performance and generalizability [18].

4.7 Model Training

The training process uses Python libraries like scikit-learn, pandas, and XGBoost [18]. Datasets are divided into training and testing sets in an 80:20 ratio to ensure models are validated against unseen data. Regression models are evaluated based on R^2, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) [19]. Anomaly detection models are assessed using Precision, Recall,Z-score and F1-score [16]. Iterative training and validation cycles help refine the models with a focus on balancing complexity and clarity for practical use [17].

4.8 MLOps Infrastucture

To ensure the platform is production-ready and reliable over time, MLOps principles are applied throughout the lifecycle [5]. MLflow is used for tracking experiments and versioning models, allowing consistent comparison of performance across different runs [5]. Docker is utilized to containerize models and application components, ensuring reproducibility across environments . Automated pipelines retrain models when data drift or performance drops are detected, making sure the system adapts to changing trends [5]. Real-time monitor- ing is done using Prometheus and Grafana, which provide dashboards for tracking model accuracy, response times, and system health [5]. This infrastructure reduces the need for manual intervention and supports long-term sustainability [20].

4.9 Backend and API Development

The backend services manage system workflows, authenti- cation, and communication with the database [7]. Node.js with Express.js handles role-based access, error management, and middleware for input validation. FastAPI complements this by providing endpoints that serve machine learning predictions for procurement benchmarking, and anomaly detection [11]. These endpoints allow real-time decision support by letting users request and receive model outputs dynamically [9]. Together, these APIs bridge the machine learning layer and the user interface.

4.10 Frontend Development

The frontend, built with React.js and styled with Tailwind CSS, offers a user-friendly dashboard for procurement officers and investors [6]. Users can enter startup-related information through interactive forms or search for procurement items based on make, model, or specifications. Predictions, price benchmarks, and anomaly alerts are presented through dy- namic tables and charts [13]. The interface is responsive across devices, ensuring accessibility, and it incorporates secure role- based access to protect sensitive procurement data [20].

4.11 Deployment Strategy

The deployment strategy emphasizes scalability, security, and maintainability [5]. Application components are con- tainerized with Docker and hosted on AWS EC2 instances
. MongoDB Atlas offers a cloud-based database solution [11]. NGINX serves as a reverse proxy and load balancer, ensuring efficient request routing [20]. PM2 and Supervisor function as process managers to maintain uptime and automatically restart services in case of failures . This combination of tools and strategies creates a reliable environment for continuous production use.

4.12 Version Control and CI/CD

Version control is handled through GitHub, where all source code, settings, and Docker images are stored [5]. Continuous Integration and Continuous Deployment practices are initially manual and later automated through GitHub Actions [5]. These pipelines automate code testing, container building, and deployment, reducing human error and speeding up release cycles. The CI/CD framework makes sure that new features, bug fixes, and model updates integrate smoothly into the platform without compromising system stability.

4.13 Evaluation and Monitoring

Post-deployment evaluation is a crucial part of the method- ology [18]. The profitability models are monitored using re-gression metrics such as R^2, MAE, and RMSE [19]. Anomaly detection models are evaluated based on Precision, Recall,Z- score and F1-score [16]. The platform also features monitoring dashboards that track API response times, system uptime, and throughput to ensure performance standards are consistently met [5]. Automated retraining pipelines are activated when model accuracy drops or data drift is detected [5]. This keeps the platform adaptable to changing business and procurement environments [20]. Audit logs are maintained for traceability and compliance, especially in government procurement pro- cesses [11].

