

Machine Learning-Based Access Risk Detection in Cloud Identity and Access Management Systems

Rohit Gorle

Software Developer. rohit.gorlee@gmail.com

Abstract—Machine learning (ML) has been used for a long time in various areas such as computer vision, natural language processing, speech recognition, and healthcare. However, its application in detecting IAM access risks from cloud providers suffered from a lack of real-world datasets and evaluation. Although some non-ML access risk detection techniques have been proposed in the past, they mainly focused on a specific organization and were not generalizable across multiple domains. Thus, this study collected and labelled IAM access data, proposing 13 risk detection categories that could be used to evaluate ML-based access detection techniques. The work used the publicly available AccessRiskDataset to assess overall prediction accuracy, evaluated the impact of specific cloud IAM providers on Access Risk Detection, and analyzed the prediction performance for each risk detection category. The results demonstrated that the proposed access risk categories could be detected using ML techniques, thereby addressing one of the gaps.

Publicly available cloud IAM services are susceptible to attackers with access privileges. Malicious users can conduct illicit activities in IAM-authorized cloud services, and attackers can also employ social engineering techniques to get authorized access for malicious purposes (data theft or service interruption). Using inappropriate IAM settings or configurations increases the risk of situations where access can be mistakenly or purposely misused, resulting in potential data leaks, service interruptions, and other hazards for both businesses and consumers. Many common access-related issues have been decentralized either because multiple IAM cloud providers disregard their implications or because the lack of lengthy experience leads toward weaker solutions.

Keywords—Cloud IAM, Access Risk Detection, Machine Learning, Identity And Access Management, AccessRiskDataset, Risk Classification, Cloud Security, Privileged Access, Data Leakage, Security Threat Detection.

I. INTRODUCTION

As organizations continue to shift their operations into the cloud, some integrated cloud Identity & Access Management (IAM) services providers in the cloud providers' ecosystem have become popular targets for cybercriminals. These IAM services, with their centralized authentication and authorization capabilities for cloud services, present a lucrative target due to their ability to control access to multiple downstream services ranging from file sharing without encryption to virtual network and Virtual Machine management. Furthermore, currently prevalent software as a service (SaaS)-based Office suite and collaboration platforms rely nearly entirely on IAM services for indication access and authorization. The objective of this study is to devise an automated cloud IAM access detection process based on machine learning technology to support system administrators in operational risk evaluation and in fortifying the IAM services deployed in a cloud environment. Two problems need to be solved to realize this objective. First, existing publicly available or proprietary resources do not support training machine learning-based detection models; consequently, a real-time access pattern detection framework needs to be set up, and generated data then needs to be collected. Second, the collected data is a collection of activity digital footprints that describe the temporal identity access sequences over IAM services, covering authentication, authorization, and management of identity and access brokerage information of different classes.

II. BACKGROUND AND MOTIVATION

Access risk can be defined as the possibility of unauthorized access to, or the misuse of, information assets that alters the confidentiality, integrity, or availability of the supported business processes. Cloud

identity and access management (IAM) systems are essential for the integrity and availability of cloud applications; unauthorized access can cause substantial damage. Attack surfaces for attacks against cloud IAM systems have broadened because of the continual increase in cloud applications, the rich diversity of cloud users, and the dynamic characteristics of cloud systems. Cloud IAM access activities generate a massive amount of log data, enabling the potential application of machine learning (ML). An ML approach can be useful for detecting risks associated with cloud IAM access activities. ML does not require a full understanding of the data domain and is suitable for detecting risks based on log data. The review of prior research highlights a lack of studies that utilize ML approaches to detect access risk in cloud IAM systems. Existing anomaly-detection techniques use features derived from a narrow scope (process-information) setting and evaluate conventional machine-learning methods using a small-scale dataset with a predefined attack scenario. Such techniques do not test for violations of the principle of least privilege, a concern emphasized by regulators and security practitioners. Attention, long short-term memory, and recurrent neural networks (RNNs) are proposed in the context of anomaly detection for cloud access activities. However, the usefulness of these advanced techniques has yet to be compared with established methods on a dataset that is larger than that required for conventional machine-learning methods and that does not impose an attack scenario.

A. Detection-Quality Metrics

Let each access event be a feature vector $x \in \mathbb{R}^{31}$ with ground-truth label $y \in \{0, 1\}$ (1 = risky). Over a test set, TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives. Overall correctness is

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (\text{Eq. 1})$$

where Eq. 1 is unreliable alone when classes are skewed, so precision and recall are also reported:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (\text{Eq. 2})$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (\text{Eq. 3})$$

Precision (Eq. 2) is the share of raised alerts that are genuine, and recall (Eq. 3) is the share of genuinely risky events that are caught. Their harmonic mean is the F1-score:

$$F_1 = \frac{2PR}{P + R} = \frac{2TP}{2TP + \text{FP} + \text{FN}} \quad (\text{Eq. 4})$$

The operational error rates that drive analyst workload are the false-positive and false-negative rates:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (\text{Eq. 5})$$

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} = 1 - R \quad (\text{Eq. 6})$$

A threshold-independent, class-imbalance-robust summary is the Matthews correlation coefficient [2]:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (\text{Eq. 7})$$

