

CloudPulse AI: AI-Powered Cloud FinOps Advisor

Mr. Shivprasad Satish Mali

Ashokrao Mane Group of Institutions, Kolhapur
Department of computer science & engineering
shivprasad129806@gmail.com

Mr. Pratik Sanjay Gaikwad

Ashokrao Mane Group of Institutions, Kolhapur
Department of computer science & engineering
gaikwadpratik8105@gmail.com

Mr. Roshan Prakash Bhatale

Ashokrao Mane Group of Institutions, Kolhapur
Department of computer science & engineering
roshanbhatale0504@gmail.com

Mr. Vivek Raghunath Bachate

Ashokrao Mane Group of Institutions, Kolhapur
Department of computer science & engineering
vivekbachate6377@gmail.com

Mr. Prathamesh Tukaram Mhetar

Ashokrao Mane Group of Institutions, Kolhapur
Department of computer science & engineering
prathmeshmhetar@gmail.com

Mrs. Almas Amol Mahaldar

Assistant professor, AMGOI Kolhapur
Department of computer science & engineering
aam@amgoi.edu.in

Abstract

Cloud services offer flexible infrastructure but variable billing can make cost monitoring difficult. FinOps practices support financial accountability by making cloud expenditure visible and actionable.

This paper presents CloudPulse AI, a FinOps prototype implemented with Django and Python. The prototype uses simulated hourly billing records, an Isolation Forest model for unusual cost detection, service-specific recommendation rules, dashboard cost summaries, short-term trend projections and email alerts for manually triggered critical anomalies.

The implemented system provides an authenticated dashboard, hourly cost visualisation, anomaly ledger, recommendation ledger, estimated saving values and alert email delivery. Functional demonstration screenshots show the output of these modules. The current prototype has not been validated on production AWS billing exports and does not claim benchmark accuracy.

CloudPulse AI demonstrates how a lightweight web application can combine cost visibility, unsupervised anomaly detection and practical recommendations in one FinOps workflow. Further work is required for production billing ingestion, deterministic recommendation estimates and formal model evaluation.

Keywords: AWS, Cloud FinOps, cost optimisation, anomaly detection, Isolation Forest, Django, cloud billing

Glossary of Abbreviations

AI - Artificial Intelligence

AWS - Amazon Web Services

CSV - Comma-Separated Values

EC2 - Elastic Compute Cloud

EBS - Elastic Block Store

FinOps - Financial Operations

ML - Machine Learning

RDS - Relational Database Service

S3 - Simple Storage Service

1. Introduction

Cloud computing allows organisations to provision infrastructure on demand, but its pay-as-you-go model introduces a continuing need for expenditure visibility. Teams must identify unusual increases, understand which services contribute to cost, estimate near-term expenditure and prioritise optimisation actions. FinOps addresses this operational challenge by connecting engineering decisions with financial accountability.

CloudPulse AI is a final-year academic prototype designed to demonstrate this workflow for AWS-oriented billing data. It presents hourly spend telemetry, detects unusual billing records, produces service-specific optimisation suggestions and sends alert emails for manually injected critical anomalies. The system is intended as a demonstrator rather than a production AWS billing platform.

The main contribution of this work is an integrated Django application that combines simulated cloud billing data, an unsupervised machine-learning detector and readable recommendation rules in a single interface. The design favours clarity and reproducibility: each implemented feature can be traced to source data and a defined application module.

Research Question: Can a lightweight AI-powered FinOps platform improve visibility of cloud expenditure and assist users in identifying anomalous cloud costs and optimization opportunities?

2. Literature Survey

Deochake (2023) presented a comprehensive review of cloud cost optimization strategies, focusing on resource allocation, cloud pricing models, workload management, and cost-control techniques. The study highlighted that organizations can significantly reduce operational expenditure through continuous monitoring, intelligent resource utilization, and proactive FinOps practices.

Khan et al. (2024) proposed a graph-based approach for cloud cost modelling and optimization. Their framework models relationships between cloud resources, utilization, performance, and cost factors to support informed decision-making. The study demonstrated that graph-theoretic analysis can improve scalability, resource placement, and cloud cost optimization across complex cloud environments.

Luong et al. (2017) surveyed economic and pricing models used in cloud networking and resource management. Their work explained how pricing mechanisms, incentives, and market-based allocation strategies influence cloud resource utilization and cost efficiency, providing a foundation for cost-aware cloud management systems.

Srinivasa (2024) investigated cloud cost optimization methodologies for cloud migration projects. The study emphasized expenditure forecasting, resource assessment, and anomaly identification techniques to reduce migration-related costs and improve operational efficiency during cloud adoption processes.