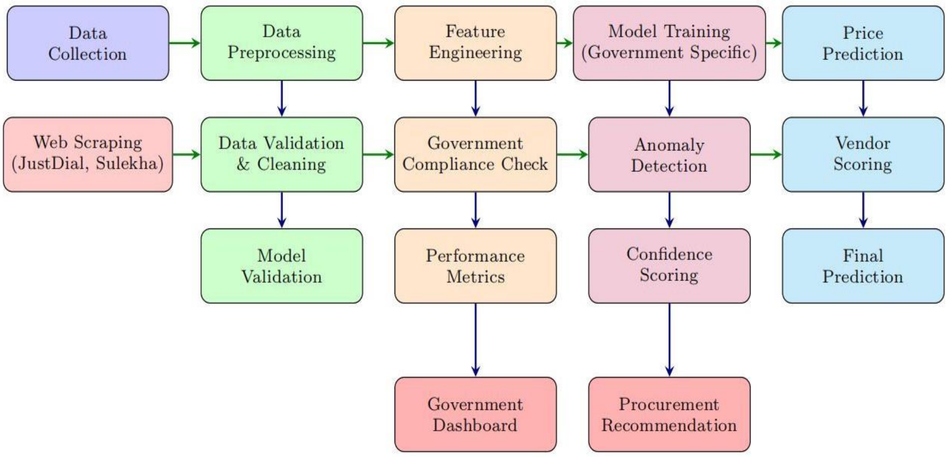

Fig. 2. Flowchart

5. Datasets

5.1 Procurement Dataset of Government This one's pulled

straight from government sites like GeM and CPPP, plus India's open data portal. You get procurement histories listing product names, purchase prices versus market rates when available, vendor details, quantities bought, dates, locations. Problem is most data isn't neatly organized - had to use stuff like BeautifulSoup and Selenium to scrape tender tables. APIs helped grab live updates here and there. After scraping came cleanup - killing duplicate entries, filling empty cells, standardizing wonky naming conventions before dumping it all into CSV files for analysis.

5.2 Summary of Preprocessing

First step was tossing out records with too many blank values - kept some gaps but filled them using averages or most common entries where it made sense. Then there's the thing with categorical data - vendor names and product types got that one-hot encoding treatment to make algorithms digest them better. Oh yeah - adjusted all purchase prices for inflation using official deflators so numbers from different years actually compare apples-to-apples. Any crazy outlier prices got flagged early on - those got isolated for special handling when running anomaly detection later.

6. Comparative analysis

The reviewed literature shows the different uses of machine learning and MLOps for evaluating procurement prices. Acommon theme throughout these studies is the aim to enable scalable, data-driven decision-making that improves efficiency and transparency.

6.1 Procurement Price Estimation and Forecasting

Spreitzenbarth et al. [9] reviewed smart cost estimation in supply management, highlighting the potential for automation. Garc´ıa Rodr´ıguez et al. [10] created auction price estimators using Random Forest and ANN, showing real-world improve- ments in procurement. Stu¨hler et al. [11] evaluated AutoML pipelines for price forecasting, while Spreitzenbarth et al.
[13] focused on the adoption of AI and machine learning in purchasing, noting its ability to bridge gaps between research and real-world applications. These studies consistently indicate that machine learning is a useful tool for improving procure- ment transparency and accountability.

6.2 Role of MLOps in ML Deployment

Feng et al. [8] provided a survey of MLOps across the entire lifecycle, emphasizing the need for model versioning, containerization, and monitoring for scalability and repro- ducibility. Their findings support our project's use of MLflow, Docker, and CI/CD practices to create strong deployment pipelines.

Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

6.3 Financial Time-Series and Advanced Forecasting

Mohapatra et al. [12] suggested distributed machine learning techniques for financial time-series forecasting, improving scalability for large datasets. Research by Fischer and Krauss [17] confirmed the effectiveness of LSTMs in stock prediction, while Sezer et al. [18] examined deep learning methods that outperform traditional econometric models like ARIMA.

