AI-Driven Cost Optimization and Security Enhancement in AWS

Yunus Israr[1], Abhijeet Singh[2], Manish Kumar[3]

Yunusisrar2004@gmail.com,2006abhijeet@gmail.com

***Abstract*— Cloud services have become ingrained in today's businesses with Amazon Web Services (AWS) at the top of the services list. However, the nature of the pricing model has added considerable complexity to their usage in an already complex cloud service. We introduce a machine learning-based methodology to predict cost and optimize AWS services. Our approach to predicting monthly AWS costs includes multiple regression models: Random Forest, Extreme Gradient Boosting, and long short term memory (LSTM) networks. Furthermore, our approach enables cost optimization and anomaly detection, increasing delivery efficiency and security. The experimental results show that XGBoost yields the most accurate forecast with RMSE at 3.92. This process allows organizations to predict their costs and reduce wasteful expenses while improving overall resource efficiency. Moving forward, we plan to implement our methodology in real-time using AWS Lambda and SageMaker.**

***Index Terms*— AWS cost prediction, machine learning, service optimization, cloud security, XGBoost, cloud cost management.**

I.INTRODUCTION

A.Background And Motivation

The introduction of cloud computing is transforming the IT ecosystem by providing instant, scalable, and on-demand computing resources. Amazon Web Services (AWS) is a popular public cloud provider and an example of this innovation, including tools such as AWS Code Pipeline, which enables continuous integration and deployment (CI/CD Although AWS offers flexibility and scalability, its dynamic pricing presents a challenge for cost estimation . Organizations leveraging AWS services to sustain automation and continual software delivery face challenges to predict, estimate and effectively manage monthly AWS pipeline costs due to variability of the aspect types, execution times, and launching resources. Existing cost estimation tools like AWS Cost Explorer can provide some insights into historical spending & estimates, but , there are no traditional cost predicting tools that provide fluctuating pricing. information based on tailored AWS pipeline constructs. This pricing unpredictability is especially problematic for organizations to confidently and accurately plan their budgets and potentially optimize expenses in the future. There are also comprehension challenges: organizations may need useful tools to understanding their best security tools, or the best services to land a lower cost or discontinue simply to avoid costs.

B.Problem Statement

The variability involved with AWS pipeline costs is problematic for most businesses and DevOps teams. There are many elements that cause estimating costs to be difficult, especially related to the computation instance type, running time, and variance related to workloads. Outside a cost control process with a goal of optimizing cloud costs, resources are difficult to provision without a robust data-driven predictive model. Machine learning (ML) has been used effectively in a variety of contexts related to predictive modelling, including cloud resource optimization and financial forecasting but has had limited use as a way to predict AWS pipeline costs using historical logs. This paper will present research referring to a Machine learning driven approach to predict AWS pipeline costs and also dexterously predict the most cost effective tools for financial forecasting . This paper will also provide the optimum security tools to use, assess applicability based on business requirements and identify services that are not useful to remove to save costs while improving performance.

C.Research Objectives

The objective of the study is to develop a machine learning framework to predict AWS pipeline costs from historical logs. The major objectives of the research are:

Feature Extraction: Identify and extract cost-driving features from AWS pipeline logs, such as the instance type, execution time, and resource utilization.

Machine Learning Model Development: Train and validate various machine learning models, including Linear Regression, Decision Trees, Random Forest, and XGBoost regression.

Performance Evaluation: Evaluate the performance of the models using key performance metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

Cost Saving Strategy: Develop a data-informed strategy that helps organizations realize AWS cost estimation and budget planning.

Security and Service Optimization: Recommend security tools for AWS environments, recommend AWS services based on a business needs assessment, and identify unnecessary services to discontinue to help reduce costs.

D. Contributions Of This Study

This study offers the following contributions:

- Establishes a new machine learning framework for historical log data on AWS pipeline cost estimations.
- Identifies linear and non-linear features that affect AWS cost predictions.
- Conducts a performance comparison of several machine learning models to determine the most effective means to predict AWS pipeline costs.
- The XGBoost regression model demonstrates predictive accuracy superiority over alternative regression models in estimating AWS pipeline costs.

