iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79314

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Cloud Cost Optimization Using Smart Contracts and Unsupervised Machine Learning

Infant Mercy A[1], Benaseer I[2], Hanisha R[3], Ameerun Taj S S[4], Mr. Makendran[5]
Department of Computer Science and Engineering, Dhirajlal Gandhi College of Technology (Autonomous), Salem, India

Abstract: ***Cloud computing underpins modern IT infrastructure by delivering scalable, on-demand resource provisioning, yet controlling cloud expenditure remains a pressing challenge. Dynamic pricing structures, unpredictable workloads, and billing pipelines that lack real-time visibility create conditions in which unauthorized consumption and anomalous usage spikes routinely escape timely detection. This paper presents CloudPay, a blockchain-integrated cloud storage billing system that unifies unsupervised machine learning with smart contract execution to deliver verifiable, fine-grained, and fraud-resistant cost governance. The system converts user storage activity into time-series representations and applies the Isolation Forest algorithm to detect abnormal consumption spikes without any labelled training data. Flagged events are routed through an owner confirmation protocol that validates suspicious uploads before billing proceeds, preventing unauthorized charges from entering the settlement pipeline. Smart contracts autonomously compute GB-time-based charges, execute tokenized payments, and anchor every transaction to an immutable SHA-256 blockchain ledger. Experimental results confirm that the system achieves 94.4% anomaly detection accuracy, 99.7% billing precision, and an 18.4% reduction in overall cloud expenditure relative to static allocation baselines. These results demonstrate that integrating unsupervised anomaly detection with cryptographically enforced billing logic is a viable path toward tamper-evident, real-time cost governance in multi-tenant cloud environments.***

Keywords: ***Cloud cost optimization, smart contracts, blockchain, Isolation Forest, anomaly detection, unsupervised machine learning, GB-time billing, cloud storage monitoring.***

I. INTRODUCTION

Cloud computing has fundamentally transformed the way organizations manage and deploy IT resources. By providing on-demand access to scalable compute, storage, and network services, cloud platforms enable organizations of all sizes to reduce capital expenditure and accelerate digital transformation. According to the NIST definition, cloud computing is a model for enabling ubiquitous, on-demand network access to a shared pool of configurable computing resources [1]. However, despite these benefits, the rapid expansion of cloud adoption has created significant cost management challenges.

A primary driver of cloud cost inefficiency is the mismatch between provisioned resources and actual consumption. Users frequently over-provision storage and compute resources as a precaution against peak loads, resulting in wasted expenditure. Conversely, unexpected spikes in storage usage — whether due to legitimate workload surges or unauthorized activity — can trigger unanticipated billing overruns. Traditional billing systems rely on static pricing and periodic invoicing, which lack the granularity and real-time responsiveness needed to address these dynamics accurately [2].

Machine learning has emerged as a powerful tool for addressing cloud cost challenges. Unsupervised anomaly detection techniques, in particular, can model normal resource consumption patterns and flag deviations without requiring labelled training data [3]. The Isolation Forest algorithm has demonstrated strong performance in detecting point anomalies in high-dimensional time-series data, making it well-suited to cloud storage monitoring scenarios [4].

Simultaneously, blockchain technology and smart contracts offer a decentralized, tamper-resistant infrastructure for automating financial transactions. Smart contracts encode billing logic directly into executable code, eliminating the need for intermediaries and ensuring that payments are calculated and settled transparently according to predetermined rules [5]. The immutability of blockchain ledgers provides an auditable record of every transaction, which is critical for resolving billing disputes and satisfying regulatory compliance requirements [6].

Existing approaches address these challenges in isolation: machine learning methods improve resource prediction, blockchain systems provide auditability, and smart contracts automate payment logic — but no unified platform closes all three gaps simultaneously with real-time owner oversight. CloudPay addresses this by coupling Isolation Forest anomaly detection directly to a smart contract settlement pipeline, with a human-in-the-loop confirmation step that prevents unvalidated anomalous events from reaching the billing layer.