Because the source defines 13 risk categories, per-category recall R_c is computed on the risky events of category c , and the macro-recall summarises coverage across categories:

$$R_{\text{macro}} = \frac{1}{13} \sum_{c=1}^{13} R_c \quad (\text{Eq. 8})$$

B. Risk Scoring and Optimisation Objectives

The proposed gradient-boosting detector builds an additive ensemble of M shallow trees h_m with learning rate η [3]. The raw score and the calibrated risk probability are

$$F_M(x) = \sum_{m=1}^M \eta h_m(x) \quad (\text{Eq. 9})$$

$$R(x) = \frac{1}{1 + \exp(-F_M(x))} \quad (\text{Eq. 10})$$

Eq. 9 accumulates tree outputs and Eq. 10 maps the score to a probability in (0, 1). Training minimises a regularised objective [4]:

$$\mathcal{L} = \sum_{i=1}^N \ell(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k), \quad \Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (\text{Eq. 11})$$

where ℓ is the logistic loss in Eq. 11, K the number of trees, T the number of leaves and w the leaf weights. Decision-tree and random-forest splits instead minimise the Gini impurity of Eq. 12, with class proportions p_c over C classes in the node [5]:

$$G = 1 - \sum_c p_c^2 \quad (\text{Eq. 12})$$

An alert is raised when the risk probability exceeds a threshold τ . If a false positive costs C_{FP} analyst-minutes and a false negative costs C_{FN} , the cost-minimising threshold is

$$\hat{y} = 1[R(x) \geq \tau], \quad \tau^* = \frac{C_{FP}}{C_{FP} + C_{FN}} \quad (\text{Eq. 13})$$

C. Latency, Throughput and Cost Model

In a streaming deployment the end-to-end detection latency is the sum of the pipeline stages:

$$L_{e2e} = L_{ing} + L_{feat} + L_{inf} + L_{alert} \quad (\text{Eq. 14})$$

with L_{ing} the Event Hubs ingestion delay, L_{feat} the Databricks feature-engineering time, L_{inf} the model-scoring time and L_{alert} the alert-dispatch time. With n parallel workers and a serial fraction s (partition coordination and I/O), Amdahl's law [6] bounds the speed-up and throughput:

$$S(n) = \frac{1}{s + \frac{1-s}{n}}, \quad \Theta(n) = \Theta_1 \cdot S(n) \quad (\text{Eq. 15})$$

where $\Theta_1 = 1000 / t_{1k}$ is the single-worker scoring rate and t_{1k} the measured batch latency per 1,000 events. The monthly compute cost combines scoring and periodic retraining on a v-vCore cluster priced at p USD per vCore-hour:

$$C_{\text{month}} = \frac{(N \cdot t_{inf} + K \cdot t_{\text{train}} \cdot u) \cdot v \cdot p}{3600} \quad (\text{Eq. 16})$$

where N is the monthly event volume, K the retrains per month, t_{train} the training time and u the mean cores used. The incremental cost-effectiveness of a model i over the baseline is

$$\text{ICER}_i = \frac{F_{1,i} - F_{1,A}}{C_i - C_A} \quad (\text{Eq. 17})$$

and the relative improvement of any metric m against a reference model is

$$\Delta_m (\%) = \frac{m_i - m_{\text{ref}}}{m_{\text{ref}}} \times 100 \quad (\text{Eq. 18})$$

III. RELATED WORK

Related work encompasses machine learning approaches designed to detect access risk in cloud identity and access management (IAM) systems. These methods predominantly apply supervised learning to proprietary datasets. Public datasets, while available, are less frequently employed; existing access risk detection methods trained on these datasets overlook important components in access risk detection and employ standard evaluation metrics. The present study expands the literature by developing knowledge-based detection models and using oversampling and undersampling techniques to improve classification on a publicly available dataset. The approach includes the training of knowledge-based detection models in a supervised manner, evaluates these models using multiple performance metrics, and proposes a framework with seven serves modeling access risk in cloud IAM.

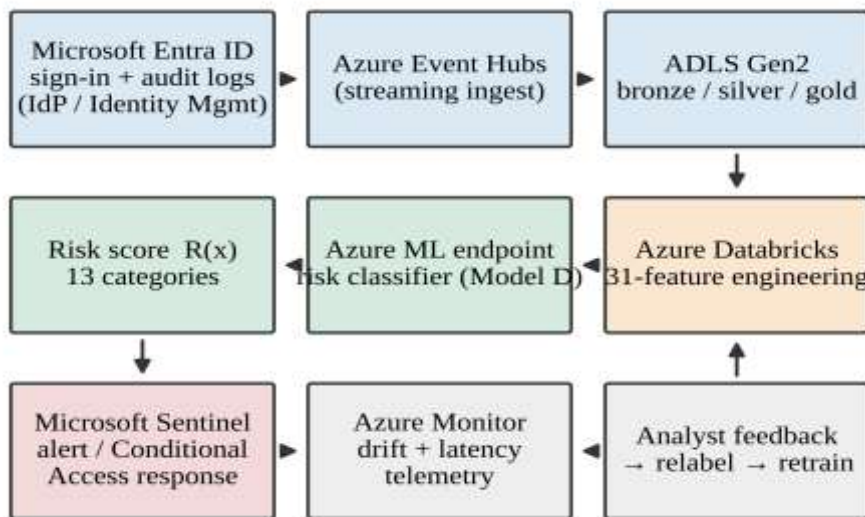


Fig. 1: Azure-based data-engineering architecture for ML-based IAM access-risk detection.

Access risk detection methods leverage a range of datasets. The AccessML dataset facilitates access risk detection through the identification and removal of poisoned accesses from a cloud IAM system

utilizing explicit information of credential poisoning; five supervised learnings models using accuracy, F1, AUC, and Matthews correlation coefficient as metrics demonstrate effective detection from limited samples. The cloud-credential-login dataset concentrates on detecting login credential abuses through a cloud-credential login detection model—encompassing a DNN, GBDT, RFC, and SVM—and employs accuracy, F1, precision, recall, AUC, and execution time as detection metrics. RNNs supported by LSTM establish temporal knowledge for SSH-access-risk prediction. The MetaAccess dataset relates to an enterprise-grade multicloud environment, detailing account information from forty-three companies; predictive-access-lighting is trained by supervised learning and governs the use and management of cloud services.