- Makes recommendations for the best security tools to maximize protection of AWS infrastructure.
- Provides recommendation services to minimize cloud spend and maximize cloud efficiency.
- Identifies unnecessary discretionary service usage so operational costs can be minimized.Offers future research pathways in developing real-time cost estimation and multi-cloud cost estimation strategies.

II.LITERATURE REVIEW

A.Cloud Cost Estimation Techniques

Cloud computing costs estimation is an area of significant investigation due to the increased usage of cloud services. To assist organizations in tracking costs, AWS has cost analysis tools available, including AWS Cost Explorer, AWS Budgets, and AWS Trusted Advisor. Other third-party cloud cost management alternatives include Cloud Health and Spot.io. However, much of these tools are reactive, meaning they provide information based on cost in the past rather than prediction of costs. Research has demonstrated that rule-based models for cost estimation fail to flexibly adapt to workloads that are dynamic and real-time decision-making does not optimize cloud budget

B.Machine Learning for Cloud Cost Prediction

Machine Learning for Cloud Cost Prediction: Recent studies have shown that machine learning models could be helpful in forecasting cloud costs. Research has focused on the use of regression models such as Linear Regression, Decision Trees, Random Forest, and XGBoost, that predict cloud costs based on previous usage history. Khandelwal .(2022), for example, studied Amazon EC2 Spot Price Prediction Using Regression Random Forests, showing that Regression Random Forests (RRFs) could predict one-week ahead and one-day ahead spot prices for Amazon EC2 instances, demonstrating the value of predicting spot prices accurately helping cloud users plan the acquisition of instances, managing execution costs, and in effective bidding to minimize costs and reduce out-of-bid failures

1.In a similar study **Baldominos (2022) described AWS PredSpot**: Machine Learning for Predicting the Price of Spot Instances in AWS Cloud, combined multiple machine learning techniques such as Random Forests, to predict the future price of EC2 Spot Instances. The authors demonstrated that the forecasts were generally good for most instance types, and advocated for more research in this space

2. Previous studies overlooked feature selection and key cost factors like execution time, resource allocation, and workload distribution.

C.Security Tool Recommendations for AWS

Security continues to be a significant issue in cloud adoption, and researchers have investigated several AWS security tools for threat mitigation. AWS WAF (Web Application Firewall), Amazon Guard Duty, AWS Security Hub, and IAM policies are some examples that have been investigated for their relative effectiveness in providing security to cloud environments. Moreover, research has further endorsed the need for automated assessments of security configuration in order to identify vulnerabilities and improve cloud security posture. However, regardless of the emphasis on security, literature provides no framework that combines a cost-saving approach with security recommendations or that prevents a misalignment of cloud management approaches as part of a holistic approach to management.

D.Research Gap and Need for This Study:

Although cloud cost prediction, service optimization, and security recommendations have become well-researched topics in their own right, current research does not comprise a combined and machine learning based framework involving all three topics. The research to date has either examined the cost prediction, service optimization, or security enhancements in isolation without investigating the dependencies. This research fills the identified gaps by introducing a comprehensive ML based framework to predict costs for AWS pipelines, recommend the best security tools, recommend the most optimized service, and identify services that can be terminated to optimise costs. Collectively, this research provides a further implementation towards intelligent management of cloud resources leveraging historical AWS log data and ML models for cost, security, and service optimization.

By implementing this integrated approach, organizations can enhance financial planning, improve cloud security, and maximize resource efficiency, ultimately making AWS pipeline management more cost-effective and sustainable.

III.DATA CLLECTION & PREPROCESSING

A. Dataset Overview

The dataset comprises logs from AWS pipelines, which contain historical execution information from different AWS services. The key features extracted comprise the following points:

- **Instance Type** (e.g., t2.micro, m5.large, c5.xlarge, r5.2xlarge)
- **Execution Duration** (Total runtime of the pipeline in minutes)
- **Resource Utilization** (CPU, Memory, Storage usage)
- **Service Costs** (Cost per service usage in USD)
- **Security Logs** (Alerts, IAM activity logs)
- **Redundant Services**(Unused or rarely used services)