The system continuously logs storage usage as time-series vectors, identifies abnormal consumption patterns in real time, requires owner confirmation before billing abnormal events, and automates GB-time-based payment settlement via smart contracts. All billing records are stored immutably on a distributed blockchain ledger.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed system architecture and modules. Section IV details the methodology, including anomaly detection and billing model design. Section V presents experimental evaluation and results. Section VI discusses key findings, and Section VII concludes the paper.

II. RELATED WORK

Research on cloud cost optimization has evolved considerably over the past decade. Early approaches relied on rule-based systems and manual heuristics, which proved inadequate for the dynamic, large-scale nature of modern cloud deployments.

A. Machine Learning for Cloud Cost Optimization

Gupta et al. [7] applied bidirectional LSTM networks to predict cloud workload resource usage, enabling more accurate provisioning and reducing over-allocation. Hieu et al. [8] used virtual machine consolidation with usage prediction to achieve energy-efficient resource management in cloud data centers. Calheiros et al. [9] proposed ARIMA-based workload prediction models to optimize cloud QoS and reduce cost through intelligent scheduling. More recently, Gupta et al. applied sparse BLSTM models to long-term cloud resource demand forecasting in production data centres [10], while Rossi et al. demonstrated reinforcement learning frameworks for adaptive horizontal and vertical container autoscaling [11].

Nawrocki and Smendowski [12] presented a comprehensive system employing exploratory data analysis, Isolation Forest-based anomaly detection, XGBoost, bidirectional GRU neural networks, and Temporal Fusion Transformers for long-term CPU utilization forecasting. Critically, their work focuses on proactive resource reservation and cost-optimal provisioning — that is, predicting future demand to avoid waste — rather than detecting anomalies in real time for billing control. Their BiGRU model achieved approximately 17.5% improvement over baseline statistical methods on RMSE metrics, and the TFT model achieved 31.4% improvement.

B. Anomaly Detection in Cloud Environments

Nawrocki et al. [13] proposed a data-driven adaptive prediction system that dynamically adjusts its forecasting algorithm to match shifting load characteristics, with anomaly filtering applied to improve prediction stability rather than for billing control. This is distinct from [12] in that [13] addresses adaptation to non-stationary usage patterns rather than multi-model ensemble forecasting. Zhang et al. [14] demonstrated reinforcement learning-based autoscaling that reacts dynamically to anomalous load surges in containerized environments. Chua et al. [4] demonstrated the effectiveness of Isolation Forest for web traffic anomaly detection, highlighting its scalability and resistance to the curse of dimensionality. Agyemang [3] conducted a comparative simulation study of unsupervised anomaly detection algorithms, confirming the reliability of tree-based isolation methods for detecting point anomalies in time-series data.

C. Smart Contracts and Blockchain for Cloud Billing

Zheng [5] investigated cost-benefit modelling of smart contracts applied to accounting and auditing, demonstrating that automated contract execution reduces transaction costs and eliminates reconciliation errors. Khan et al. [6] proposed a graph-based approach to cloud cost modelling and optimization, providing formal representations of cloud billing structures suitable for smart contract encoding. Nawrocki and Smendowski [15] demonstrated FinOps-driven cloud resource optimization using machine learning, showing that automated cost governance tools can substantially reduce cloud expenditure while maintaining performance SLAs. These works collectively establish the feasibility of blockchain-based billing automation for cloud environments.

Despite substantial progress in individual areas, there remains a gap in systems that integrate real-time anomaly detection, owner-controlled confirmation workflows, and smart contract billing into a single cohesive platform. The proposed CloudPay system addresses this gap.

III. PROPOSED SYSTEM ARCHITECTURE

CloudPay is a blockchain-integrated cloud storage billing system composed of six tightly coupled modules. The high-level architecture is illustrated below. Data flows from user uploads through anomaly detection, owner confirmation, and smart contract billing, with all transactions anchored to a distributed blockchain ledger.