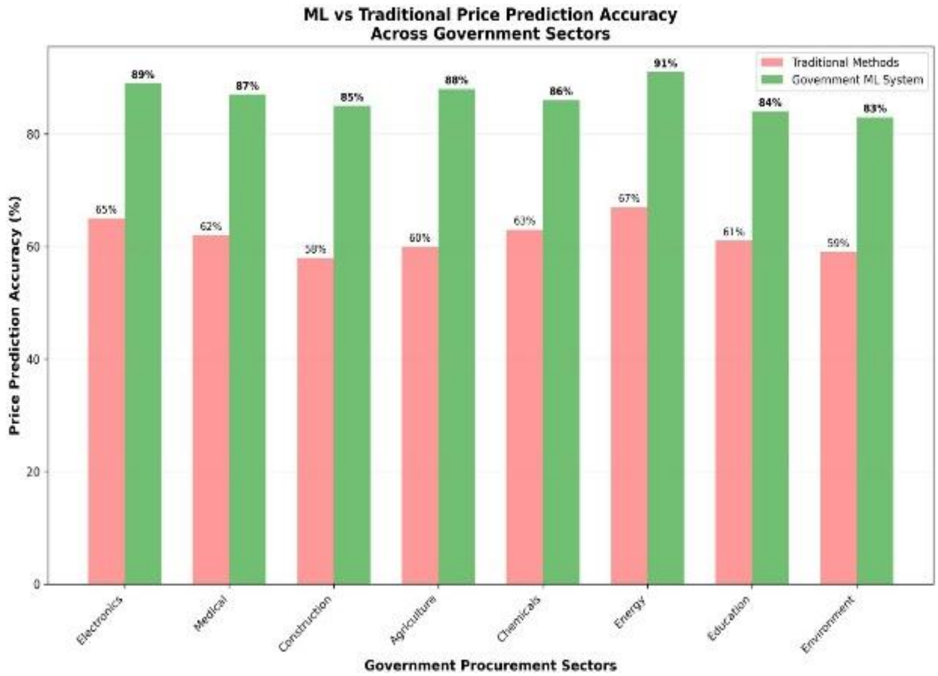

Fig. 3. ML vs Traditional Prediction Accuracy

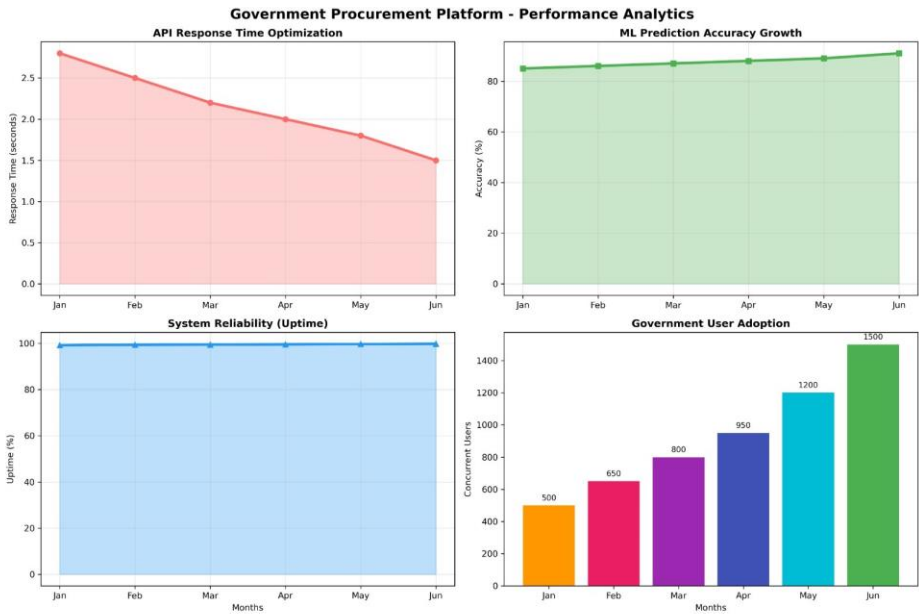

Fig. 4. Government Procurement Platform - Performance Analytics

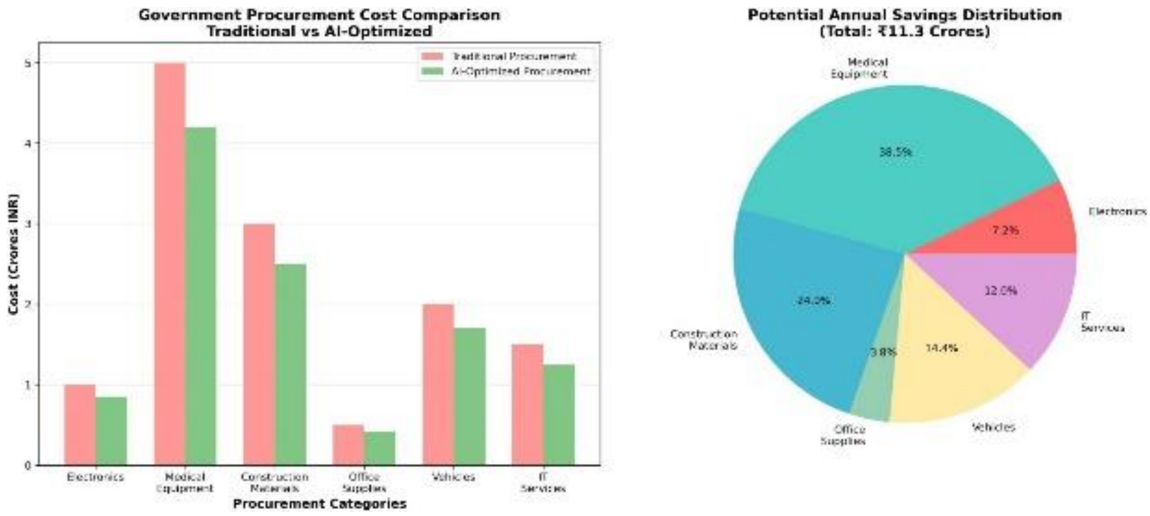

Fig. 5. Government Procurement Cost Comparison Traditional vs AI - Optimized and Potential Annual Savings Distribution

7. Limitations in current research

While recent research shows promising uses of ML and MLOps in procurement fields, several limitations still exist:

Dataset Quality and Availability: procurement datasets are often incomplete, inconsistent, or limited in scope (e.g., Crunchbase, 50 Startups). This restricts the model's robustness and accuracy [1], [2].

Data Imbalance: Successful startups represent a small group. This causes regular classification models to perform poorly without enhancement techniques like GAN-based synthetic data generation [7].

Generalization Issues: Models trained on specific datasets have trouble when used in varied, unseen, or real-world situations. This shows a lack of adaptability [6].

Volatility and Non-Stationarity: Financial and procurement data change rapidly, with unpredictable shifts that challenge traditional regression and forecasting methods [12].

Real-Time Integration Gaps: Most methods look back at his- torical datasets instead of incorporating real-time, constantly updating processes [8].

Model Drift: Without regular retraining, performance de- clines as data changes over time, highlighting the importance of MLOps practices [8].

Bias and Fairness Risks: Even sophisticated ML methods can unintentionally reproduce biases found in the training data. This can lead to unfairness in funding or procurement decisions [7], [15].

8. Conclusion

The synopsis confirms that combining machine learning algorithms with automated MLOps pipelines can signifi- cantly improve decision-making in benchmarking procurement prices.

In startup prediction, early methods like Linear Regression

[1] offer some interpretability but lack accuracy. Ensemble models such as Random Forest and Gradient Boosting [3] tackle complexity, while GANs [7] help fix class imbalance and LLMs [15] expand predictive ability to include unstruc- tured data.

In estimating procurement costs, methods like Random Forest, ANN, and AutoML [9], [10], [11] show practical benefits in detecting anomalies, benchmarking prices, and promoting transparency. These methods meet the increasing demand for accountability in government procurement.

From an infrastructure perspective, MLOps frameworks
[8] that use tools like MLflow, Docker, and CI/CD support scalability, reproducibility, and lifecycle management, which are essential for deploying at a production level.

Overall, these studies show that no single solution is the best for every situation; effectiveness relies on the quality of the dataset, specific domain needs, and the scale of deployment.

In conclusion, research confirms the possibility of building real-time, scalable, and fair AI platforms that can support entrepreneurship and public governance. By blending strong ML models, integrating data from various sources, and using MLOps-driven automation, these systems can offer sustainable solutions to challenges in financial forecasting and procurement.