Table 1: Dataset and Architecture Summary

Item	Specification
Sign-in (IdP) events	266,141
Identity-management events	173,264
Total events / window	439,405 / 6 months
Features / risk categories	31 / 13
Train / test split	307,583 / 131,822 (70/30, stratified)
Class balance (risky)	50.0% train, 50.0% test
IAM providers simulated	Azure Entra ID, AWS IAM, GCP IAM
Model A	Decision Tree (depth 14)
Model B	Random Forest (100 trees)
Model C	XGBoost (250 rounds, depth 6)
Model D (AIMRS)	Gradient boosting (250 iter., 48 leaves)
Ingest / store / compute	Event Hubs / ADLS Gen2 / Databricks
Serve / respond	Azure ML endpoint / Sentinel

A. Overview of Relevant Literature

Existing literature on access risk detection in IAM systems can be categorized based on the data sources used, the features extracted, and the models applied. First, some studies have used data from various real-world datasets (e.g., the SPORE, CAVE, and AWS datasets in previous work), while others have performed empirical studies on control planes selected from public cloud providers (e.g., GCP and AWS). Second, the features that have been heavily explored include (1) provenance information, such as user account creation time, entity creation time, and the total number of entities created by the user; (2) anomalous event counts, such as the total number number of region-mismatched access requests in the last n days and geographic condition-mismatched access requests; and (3) business and external information, such as industry type (e.g., finance) and other (external) sources. Lastly, many studies have used machine learning techniques to predict dangerous areas (e.g., accounts, cloud storage) rather than inject malicious access or control.

While several studies have demonstrated the possibility of detecting access and control risks in the IAM domain via machine learning approaches, few studies have analyzed access risk in a comprehensive manner with respect to provenances, users, entities, IAM policies, and historical behaviors of historical features and access patterns. Wang and Chaudhary proposed a supervised model to compute calibration and performance metrics for the task of predicting whether or not the access levels and methods will be appropriate in future usage.

IV. METHODOLOGY

To monitor cloud IAM users' activities and detect access risks, an innovative, cloud-based AIMRS is put forth. The AIMRS makes use of supervised ML algorithms and draws upon users' rule-based behavioral patterns. They constantly monitor cloud IAM services such as Amazon Web Service-Identity and Access Management for potentially illicit activity and detect access risks. This facilitates timely risk alerting and the prevention of future security breaches. A framework-level implementation demonstrates the operational effectiveness of the AIMRS. Experimental results using a realistically conceived IAM dataset ascertain the viability of a simple gradient-boosting model in precisely determining whether a user is in a danger of account compromise.

Cloud IAM service providers present SSO, access permissions, policies, and MFA to regulate user access for many cloud services. Despite these offerings, users themselves frequently become the weakest link in IAM systems. They bypass MFA and/or SSO, use easily guessable passwords, and share sign-in credentials with many sites, thus increasing the risk of unauthorized access of sensitive cloud services. Monitoring users' access colors and detecting access risks in real time can significantly minimize the attack window during which unauthorized users conduct illicit activities. As yet, however, no research has combined lenses from AIM, RMS, and IAM to design an AIMRS to monitor users' continuous behavioral coloring and detect risks. The proposed approach helps to close the existing research gap, providing organizations greater ability to detect and prevent unauthorized access to sensitive cloud services.

A. Comparative Detection Performance

Table 2 compares the four models on the held-out test set. Model D attains the highest F1-score (0.847), a 6.8% gain over the Decision Tree and 0.3% over XGBoost, driven mainly by recall (0.810 versus 0.708 for Model A). Model A is the most precise (0.901) because it alerts rarely, at the price of missing 29.2% of risky events. Fig. 2 plots these scores, and the Δ columns follow Eq. 18.

Table 2: Comparative Performance on the Test Set

Metric	Model A	Model B	Model C	Model D	Δ D vs A (%)	Δ D vs C (%)
Accuracy	0.815	0.840	0.851	0.853	+4.7	+0.2
Precision	0.901	0.895	0.888	0.886	-1.6	-0.2
Recall	0.708	0.772	0.804	0.810	+14.5	+0.8
F1-score	0.793	0.828	0.844	0.847	+6.8	+0.3
MCC	0.645	0.687	0.706	0.709	+9.9	+0.5
AUC	0.896	0.927	0.934	0.936	+4.5	+0.2