Ahmad et al. (2023) explored deadline-aware cloud cost optimization techniques where workload execution deadlines are considered alongside resource costs. Their research showed that balancing performance requirements with cost objectives can improve resource allocation efficiency while maintaining service quality.

Deochake (2024) further proposed ABACUS, a FinOps-based cloud cost optimization service that combines budget monitoring, deployment-cost estimation, and automated alerting mechanisms. The study demonstrated how FinOps practices can improve cloud-spend visibility and financial accountability within organizations.

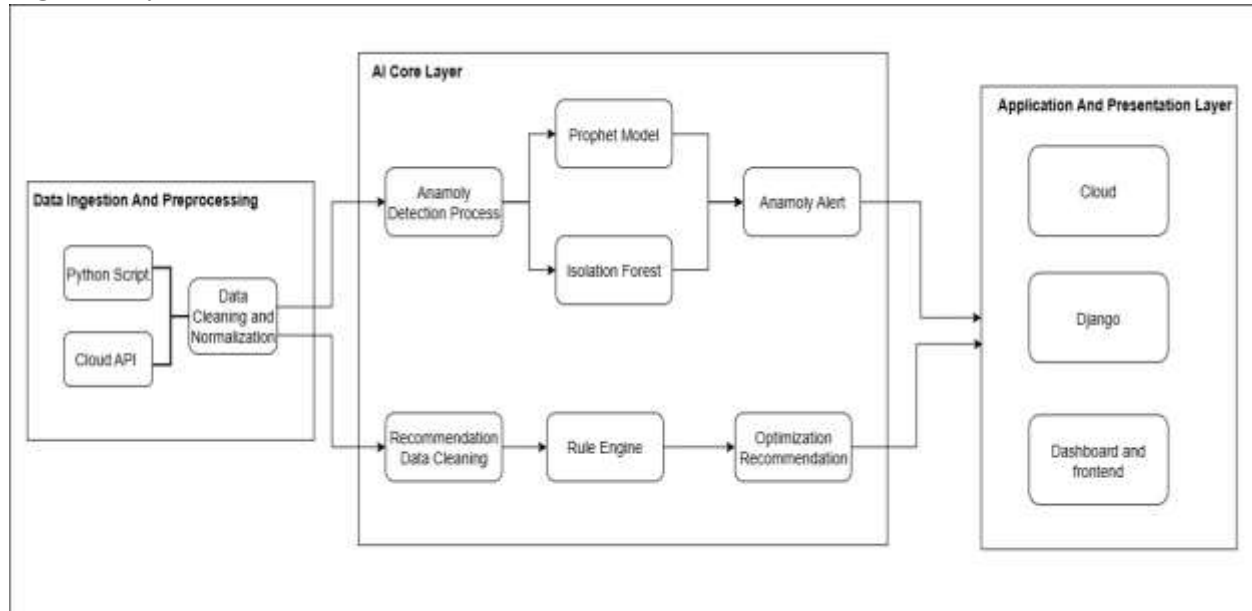
The reviewed literature demonstrates that effective cloud cost management requires a combination of continuous monitoring, intelligent resource allocation, anomaly detection, expenditure forecasting, and FinOps practices. Existing studies provide valuable insights into cloud cost optimization; however, most focus on specific aspects such as pricing models, migration strategies, or resource scheduling. CloudPulse AI builds upon these concepts by integrating cost monitoring, anomaly detection, forecasting, and recommendation generation into a unified platform aimed at improving cloud financial visibility and operational decision-making.

3. Materials and Methods

3.1 System Scope and Architecture

CloudPulse AI is implemented as a Django web application with HTML, CSS and JavaScript templates, a SQLite database for user-related records and CSV files for billing telemetry. Authentication pages support registration, login, logout and password-reset email verification. A user profile stores organisation-related fields, an AWS-oriented cloud-provider value, monthly budget information and optional additional email recipients.

Figure 1. System Architecture



The prototype does not currently connect to live AWS Cost and Usage Reports or other cloud-provider APIs. Instead, a local generator creates reproducible simulated billing records. A live-update endpoint appends additional hourly records to support dashboard demonstrations.

3.2 Simulated Billing Data

The billing data generator creates hourly costs across EC2, RDS, S3 and CloudFront services for the period from 1 January 2024 to 31 March 2024. It applies a gradual cost trend, weekday and weekend differences, office-hour variation and random noise. It also injects selected anomalous records. During live dashboard use, additional hourly records may be appended for EC2, S3, RDS, CloudFront, Lambda and EBS, with service weights and time-of-day adjustments.