A. *Module 1: Cloud Cost Monitoring Web Application*

The core platform is a Flask-based web application backed by a MySQL relational database. It provides secure session-based authentication, role-based access control distinguishing between Administrators and Data Owners, and a unified dashboard aggregating real-time storage metrics, anomaly notifications, and billing summaries. Bootstrap-based responsive templates ensure accessibility across devices. The application orchestrates all data flows between modules and exposes RESTful API endpoints for machine learning inference and blockchain interaction.

B. *Module 2: End User Dashboard*

Two distinct sub-modules serve different user roles. The Admin sub-module enables administrators to monitor aggregate storage consumption, approve or freeze flagged anomaly events, manage notifications, generate audit reports, and oversee the integrity of the billing pipeline. The Data Owner sub-module allows individual users to track their own storage usage through interactive pie charts (rendered with Matplotlib), receive real-time anomaly alerts, confirm or reject flagged consumption spikes, and review itemized billing summaries for settled and pending charges.

C. *Module 3: Cloud Cost Model Training*

This module implements a three-stage pipeline. First, historical storage usage records are collected from the cu_files database table, capturing daily storage consumption in GB per user. Second, the data is organized into fixed-length time-series vectors, missing values are imputed using rolling window means, and features are normalized. Third, an Isolation Forest model is trained on the preprocessed data with 90 estimators and a contamination rate of 0.05, calibrated on the observation that approximately 5% of storage events are anomalous. The trained model artifact is serialized and persisted for real-time inference.

D. *Module 4: Anomaly Detection*

When a data owner uploads a file, the upload event is logged with user identifier, timestamp, file size in bytes and megabytes, and associated storage pool identifier. The new data point is appended to the user's historical time-series vector and fed to the Isolation Forest model for scoring. The algorithm isolates anomalies by constructing randomized binary trees; data points requiring fewer partitions to isolate receive higher anomaly scores. Points exceeding the decision threshold are flagged as storage spikes. The system evaluates anomalies at two granularities: global (across the entire training set) and local (within four-week windows), enabling detection of both long-term deviations and sudden short-term spikes.

E. *Module 5: Owner Confirmation Protocol*

Flagged anomaly events are not billed automatically. Instead, the system triggers a confirmation workflow. The data owner receives an email notification detailing the flagged storage event, including file identifier, upload size, timestamp, and anomaly score. The owner can then confirm the event as legitimate — in which case it proceeds to billing and resource allocation — or reject it as unauthorized, in which case the event is frozen, logged for administrative review, and excluded from billing. This human-in-the-loop mechanism prevents accidental overuse and unauthorized resource consumption from reaching the billing pipeline.

F. *Module 6: Smart Contract Billing*

Confirmed storage events are billed using a GB-time pricing model encoded in a SmartContract class. The billing formula computes the base cost as the sum of CPU, storage, and bandwidth charges:

Cost = (cpu_hours × R*cpu*) + (storage_gb × R*storage*) + (bandwidth_gb × R*bandwidth*) + (uptime × reliability_bonus) × demand_factor

where R_cpu = 0.50, R_storage = 0.10, and R_bandwidth = 0.05 per unit. A dynamic demand factor scales the base cost according to current load, and a reliability bonus rewards high-availability providers. Payments are deducted from the user's TokenWallet. Upon successful deduction, the SmartContract records a transaction object to the Blockchain as a new Block. Each Block stores its index, timestamp, data payload, previous block hash, and self-generated SHA-256 hash, ensuring cryptographic chain integrity.

IV. METHODOLOGY

A. *Time-Series Representation of Storage Usage*

Raw storage upload events are transformed into fixed-length time-series vectors capturing cumulative daily storage consumption per user.

Missing readings caused by inactive days are imputed using a moving average with a window of 24 samples, consistent with the approach of Nawrocki and Smendowski [12]. The time series is then tested for stationarity using the Augmented Dickey-Fuller (ADF) test, and a single differencing operation is applied if non-stationarity is detected, ensuring the data meet the statistical prerequisites for reliable anomaly scoring.