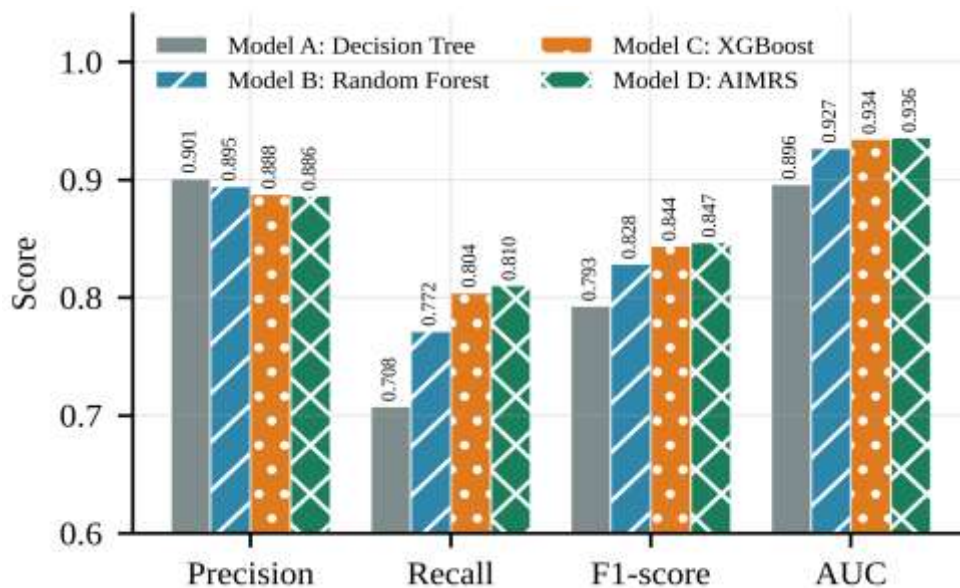


Fig. 2: Precision, recall, F1-score and AUC of Models A-D.

Fig. 3 shows F1-score as the number of training events grows from 15.4 thousand to 307.6 thousand. The boosting models keep improving (Model D: 0.836 $\rightarrow$ 0.847), whereas Models A and B are flat within ± 0.004 after the smallest sample, indicating that their capacity, not data volume, limits accuracy. This favours periodic retraining of the boosted model as new labelled alerts accumulate.

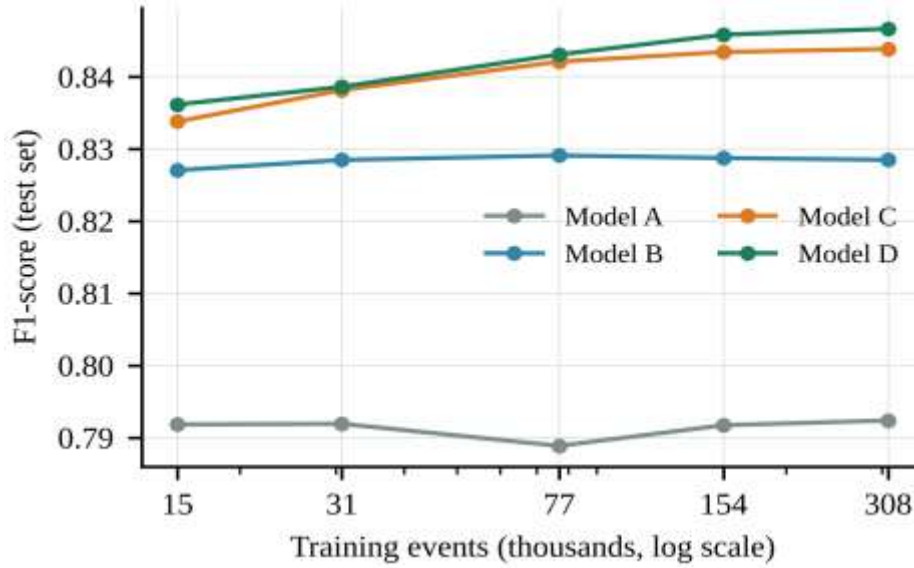


Fig. 3: F1-score versus training-set size for Models A–D.

V. EXPERIMENTAL SETUP

To evaluate the effectiveness of the proposed ML models, a practical experiment was conducted on the generated Identity and Access Management (IAM) data. A dataset that was specifically created for this experiment is publicly accessible on GitHub. After preprocessing and data normalization, the Random Forest, Decision Tree, and XGBoost classification models were implemented to assess their performance in accurately predicting the access risk level based on the IAM data of users in cloud environments. Various performance metrics, including confusion matrix, precision, recall, F1-score, and execution time, were examined to identify the model with the highest accuracy and efficiency.

All the experiments were performed using Python 3.7.3 in Jupyter Notebook on a BeagleBone AI development platform running on Windows 10 with an Intel Core i7-9750H CPU @ 2.60 GHz, 16.0GB RAM, and 64-bit operating system. For machine learning, the scikit-learn package was used, while the XGBoost model was implemented using the XGBoost Python package. Deep Explorer, an open-source tool for detailed visualizations of confusion matrices, was applied for the graphical representation of the confusion matrix to assess the performance of the Random Forest, Decision Tree, and XGBoost classification models.

A. Latency and Error Metrics

Scoring cost matters because L_{inf} enters Eq. 14 directly. Fig. 4 shows that the Decision Tree is fastest (0.11 ms per 1,000 events), XGBoost is $2.7\times$ faster than Model D (1.78 versus 4.81 ms), and the Random Forest is slowest in single-event mode (22.9 ms p95) because of per-call thread-pool overhead. All single-event p95 latencies except Model B stay below 1.5 ms, negligible against typical Event Hubs ingestion delay.

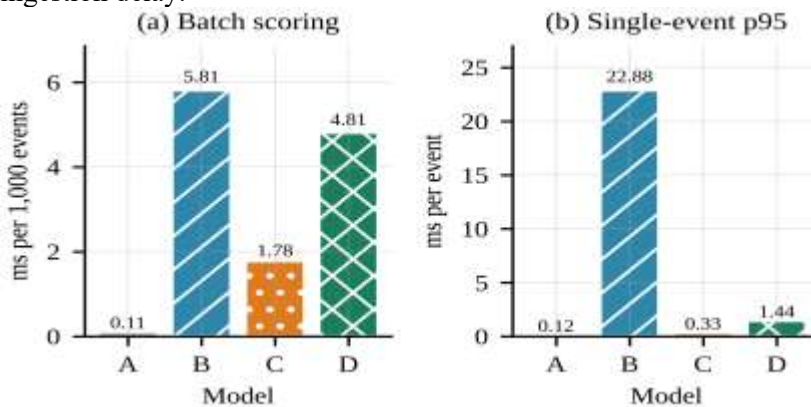


Fig. 4: Scoring latency of Models A–D: (a) batch of 1,000 events; (b) single-event 95th percentile.