3.3 Anomaly Detection

The anomaly detector uses the Isolation Forest algorithm from scikit-learn. For each hourly record, the feature set contains cost, hour of day, weekday and an encoded service identifier. The model is configured with 150 estimators, a contamination value of 0.04 and a fixed random state of 42. Records predicted as outliers are labelled with low, medium or high severity according to the decision score. A manually forced anomaly is labelled critical to support demonstrations and email-alert testing.

3.4 Cost Forecast and Dashboard Summary

The dashboard visualises the latest 72 aggregated hourly cost points. It derives a short twelve-point forecast from the recent average with a smoothed trend variation for display. The dashboard summary shows actual cost, forecasted cost and saving potential. These values are prototype estimates for decision support and should not be interpreted as validated financial forecasts.

The repository also contains an optional daily forecasting helper that uses Prophet when available and otherwise falls back to a rolling mean. The dashboard path demonstrated in this paper uses the lightweight projection described above.

3.5 Recommendation Engine and Email Alerts

The recommendation module groups records by service and calculates an average cost. For services whose average cost is at least USD 5, it returns service-specific advice such as EC2 rightsizing, S3 intelligent tiering, RDS allocation optimisation, CloudFront cache improvement and Lambda execution tuning. Saving figures are indicative estimates based on the service average and a bounded percentage range.

When a user manually triggers a critical anomaly, the application sends an email to registered users and optional profile recipients. The message identifies the service, severity, detection time, observation and recommended review action. A fifteen-minute cooldown per service and severity reduces repeated email delivery.

3.6 Statistical Analysis

Descriptive statistical measures including mean, moving average and trend analysis were used for cost summarization. Isolation Forest anomaly scores were analyzed to categorize observations into severity levels. No hypothesis testing was performed because the study used simulated data.

4. Results

Functional testing confirmed that the implemented prototype presents the principal FinOps workflow in the browser. The dashboard displays recent simulated cost telemetry, an indicative projection and summary cards for actual cost, forecasted cost and saving potential. The anomalies page lists detected unusual records with service descriptions and severity. The recommendations page displays readable service-specific actions with estimated savings. A manually triggered critical anomaly produces an email notification.

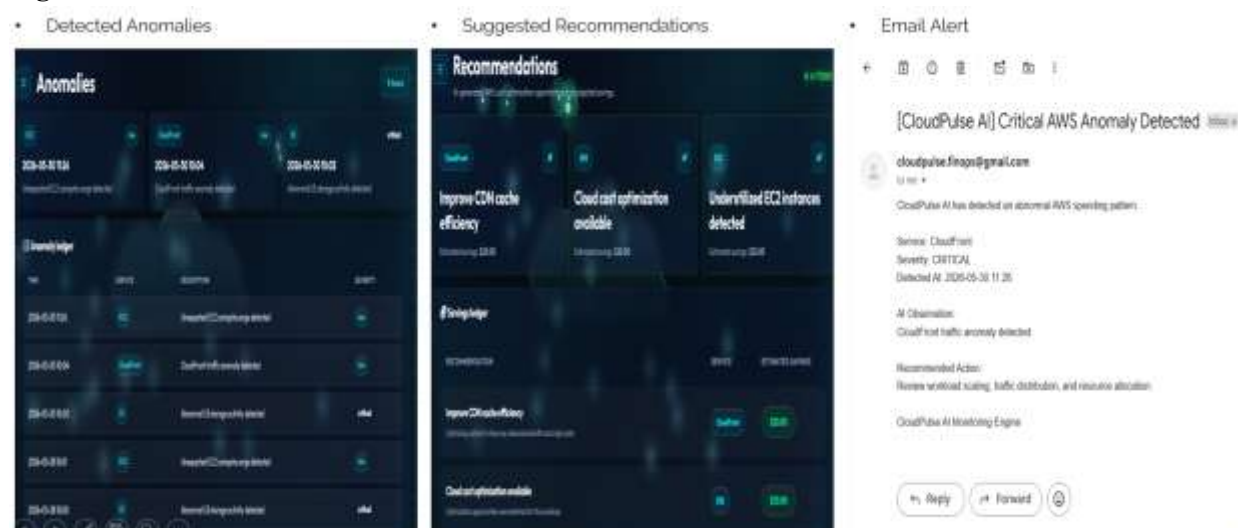
The screenshots below should be inserted as separate figure files during final submission. They document implemented behaviour rather than benchmark performance. Because no labelled production dataset or controlled user study was used, accuracy percentages, false-positive rates and user-satisfaction scores are intentionally not reported.

Figure 2. Dashboard summary and hourly AWS cost simulation



Legend: Dashboard cards show actual cost, forecasted cost and saving potential. The graph shows recent hourly spend points and the projected continuation.

Figure 3. Detected anomalies, recommendations and email alert



Legend: The anomaly ledger, recommendation ledger and critical anomaly email demonstrate the monitoring-to-action workflow.