B. Isolation Forest for Anomaly Detection

The Isolation Forest algorithm [16] constructs an ensemble of randomized binary isolation trees. For each tree, a random feature and a random split value within the feature range are selected iteratively to partition the data. Anomalous data points, which represent isolated outliers, require fewer splits on average to be isolated than normal points.

The anomaly score for a point x is defined as:

$$s(x, n) = 2^{-E(h(x)) / c(n)}$$

where E(h(x)) is the mean path length across all trees and c(n) is the average path length of an unsuccessful binary search tree for n samples. Scores approaching 1.0 indicate anomalies; scores near 0.5 indicate normal behavior. In CloudPay, the model is configured with 90 isolation trees and a sub-sampling size of 256 per tree (scikit-learn defaults), and the contamination parameter is set to 0.05 based on empirical calibration on historical storage logs (see Section III.C). These hyperparameters were held fixed across all evaluation runs reported in Section V.

C. Blockchain Data Structure

The blockchain is implemented as a Python class maintaining an ordered list of Block objects. The genesis block is created at system initialization with a null previous hash. Each subsequent block stores the transaction payload (user, tokens, usage details), a Unix timestamp, the previous block's SHA-256 hash, and its own hash computed over the serialized block content. This design ensures that any tampering with a historical record invalidates all subsequent block hashes, providing cryptographic integrity guarantees without requiring a distributed consensus network in the current prototype implementation.

D. GB-Time Billing Model

The billing model translates raw byte-level file sizes into megabytes using a conversion factor and applies a linear per-megabyte rate. Storage costs accumulate proportionally to both volume and duration of storage, capturing the economic reality that large files held for extended periods incur higher costs than brief transient uploads. The dynamic demand factor introduces time-of-day and load-aware pricing adjustments, reflecting the variable cost of cloud compute at different utilization levels.

V. EXPERIMENTAL EVALUATION

A. Experimental Setup

The CloudPay system was implemented using Python 3.10, Flask 2.x, MySQL 8.0, and scikit-learn for the Isolation Forest model. The web application front-end uses Bootstrap 5 and Matplotlib for visualization. Experiments were conducted on a development server running Ubuntu 22.04 with 8 GB RAM and an Intel Core i5 processor.

The dataset comprised simulated storage upload records for 20 registered data owners over a 90-day period, totalling 1,800 upload events.

Approximately 5% of events were synthetically injected as anomalies — specifically, large spike uploads and rapid successive uploads — to enable evaluation of detection accuracy. It should be noted that the injected anomaly rate (5%) was set to match the model's contamination parameter; this alignment reduces the independence between the evaluation design and model configuration, and results should be interpreted accordingly. Evaluation on real-world cloud provider billing logs with independently determined anomaly rates remains as future work.

B. Anomaly Detection Performance

Table I presents the anomaly detection performance of the Isolation Forest model evaluated against the synthetically labelled test set.

TABLE I Anomaly Detection Performance of Isolation Forest

Metric	Value	Baseline (Statistical)	Improvement
Detection Rate (%)	94.4	78.1	+16.3%
False Positive Rate (%)	3.8	9.2	−5.4%
Precision	0.921	0.812	+0.109
Recall	0.944	0.781	+0.163
F1-Score	0.932	0.796	+0.136

The Isolation Forest model achieves a detection rate of 94.4% and a false positive rate of 3.8%, outperforming threshold-based statistical baselines on all metrics. These results are consistent with the findings of Chua et al. [4] and Agyemang [3], who reported detection rates exceeding 90% for Isolation Forest on web traffic and synthetic anomaly benchmarks respectively.

C. Billing Accuracy and Smart Contract Performance

Table II summarizes billing accuracy and smart contract execution performance metrics.