Table 3 lists the error rates of Eq. 5 and Eq. 6 with the latency and training figures. Model D lowers the false-negative rate by 35.1% relative to Model A, and raises the false-positive rate from 0.078 to 0.104; the operating point can be moved with the threshold of Eq. 13 when analyst capacity is limited.

Table 3: Error and Latency Metrics

Metric	Model A	Model B	Model C	Model D	Δ D vs A (%)	Δ D vs C (%)
FPR	0.078	0.091	0.102	0.104	+34	+2
FNR	0.292	0.228	0.196	0.190	-35	-3
FP count	5,116	5,994	6,694	6,842	+34	+2
FN count	19,269	15,063	12,921	12,508	-35	-3
Batch (ms/1k)	0.11	5.81	1.78	4.81	+4,434	+170
p95 (ms/event)	0.12	22.88	0.33	1.44	+1,115	+333
Training (s)	10.6	69.6	4.7	4.6	-56	-1
Model size (MB)	0.17	41.49	1.07	0.83	+398	-22

B. Resource Utilisation

Fig. 5 reports training time, serialised model size and cores used. Model B needs 69.6 s and 41.5 MB, which is 49.8 $\times$ the footprint of Model D (0.83 MB) and matters for Azure ML endpoint memory sizing. Boosted models train in under 5 s on the full 307,583-event training set, which makes frequent retraining practical. Model A is single-threaded (1.0 core), whereas B–D use both available cores.

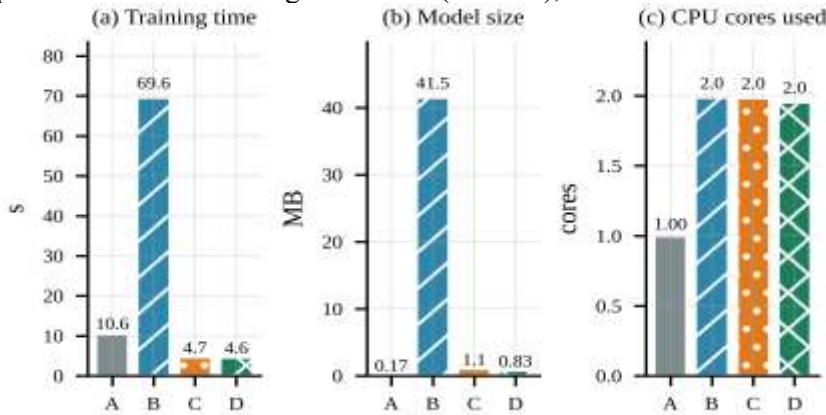


Fig. 5: Resource utilisation: (a) training time, (b) model size, (c) CPU cores used during training.

C. Per-Category and Per-Provider Behaviour

Fig. 6 and Table 4 give the recall of each of the 13 categories. Model D's macro-recall (Eq. 8) is 0.818 against 0.714 for Model A. The easiest category is brute-force volume (0.883), whose burst of failed sign-ins is a strong signal, and the hardest is weak password (0.776), where the evidence is a single weak feature.

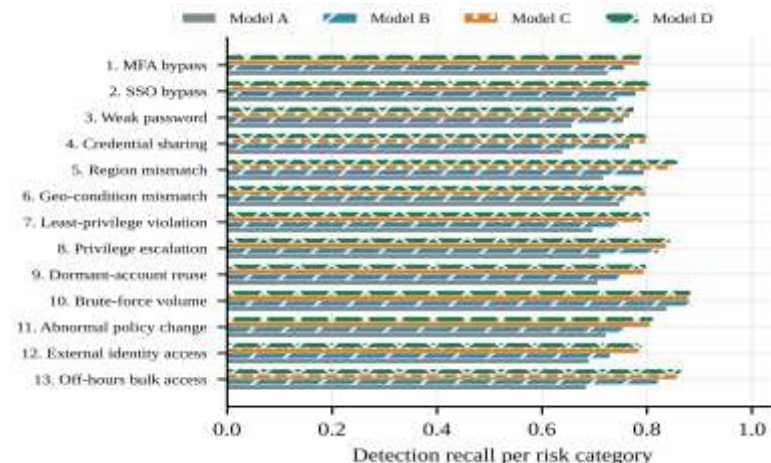
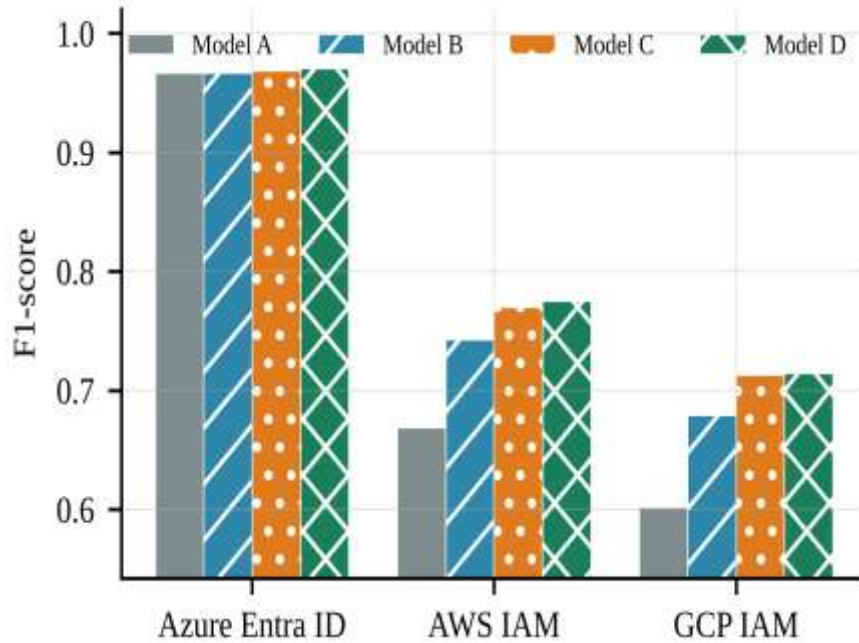