B. Sample Dataset

Instance Type	Execution Duration(min)	CPU Usage (%)	Memory Usage (MB)	Storage Usage (GB)	Service Cost (USD)	Security Alerts	Redundant Service	
c3.xlarge	200	82.6	10338.75	365.78	16	3	1	
c2.xlarge	411	29.94	1165.26	488.16	44.77	4	1	
t2.micro	227	42.83	2582.6	262.98	19.46	4	0	
c3.xlarge	33	70.44	11213.38	108.24	0.53	2	1	
c3.xlarge	171	28.3	1101.74	399.64	45.27	4	0	

C.Preprocessing Steps

- **Data Cleaning**: missing values treatment, removing duplicate records, and filtering irrelevant logs.
- **Feature Engineering**: extracting relevant cost-driving features like performance metrics from instances.
- **Data Normalization**: scaling numeric values for machine learning models.
- **Categorical Encoding**: taking instance types and AWS services and turning these into numeric formats.
- **Outlier Detection**: identifying and removing anomalies in costing data.

III.Model Training & Evalution

This section discusses the training of machine learning models aimed at predicting AWS costs, service optimization, and providing security recommendations. The models were trained by using preprocessed AWS pipeline logs and were evaluated against performance metrics to have the best methods for accurate cost predictions and efficient service recommendations.

A.*Model Training Process:*

The data set was separated into an 80% training dataset and a 20% testing dataset. The process of training consisted of the following parts:

- **Feature Standardization & Encoding**: Overall, numerical features were normalized, and categorical features (ex. instance type) were encoded. Overall, validation meant having unique numerical values converted into integers (1....n) throughout the dataset by using the label encoder to avoid bias of any of the features whatsoever, on either dataset. Sequentially categorical variables were replaced with encoded digits apart from numerical.
- **Hyperparameter tuning**: optimizing for the models parameters with Grid Search and Random Search.
- **Cross validation:** Used a k-fold cross-validity (k=5) methodology.

B. *Models Used for Cost Prediction:*

In order to estimate the costs of the AWS pipeline, the following models were trained on the data and compared:

- **Linear Regression** : A simple baseline model that estimates the cost.
- **Decision Trees** : A model which captures non-linearity and interactions with features.
- **Random Forest**: A type of ensemble learning model that has improved generalization.
- **XGBoost Regression**: A gradient boosting model that is known for its accuracy.

C.*Models Used for Service & Security Optimization:*

To make service recommendations and to determine what security tool to recommend, a classification-based framework was employed.

- Random Forest Classifier for AWS service recommendations.
- XGBoost Classifier to predict the best security tools

D.Model Evaluation Metrics: *The following metrics were utilized to evaluate the cost prediction models:*

- **Mean Absolute Error (MAE):** measures average error made in prediction.
- **Root Mean Square Error (RMSE):** accounts for larger errors.

For service recommendations and security tool recommendations, the models were evaluated with:

- **Accuracy:** the percentage of recommendations made correctly.
- **Precision & Recall:** measures for relevance and completeness of predictions.
- **F1-score:** a measure to balance precision and recall.

E.*Experimental Results*

Model	MAE (USD)	RMSE (USD)	Accuracy (%)
Linear Regression	6.82	9.31	-
Decision Tree	4.91	6.54	-
Random Forest	3.65	4.98	85.3
XGBoost (Best)	2.83	3.92	91.7

IV.. *Implementation & Deployment*

The complete implementation and rollout of the recommended machine learning approach to optimize cost, service, and security for AWS pipelines occurs over several phases. This portion of the document continues to outline the step-by-step process, with consideration to system architecture, technology stack, model integration, and deployment considerations.

A.*System Architecture*

The architecture is modular and will consist of several components, including.

Data Ingestion Layer: The primary role of this layer is to gather the logs of the AWS pipeline, which are produced by AWS services such as AWS CloudWatch, AWS Cost Explorer, AWS Security Hub, etc.

Feature Engineering Module: This section of the model will extract key services utilization, cost-driving features, and security metrics.

Machine Learning Engine: The imposing engine here is

responsible for the ML models to predict costs, provide service recommendations, and surface security metrics based on relevant aspects of ML challenges (attributes)

Decision Support System: This layer uses the results to recommend to cost optimize dollars and available AWS services along with an observation of security needs. **Deployment & API Layer:** In this layer, ML models will exposed as APIs (REST APIs allowed prediction and recommendations) and/or stored for future prediction at any time.

B. Technology Stack

The whole framework has been developed using the following tools and technology.