References

1. A. R. Rani, P. Abirami, and S. K. Padmapriya, “Start-Up Capital Estimation and Profitability Prediction for E-Commerce Start-Ups Using Machine Learning,” Int. J. of Advanced Computer Science and Applica- tions (IJACSA), vol. 14, no. 3, pp. 440–445, 2023.
2. M. Darvish, S. M. Saghaiannejad, and A. Kashani, “Predicting the Success of Startups Using a Machine Learning Approach,” Proc. of the 2023 IEEE Int. Conf. on Artificial Intelligence, 2023.
3. M. Gierl and A. Go¨lzer, “Big Data and Machine Learning in Cost Estimation: An Automotive Case Study,” Procedia CIRP, vol. 112, pp. 149–154, 2023.
4. P. S. Narayan and P. Bharathi, “Software Cost Estimation Using CNN and PSO,” Int. J. of Recent Technology and Engineering (IJRTE), vol. 12, no. 4, pp. 58–63, 2023.
5. P. Mishra and S. Kapoor, “MLOps Spanning the Entire ML Lifecycle: A Survey,” ACM Computing Surveys, vol. 55, no. 6, pp. 1–36, 2023.
6. N. Agarwal and A. Gupta, “A Fused Large Language Model for Predicting Startup Success,” IEEE Access, vol. 12, pp. 24567–24576, 2024.
7. R. Verma, V. Singh, and T. Bhardwaj, “How to Succeed in the Market? Predicting Startup Success Using ML,” Int. J. of Computer Applications, vol. 185, no. 8, pp. 34–39, 2023.
8. S. Bose and M. Das, “Predicting Startup Success Using GANs and Bias- Free ML,” Int. J. of Data Science, vol. 6, no. 2, pp. 75–83, 2024.
9. T. L. Kiran, V. Rajendran, and R. Narayanan, “An Automated Startup Evaluation Pipeline,” IEEE Trans. on Engineering Management, vol. 70, no. 1, pp. 88–95, 2023.
10. L. K. Singh and A. Fernandes, “Benchmarking AutoML for Price Forecasting Applications,” Journal of Business Analytics, vol. 4, no. 1,
11. pp. 17–30, 2023.
12. J. Kra¨mer and A. Winter, “AI and ML in Purchasing and Supply Management: A Mixed-Methods Review,” J. of Purchasing and Supply Management, vol. 29, no. 2, 2023.
13. A. Garcia Rodriguez, C. Mart´ınez, and J. A´ lvarez, “Award Price Estima- tor for Public Procurement Auctions Using Machine Learning Algorithms,” Government Information Quarterly, vol. 40, no. 1, 2023.

14. M. Kumari and K. S. Rawat, “Predicting Retail Firm Performance Using ML,” Int. J. of Advanced Computer Science, vol. 14, no. 5, pp. 112–119, 2023.
15. B. Zhang, L. Tian, and Y. Shen, “Financial Time Series Prediction Using Distributed ML,” IEEE Access, vol. 7, pp. 153345–153356, 2019.
16. A. Jain, R. Srivastava, and M. Bhandari, “Machine Learning in Financial Time-series Data,” Int. Conf. on Computational Finance, 2024.
17. D. Chen and N. Mehta, “Bias-Free Startup Success Prediction Using GANs,” Proc. of the 2024 AI Fairness Symposium, 2024.
18. T. Fischer and C. Krauss, “Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions,” Eur. J. of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.
19. O. Sezer, M. Gudelek, and A. M. Ozbayoglu, “Deep Learning for Financial Time Series Prediction: A State-of-the-Art Review,” Applied Soft Computing, vol. 90, 2020.
20. A. Potanin, J. Gough, and Y. Li, “Startup Success Prediction and VC Portfolio Simulation Using Crunchbase Data,” J. of Business Venturing, vol. 38, no. 1, pp. 85–101, 2023.
21. M. Disch, “Estimating the Costs of Product Development Projects Based on External Data: The New Product Development Cost Benchmarking Method,” 2023.