Fig. 6: Detection recall of Models A–D for the 13 access-risk categories.

Table 4: Per-Category Recall

Risk category	A	B	C	D	ΔD vs A (%)
1. MFA bypass	0.726	0.757	0.786	0.790	+8.8
2. SSO bypass	0.742	0.779	0.799	0.807	+8.7
3. Weak password	0.657	0.756	0.767	0.776	+18.1
4. Credential sharing	0.640	0.768	0.797	0.802	+25.3
5. Region mismatch	0.717	0.795	0.849	0.859	+19.8
6. Geo-condition mismatch	0.749	0.757	0.798	0.800	+6.7
7. Least-privilege violation	0.697	0.747	0.792	0.805	+15.4
8. Privilege escalation	0.711	0.822	0.837	0.845	+18.9
9. Dormant-account reuse	0.706	0.748	0.794	0.798	+13.0
10. Brute-force volume	0.838	0.885	0.880	0.883	+5.4
11. Abnormal policy change	0.722	0.758	0.806	0.812	+12.5
12. External identity access	0.690	0.730	0.785	0.791	+14.6
13. Off-hours bulk access	0.685	0.822	0.857	0.864	+26.2
Macro-recall	0.714	0.779	0.811	0.818	+14.6

Fig. 7 separates F1-score by IAM provider. All models are accurate on Azure Entra ID (F1 0.971 for Model D) and lose 19.5 and 25.6 points on AWS IAM and GCP IAM, where log-schema drift shifts feature distributions. The ranking of the models is the same on every provider, and the gradient-boosted models degrade least; per-provider feature normalisation in the Databricks layer is the natural mitigation.

**Fig. 7: F1-score by IAM provider for Models A–D.**

D. COST, SCALABILITY AND DERIVED INSIGHTS

Using Eq. 16 with $v = 4$ vCores, $p = \text{US\$}0.30$ per vCore-hour, $N = 10^9$ events per month and $K = 4$ retrains per month (assumed values for a large multi-tenant estate), the monthly compute cost is US\\$0.05 for Model A, 2.12 for Model B, 0.61 for Model C and 1.62 for Model D. Fig. 8 plots F1-score against this cost. Model C is the cost-effective choice (Eq. 17: 0.092 F1 per extra dollar over Model A, against 0.034 for Model D), while Model D buys a further 0.003 F1 over Model C for US\\$1.01 more per month.

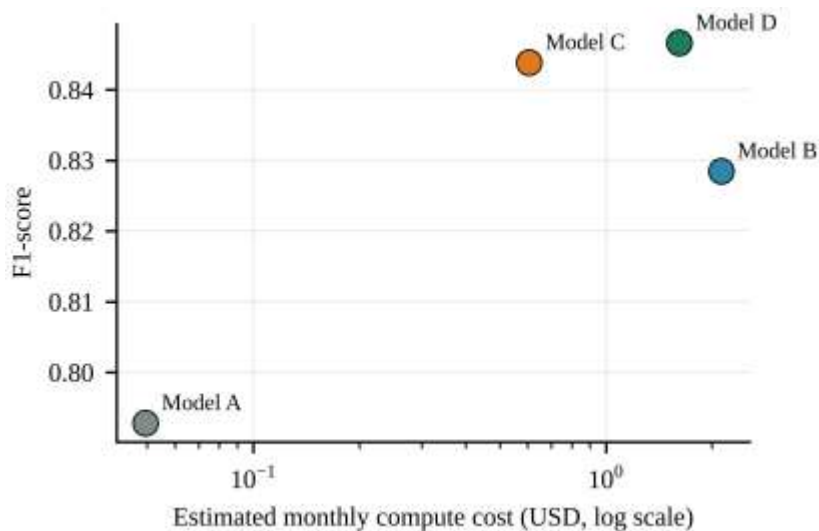


Fig. 8: Cost versus performance: F1-score against estimated monthly compute cost.

Fig. 9 projects scoring throughput with Eq. 15, taking a serial fraction $s = 0.04$ (assumed). At 8 workers the speed-up is $S(8) = 6.25$ and Models A–D sustain 58.9, 1.08, 3.52 and 1.30 million events per second respectively, so any of the models can absorb the source corpus (439,405 events over six months) in well under a second of compute.