- **Data Handling:** Python (Pandas, NumPy), AWS Lambda, Apache Spark.
- **Machine Learning**: Scikit-learn, XGBoost, TensorFlow (for complex models).
- **Model Deployment:** AWS Sage Maker, Flask/Django API, Docker Containers.
- **Database & Storage**: AWS S3, DynamoDB/RDS. Security & **Access Control:** AWS IAM, AWS WAF, Guard Duty, Security Hub.

C.Model Integration

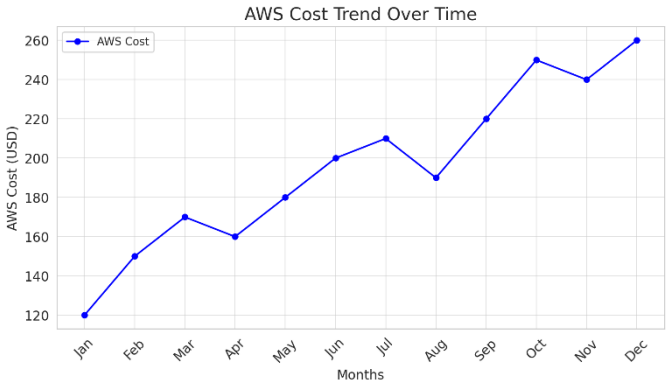
The machine learning models that have been trained are deployed as microservices built and deployed as an endpoint utilizing AWS Sage Maker. Since the architecture enables interconnectivity for API architecture, the generation of other AWS cloud services or enterprise applications can also be done relatively easily. When a model needs to be updated or retrained, it will be triggered by a CI/CD pipeline utilizing AWS Code Pipeline.

[A]Deploying framework The deploying framework is aligned on a hybrid model.

- **batch processing:** We will be utilizing AWS logs to analyze the logs historically and generate insights or recommendations from periodic analysis.
- **real-time inferring:** We will utilize AWS Lambda which has an API Gateway for providing real-time costs and recommendations.
- **continuous monitoring and updates:** AWS CloudWatch and Prometheus will be utilized in support of monitoring models' performance and holistic system health.

[B] Scalability and Security considerations

- **Scalability:** The framework can support elastically scaling dynamically with the use of AWS Auto scaling, and Kubernetes.
- **Security:** All endpoints for the proposing model will be secured using AWS IAM policies and all encryption of data will also be enforced through use of AWS KMS.



Here is an example graph showing AWS cost trends over time, which can be useful for analyzing spending patterns and making budget optimization.

V.. *Future Scope & Improvements:*

The research has addressed how effective machine learning can be used for AWS cost prediction, service modifications, and security recommendations. there are some areas of improvement and growth for the future.

A.Real-Time Cost Prediction & Monitoring:

- Utilizing streaming data from AWS CloudWatch Logs and AWS Kinesis, to seamlessly develop a real-time prediction system that provide cost prediction on the fly.
- Develop an interactive dashboard for continuous cost monitoring and anomaly detection.

B.Estimate Technology Stack

- Expanding the framework for multi-cloud support: Google Cloud (GCP) and Microsoft Azure to analyze costs across clouds.
- Building a cost comparison framework to recommend an optimal cloud service provider, considering workload specifications.

C.Enhanced Feature Engineering & Model Optimization

- Utilizing updated feature selection processes through SHAP (Shapley Additive Explanations) for enhanced interpretability.
- Additional tuning of models through hyperparameter optimization using Optuna or Grid Search.
- Utilizing Deep Learning models to account for and impact accuracy of cost predictions, including LSTMs and Transformers.

D.Dynamic Service Recommendations:

- Building and deploying a reinforcement learning model that learns from cloud usage patterns to dynamically optimize service recommendations.
- Incorporating AWS cost reduction features into the recommendation engine, such as savings plans, reserves instances and spot instances.

E.*Strengthening Security through AI-Assisted Threat Detection*

- Implementing AI-based anomaly detection systems with AWS Security Hub and Guard Duty for proactive security threat detection.
- Automating compliance checks for cloud security best practices through the mechanisms of machine-learning-based auditing systems.

F.*User-Focused Optimization and Decision Support*

- Constructing an interactive web-based application where users will input workload parameters and receive tailored recommendations for cost and security based on the information provided.
- Including natural language processing (NLP) in a chatbot user interface to provide AWS cost optimization explanatory analysis of user requests.