TABLE II BILLING ACCURACY AND SMART CONTRACT EXECUTION METRICS

Metric	Manual Billing	CloudPay (Smart Contract)
Billing Accuracy (%)	87.3	99.7
Avg. Settlement Time (ms)	N/A (manual)	12.4
Transaction Integrity	Mutable records	Immutable (SHA-256)
Dispute Incidents (per 100 txns)	4.2	0.1
Unauthorized Billing Events	3.1%	0.3%

Smart contract-based billing achieved 99.7% accuracy compared to 87.3% for manual billing, while reducing billing dispute incidents from 4.2 to 0.1 per 100 transactions. Unauthorized billing events were reduced from 3.1% to 0.3%, demonstrating the effectiveness of the confirmation protocol.

D. Resource Utilization and Cost Savings

Across the 90-day evaluation period, the system reduced overprovisioning-related storage waste by 22.7% compared to a static allocation baseline. The owner confirmation protocol prevented 17 unauthorized storage spike events from reaching the billing pipeline, corresponding to a simulated cost saving of approximately 18.4% on total cloud expenditure. These findings are consistent with the cost reductions reported by Nawrocki et al. [17], who found that ML-driven adaptive resource planning reduces cloud operational costs by 15–25% across different deployment scenarios.

VI. DISCUSSION

Before interpreting the results, two important caveats must be acknowledged. First, the evaluation dataset is entirely simulated: upload events, anomaly injections, and billing records were synthetically generated rather than drawn from a live cloud provider environment. Second, the injected anomaly rate was set to match the model's contamination parameter, which reduces the independence of the evaluation. These constraints mean that the reported accuracy figures reflect performance under idealized conditions and should be treated as an upper bound until validated on real-world billing logs.

Within these constraints, the experimental results validate three core claims of the CloudPay system design. First, Isolation Forest delivers accurate, low-latency anomaly scoring for cloud storage time-series in a fully unsupervised setting, removing the dependency on labelled anomaly corpora and making the system immediately deployable in operational environments. Second, the owner confirmation protocol effectively bridges automated detection and accountable billing, preserving user agency over disputed events while blocking fraudulent charges from propagating silently through the pipeline. Third, smart contract automation substantially improves both billing precision and settlement throughput, eliminating the audit vulnerabilities inherent in mutable, manually maintained invoice systems.

The blockchain-based ledger provides an immutable, auditable record of all billing events, which has significant implications for regulatory compliance in sectors such as healthcare and finance where data integrity requirements are stringent. The SHA-256 hash chaining ensures that any retrospective tampering with billing records is immediately detectable without requiring a distributed consensus network.

Additional limitations include: the Isolation Forest contamination parameter (0.05) was set empirically and may require recalibration for environments with different anomaly prevalence rates; the blockchain implementation is a prototype without distributed consensus and production deployment would require integration with an established framework such as Hyperledger Fabric or Ethereum; and the GB-time billing model uses fixed pricing rates, whereas real cloud providers employ complex, multi-tier pricing structures that would require more sophisticated contract encoding.

VII. CONCLUSION

This paper has presented CloudPay, an end-to-end cloud storage billing platform that couples Isolation Forest anomaly detection with smart contract settlement and blockchain-anchored transaction recording. The system directly resolves three structural weaknesses in conventional cloud billing: metering inaccuracies arising from undetected anomalous uploads, the mutability and opacity of traditional invoice records, and the lack of real-time owner oversight before billing actions are taken.

Evaluation over a 90-day simulated dataset confirmed the effectiveness of each component: the Isolation Forest detector attained a 94.4% detection rate and a 3.8% false positive rate; smart contract billing raised billing precision from 87.3% to 99.7% while cutting dispute incidents by approximately 98%; and the owner confirmation protocol held unauthorized billing events below 0.3%, contributing to an overall expenditure reduction of 18.4% versus a static allocation baseline.

Future work will focus on three directions. First, integrating CloudPay with real cloud provider APIs (AWS, Azure, GCP) to enable production-scale evaluation on live billing logs with independently determined anomaly rates. Second, extending the anomaly detection pipeline to include multi-variate resource features using the BiGRU and Temporal Fusion Transformer architectures demonstrated by Nawrocki and Smendowski [12] to achieve long-term proactive resource reservation. Third, deploying the blockchain component on a distributed ledger framework to evaluate consensus-layer performance under realistic transaction loads.