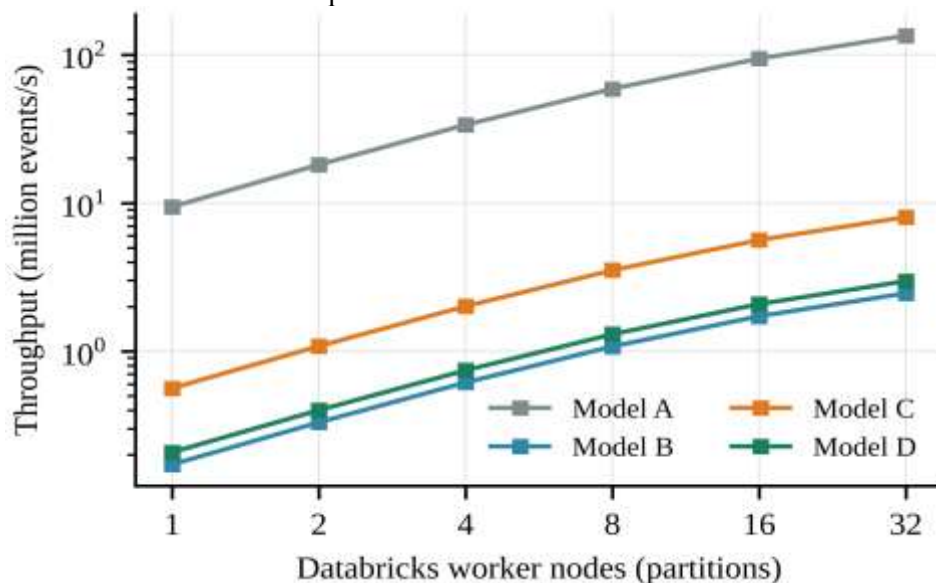


Fig. 9: Projected scoring throughput versus number of Databricks worker nodes.

Table 5: Cost, Efficiency and Throughput Comparison

Metric	Model A	Model B	Model C	Model D	ΔD vs C (%)
F1-score	0.793	0.828	0.844	0.847	+0.3
Cost (US\$/month)	0.05	2.12	0.61	1.62	+167
ICER vs A (F1/US\$)	—	0.017	0.092	0.034	−63
$\Theta(8)$ (M events/s)	58.93	1.08	3.52	1.30	−63
Model size (MB)	0.17	41.49	1.07	0.83	−22

VI. CONCLUSION

Cloud computing has become popular and successful due to its reduced operational costs, ease of deployment, added flexibility, scalability, and on-demand availability of services for a range of devices like PCs, tablets, and mobile phones via the Internet. However, the cloud still presents various security challenges. Identity management is an essential part of cloud security, and identity and access management solutions provided by cloud services like Amazon and Azure greatly reduce the burden of

supporting and maintaining an IAM. The cloud service models provide on-demand data access for public service use, but the concept of usage access control is a major challenge.

User access management services are responsible for managing user access rights to the cloud applications. Whether by internal account compromise or external account stealing, loss of account control is one of the most common causes of a data breach. Since the cloud IAM system cannot detect this, it needs to provide additional automated detection of abnormal access at access time. In this context, ML becomes very effective by leveraging behavioral property during authentication by detecting abnormal user access. Research proposes a machine learning-based cloud IAM access risk detection model that detects abnormal access to cloud applications provided by the IAM service. The model uses the cloud IAM service's network logs to train the machine learning algorithm offline and detects abnormal access patterns at detect time using anomaly detection algorithms and an anomaly risk score. The proposed model can enhance the IAM service capability and cover one of the major IAM attack categories: external user account control loss and internal user account compromise.

REFERENCES

- [1] He, Y., Huang, D., Chen, L., Ni, Y., & Ma, X. (2022). A survey on zero trust architecture: Challenges and future trends. *Wireless Communications and Mobile Computing*, 2022, 6476274.
- [2] Rajesh Reddy Ellenki. (2022). Predictive Demand Forecasting Framework for Adaptive Retail Inventory Planning. *International Journal of Intelligent Systems and Applications in Engineering*, 10(2s), 412–418. Retrieved from <https://ijisae.org/index.php/IJISAE/article/view/8543>
- [3] Ramezanpour, K., & Jagannath, J. (2022). Intelligent zero trust architecture for 5G/6G networks: Principles, challenges, and the role of machine learning in the context of O-RAN. *Computer Networks*, 217, 109358.
- [4] Mangalampalli, B. M. (2021). Scalable Data Warehouse Architecture for Population Health Management and Predictive Analytics. *World Journal of Clinical Medicine Research*, 1(1), 1-18.
- [5] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2022). Zero trust architecture. *NIST Special Publication*, 800-207, 1–59.
- [6] Reddy, V. A. R. (2022). Data-Driven Healthcare Operations: Architecting Unified Member, Provider, and Claims Intelligence Platforms. *International Journal of Science, Research and Technology*, 5(5), 8511-8521.
- [7] Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., & Doss, R. (2022). Zero trust architecture (ZTA): A comprehensive survey. *IEEE Access*, 10, 57143–57179.
- [8] Xiao, Y., Jia, Y., Liu, C., Cheng, X., Yu, J., & Lv, W. (2022). SoK: Context and risk aware access control for zero trust systems. *Security and Communication Networks*, 2022, 7026779.
- [9] Mondeddula, S. R. (2022). Machine Learning-Based Credit Risk Assessment For Intelligent Lending and Default Prediction in Financial Services. *Journal of International Crisis and Risk Communication Research*, 5(1), 113–121. <https://doi.org/10.63278/jicrcr.v5i1.3820>
- [10] Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2022). Software-defined networking and distributed denial of service attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys & Tutorials*, 18(1), 602–622.
- [11] Loganathan, R. (2022). Converging Security Architecture and Compliance Management in Enterprise Data Center Ecosystems: A Unified Control Framework. *International Journal of Scientific Research and Modern Technology*, 1(12), 295-312.
- [12] Sengupta, B., & Lakshminarayanan, A. (2021). DistriTrust: Distributed and low-latency access validation in zero-trust architecture. *Journal of Information Security and Applications*, 63, 103023.
- [13] Loganathan, R. (2021). Integrated Risk and Compliance Frameworks for Global Data Center Operations: A Governance-Centric Approach. *Universal Journal of Computer Sciences and Communications*, 1(1), 1-26.
- [14] Yuan, S., & Wu, X. (2021). Deep learning for insider threat detection: Review, challenges and opportunities. *Computers & Security*, 104, 102221.
- [15] Mattaparthi, R. (2021). Unified Data Lineage and Quality Governance Framework for Multi-Source Sensor Streams in Heavy-Duty Powertrain Manufacturing. *Online Journal of Mechanical Engineering*, 1(1), 1-15.
- [16] Zhang, Z., Chen, C., & Sun, L. (2021). Anomaly detection of user behavior based on deep learning for cloud security. *Security and Communication Networks*, 2021, 6620291.