5. Discussion

CloudPulse AI demonstrates that a compact Django application can present cloud-cost signals in a form that is easier to interpret than raw billing rows. Isolation Forest is suitable for a prototype because it can detect unusual observations without requiring labelled examples. Service-specific recommendations also keep the output understandable and directly connected to common cost-control actions.

The present implementation has important limitations. It uses simulated CSV records rather than production AWS billing exports. Recommendation saving values contain a randomised estimate and should be made deterministic before financial use. The dashboard forecast is a lightweight trend projection rather than a validated forecasting model. The prototype also does not perform automatic resource changes, ingest utilisation metrics or verify whether a recommendation has been implemented.

These limitations define the next engineering steps: integrate AWS billing exports and utilisation metrics, evaluate anomaly detection against a labelled dataset, adopt reproducible saving calculations, validate forecasting methods and add role-aware reporting. Multi-cloud support may be considered after the AWS workflow is validated.

6. Conclusion

This paper presented CloudPulse AI, an AWS-focused FinOps advisory prototype for cost monitoring, anomaly detection and resource optimisation. The application combines simulated hourly billing records, Isolation Forest outlier detection, short-term expenditure projection, rule-based recommendations and critical-anomaly email alerts within an authenticated Django interface. The prototype provides a practical foundation for further evaluation with production billing data and more rigorous financial models.

Key Points

- CloudPulse AI integrates AWS-oriented cost monitoring, anomaly detection and optimisation recommendations in one Django application.
- Isolation Forest detects unusual hourly billing observations using cost, time and service-related features.
- Service-specific recommendations translate billing observations into readable actions with indicative saving estimates.
- Critical manually triggered anomalies can generate email alerts for registered recipients.
- The current prototype uses simulated data; production ingestion and formal accuracy evaluation remain future work.

Acknowledgement

The authors acknowledge the support of the Department of Computer Science and Engineering, Ashokrao Mane Group of Institutions, Maharashtra, India.

Conflict of Interest

None declared.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

1. Deochake S. Cloud Cost Optimization: A Comprehensive Review of Strategies and Case Studies. arXiv preprint arXiv:2307.12479. 2023.
2. Khan AQ, Matskin M, Prodan R, Bussler C, Roman D, Soyulu A. Cost modelling and optimisation for cloud: a graph-based approach. J Cloud Comput. 2024;13:122.

3. Luong NC, Wang P, Niyato D, Wen Y, Han Z. Resource management in cloud networking using economic analysis and pricing models: a survey. *IEEE Commun Surv Tutor*. 2017;19(2):954-1001.
4. Srinivasa T. Cloud Cost Optimization Methodologies for Cloud Migrations. *Int J Intell Syst Appl Eng*. 2024;12(21s):4797-4809.
5. Ahmad S, Iqbal T, Munir EU, Ramzan N. Cost optimization in cloud environment based on task deadline. *J Cloud Comput*. 2023;12:18.
6. **Deochake S.** ABACUS: A FinOps-Based Cloud Cost Optimization Service for Budget Monitoring and Deployment Cost Estimation. *arXiv preprint*. 2024.

Figure Legends

Figure 1. Architecture of the proposed CloudPulse AI system showing data ingestion, AI processing, recommendation generation, and dashboard presentation layers.

Figure 2. Dashboard summary and hourly AWS cost simulation. Dashboard cards show actual cost, forecasted cost and saving potential. The graph shows recent hourly spend points and the projected continuation.

Figure 3. Detected anomalies, recommendations and email alert. The anomaly ledger, recommendation ledger and critical anomaly email demonstrate the monitoring-to-action workflow.

Authors

Mr. Shivprasad Satish Mali is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institutions ,kolhapur. He is publishing this paper under the esteemed guidance of Mrs. Almas A. Mahaldar.



Mr. Pratik Sanjay Gaikwad is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institutions ,kolhapur. He is publishing this paper under the esteemed guidance of Mrs. Almas A. Mahaldar.



Mr. Roshan Prakash Bhatale is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institutions ,kolhapur. He is publishing this paper under the esteemed guidance of Mrs. Almas A. Mahaldar.



Mr. Vivek Raghunath Bachate is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institutions ,kolhapur. He is publishing this paper under the esteemed guidance of Mrs. Almas A. Mahaldar.



Mr. Prathamesh Tukaram Mhetar is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institutions ,kolhapur. He is publishing this paper under the esteemed guidance of Mrs. Almas A. Mahaldar.



Mrs. Almas A. Mahaldar is currently working as an Assistant Professor, Department of Computer Science and Engineering, Ashokrao Mane Group of Institutions, Kolhapur. She has been guiding and mentoring students in various academic and research activities.