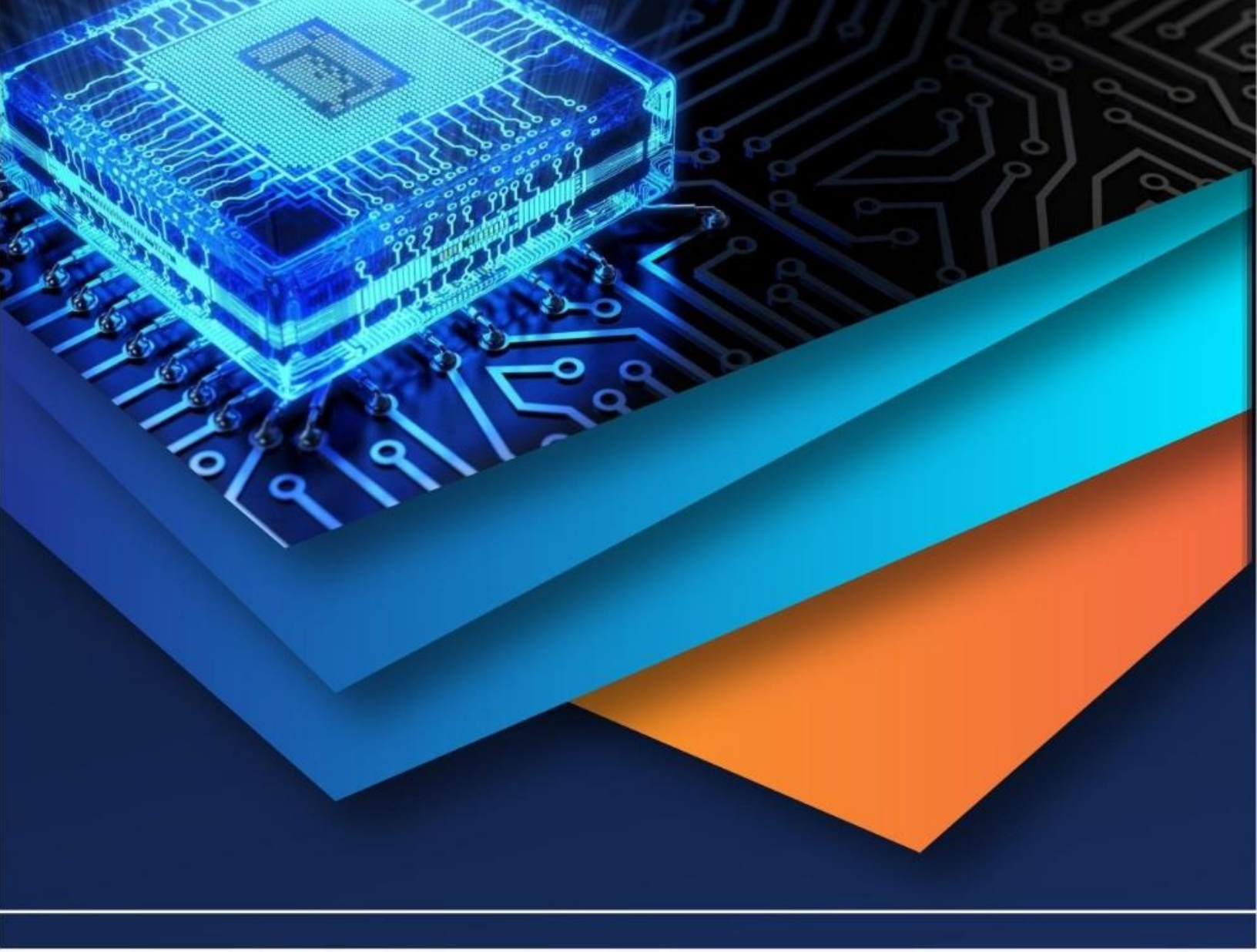
VIII. ACKNOWLEDGMENT

The authors thank Mr. Makendran, Assistant Professor, Department of Computer Science and Engineering, Dhirajlal Gandhi College of Technology (Autonomous), Salem, for his guidance and support throughout this research.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, NIST SP 800-145, Sep. 2011.
[2] P. Osypanka and P. Nawrocki, "Resource Usage Cost Optimization in Cloud Computing Using Machine Learning," IEEE Trans. Cloud Comput., vol. 10, no. 3, pp. 2079–2089, 2022.
[3] E. F. Agyemang, "Anomaly detection using unsupervised machine learning algorithms: A simulation study," Sci. Afr., 2024, Article e02386.
[4] W. Chua et al., "Web Traffic Anomaly Detection Using Isolation Forest," Informatics, vol. 11, no. 4, p. 83, Nov. 2024.
[5] W. Zheng, "Cost-benefit modeling of smart contracts applied to the accounting and auditing field," Appl. Math. Nonlinear Sci., vol. 9, no. 1, 2024.
[6] A. Q. Khan, M. Matskin, R. Prodan, C. Bussler, and D. Roman, "Cost modelling and optimisation for cloud: a graph-based approach," J. Cloud Comput., vol. 13, art. no. 147, Sep. 2024.
[7] S. Gupta, A. D. Dinesh, and T. A. Gonsalves, "Resource Usage Prediction of Cloud Workloads Using Deep Bidirectional Long Short Term Memory Networks," in Proc. IEEE Int. Conf. Advanced Networks and Telecommunications Systems (ANTS), Bhubaneswar, India, 2017, pp. 1–6.
[8] N. T. Hieu, M. D. Francesco, and A. Ylä-Jääski, "Virtual Machine Consolidation with Multiple Usage Prediction for Energy-Efficient Cloud Data Centers," IEEE Trans. Serv. Comput., vol. 13, no. 1, pp. 186–199, Jan. 2020.