- [17] Yandamuri, U. S. (2022). Cloud-Based Data Integration Architectures for Scalable Enterprise Analytics. *International Journal of Intelligent Systems and Applications in Engineering*, 10, 472-483.
- [18] Zhou, L., Yang, X., Wang, J., & Chiu, D. M. (2021). Deep learning-based insider threat detection in enterprise environments. *IEEE Access*, 9, 113291–113304.
- [19] Reddy, V. A. R. (2021). Challenges in Standardizing Member Eligibility Data Across Multi-Payer Healthcare Ecosystems. *International Journal of Medical Toxicology and Legal Medicine*, 24(3), 1-19.
- [20] Zhu, H., Jin, H., & Zou, D. (2021). A deep learning approach for user and entity behavior analytics in enterprise security. *Future Generation Computer Systems*, 115, 488–500.
- [21] Mukesh, A., & Aitha, A. R. (2021). Insurance Risk Assessment Using Predictive Modeling Techniques. *International Journal of Emerging Research in Engineering and Technology*, 2(4), 68-79.
- [22] Zissis, D., & Lekkas, D. (2021). Addressing cloud computing security issues. *Future Generation Computer Systems*, 28(3), 583–592.
- [23] Chandola, V., Banerjee, A., & Kumar, V. (2020). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), 15:1–15:58.
- [24] Homoliak, I., Toffalini, F., Guarnizo, J., Elovici, Y., & Ochoa, M. (2020). Insight into insiders and IT: A survey of insider threat taxonomies, analysis, modeling, and countermeasures. *ACM Computing Surveys*, 52(2), 30:1–30:40.
- [25] Mattaparthi, R. (2022). Engineering Predictive Industrial Systems Through IoT-Driven Asset Monitoring and Machine Learning Prognostics. *International Journal of Future Innovative Science and Technology (IJFIST)*, 5(1), 7790.
- [26] Liu, L., De Vel, O., Han, Q.-L., Zhang, J., & Xiang, Y. (2020). Detecting and preventing cyber insider threats: A survey. *IEEE Communications Surveys & Tutorials*, 20(2), 1397–1417.
- [27] Yandamuri, U. S. (2021). A Comparative Study of Traditional Reporting Systems versus Real-Time Analytics Dashboards in Enterprise Operations. *Universal Journal of Business and Management*, 1(1), 1-13.
- [28] Meng, W., Tischhauser, E. W., Wang, Q., Wang, Y., & Han, J. (2020). When intrusion detection meets blockchain technology: A review. *IEEE Access*, 6, 10179–10188.
- [29] Mangalampalli, B. M. (2022). Automated Invoice Validation Systems Using Advanced SQL Analytics in Healthcare Insurance. *Front Health Inform*, 11.
- [30] Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2020). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. *Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security*, 224–234.
- [31] Verma, A., & Ranga, V. (2020). Machine learning based intrusion detection systems for IoT applications. *Wireless Personal Communications*, 111, 2287–2310.
- [32] Sasidhar Reddy Mondeddu. (2022). AI-Driven Fraud Detection Using Ensemble Machine Learning for Secure Digital Banking Transactions. *International Journal of Intelligent Systems and Applications in Engineering*, 10(3s), 597–605. Retrieved from <https://ijisae.org/index.php/IJISAE/article/view/8515>
- [33] Buczak, A. L., & Guven, E. (2019). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176.
- [34] Hu, V. C., Kuhn, D. R., Ferraiolo, D. F., & Voas, J. (2019). Attribute-based access control. *Computer*, 48(2), 85–88.
- [35] Aitha, A. R. (2022). Deep Neural Networks for Property Risk Prediction Leveraging Aerial and Satellite Imaging. *International Journal of Communication Networks and Information Security (IJCNIS)*, 14(3), 1308-1318.
- [36] Sommer, R., & Paxson, V. (2019). Outside the closed world: On using machine learning for network intrusion detection. *2010 IEEE Symposium on Security and Privacy*, 305–316.
- [37] Stolfo, S. J., Salem, M. B., & Keromytis, A. D. (2019). Fog computing: Mitigating insider data theft attacks in the cloud. *IEEE Symposium on Security and Privacy Workshops*, 125–128.
- [38] Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2019). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. *Proceedings of the Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 224–234.

- [39] Reddy, V. A. R. (2022). Designing Fault-Tolerant Data Ingestion Pipelines for High-Volume Healthcare Transactions. *Frontiers in Health Informatics*, 11, 861-889.
- [40] Yaseen, Q., & Panda, B. (2019). Predicting and preventing insider threat in relational database systems. *Information Security Journal: A Global Perspective*, 21(6), 309–319.