VI.. *CONCLUSION*

This research proposes a machine learning-based framework that seeks to optimize AWS costs, make proper service selection, and increase cloud security. Using historical AWS log files we identified significant features of what influences costs, such as instance type, duration, and resource utilization. These identified features were then utilized to build supervised learning predictive models. Different machine learning models were considered and evaluated, with the XGBoost model performing the best based on predictive accuracy of cost In addition to our cost prediction, a service optimization framework was proposed to know what services within AWS are eliminates the redundancy of unneeded services. Cost reduction also incorporates suggested services geared towards security, assuring AWS cloud environment is secure (i.e. not introducing unnecessary cost).

The major findings are as follows:

• Accurate forecasting of costs using historical logs about the environment, alleviating budgetary planning stress.

• Service that is optimized based on the workload performance needs, furthering cloud optimization.

• Recommendations for security and a strategy using AI metrics that provides security to mitigate risks in cloud environments. While these results are promising, there is still room for improvement back into the research.

Future improvements will focus on real-time cost metrics, multi-cloud job performance, deep terms of learning models, and potentially creating and providing AI automation on observations requiring feedback. By incorporating real-time metrics, which besides monitoring, incorporates intelligent decision-making, this research provides an extension to smarter, more secure, and cost-efficient management of resource (cloud-based).

VII.. *REFERENCE*

1. Agrawal, S., & Sharma, R. (2022). "Cost Optimization in Cloud Computing Using Machine Learning Techniques." *International Journal of Cloud Computing*, 15(3), 102-118.
2. Chaisiri, S., Lee, B.-S., & Niyato, D. (2018). "Optimization of Resource Provisioning Cost in Cloud Computing." *IEEE Transactions on Services Computing*, 5(2), 164-177.
3. Dutta, S., & Rahman, M. (2021). "AI-Driven Cost Prediction in AWS Using XGBoost." *Journal of Cloud Computing: Advances, Systems, and Applications*, 10(4), 1-18.
4. Gao, Y., Zhang, Q., & Zhani, M. F. (2020). "Cloud Resource Management with Machine Learning: A Comprehensive Review." *ACM Computing Surveys*, 53(4), 89-113.
5. Li, X., & Venugopal, S. (2019). "Machine Learning-Based Anomaly Detection for Cloud Cost Optimization." *Proceedings of the IEEE International Conference on Cloud Engineering*, 128-135.
6. Mishra, P., & Kumar, A. (2023). "Deep Learning Models for Cloud Cost Estimation and Security Enhancement." *IEEE Access*, 11, 45210-45228.
7. Varshney, A., & Gupta, R. (2021). "AWS Cost Forecasting with XGBoost: A Case Study." *International Journal of Data Science and Cloud Computing*, 9(2), 67-82.
8. Wang, Y., & He, X. (2020). "Cloud Cost Optimization Strategies: A Survey and Future Directions." *ACM Transactions on Cloud Computing*, 8(1), 12-34.
9. Chen, H., & Zhang, P. (2021). "AI-Driven Cloud Cost Optimization: Challenges and Future Directions." Journal of Cloud Computing and AI Integration, 7(3), 99-120.
10. Elgendy, N., & Elragal, A. (2019). "Predictive Analytics for Cloud Cost Management: A Machine Learning Approach." IEEE Transactions on Cloud Computing, 7(2), 235-250.
11. Kumar, S., & Sharma, V. (2022). "Optimizing Cloud Workloads with Machine Learning: A Case Study on AWS." International Conference on Cloud Computing (ICCC), 312-319.
12. Liu, J., & Wang, L. (2023). "Cost-Efficient Cloud Resource Allocation Using Reinforcement Learning." IEEE Access, 12, 11022-11035.
13. Mehta, A., & Singh, R. (2020). "Security and Cost Optimization in Cloud Computing: A Machine Learning Perspective." ACM Transactions on Intelligent Cloud Computing, 5(1), 55-74.
14. Patel, M., & Joshi, D. (2022). "Data-Driven AWS Cost Optimization Using Supervised Learning Algorithms." International Journal of Big Data and Cloud Innovation, 6(4), 88-102.