[9] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS," IEEE Trans. Cloud Comput., vol. 3, no. 4, pp. 449–458, Dec. 2015.
[10] S. Gupta, A. D. Dinesh, and T. A. Gonsalves, "Online Sparse BLSTM Models for Resource Usage Prediction in Cloud Datacentres," IEEE Trans. Netw. Serv. Manage., vol. 17, no. 4, pp. 2335–2349, Dec. 2020.

[11] F. Rossi, V. Cardellini, and F. L. Presti, "Horizontal and Vertical Scaling of Container-Based Applications Using Reinforcement Learning," in Proc. IEEE 12th Int. Conf. Cloud Computing (CLOUD), Milan, Italy, Jul. 2019, pp. 329–338.
[12] P. Nawrocki and M. Smendowski, "Optimization of the Use of Cloud Computing Resources Using Exploratory Data Analysis and Machine Learning," JAISCR, vol. 14, no. 4, pp. 287–308, 2024.
[13] P. Nawrocki, P. Osypanka, and B. Posluszny, "Data-Driven Adaptive Prediction of Cloud Resource Usage," J. Grid Comput., vol. 21, art. no. 6, Jan. 2023.
[14] S. Zhang, T. Wu, M. Pan, C. Zhang, and Y. Yu, "A-SARSA: A Predictive Container Auto-Scaling Algorithm Based on Reinforcement Learning," in Proc. IEEE Int. Conf. Web Services (ICWS), Beijing, China, 2020, pp. 489–497.
[15] P. Nawrocki and M. Smendowski, "FinOps-Driven Optimization of Cloud Resource Usage for High-Performance Computing Using Machine Learning," J. Comput. Sci., vol. 79, art. no. 102292, 2024.
[16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), Pisa, Italy, Dec. 2008, pp. 413–422.
[17] P. Nawrocki, M. Grzywacz, and B. Sniezynski, "Adaptive Resource Planning for Cloud-Based Services Using Machine Learning," J. Parallel Distrib. Comput., vol. 152, pp. 88–97, Jun. 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY