International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Enhancing Cloud Security Using Multi-Factor Authentication

PRATYUSH KUMAR
Department of Computer Application
BBD University, Lucknow - 226 028, India
pratyush.kumar73@gmail.com

SHIVANI GUPTA
Department of Computer Application
BBD University, Lucknow - 226 028, India
shivanigupta41002@gmail.com

RAJ VERMA
Department of Computer Application
BBD University, Lucknow - 226 028, India
rajverma97936@gmail.com

AASTHA SINGH
Assistant Professor
Department of Computer Application,
BBD University, Lucknow - 226 028, India
Aasthasingh158@gmail.com

***Abstract*—Cloud platforms keep valuable data behind a single login, so stolen or guessed credentials remain one of the easiest ways in. Static multi-factor authentication (MFA) closes much of that gap, but it asks every user for an extra factor on every login, and that friction tempts organisations to weaken their policies. This paper presents RA-MFA, a risk-adaptive MFA framework for cloud applications. After the password check, a lightweight logistic-regression model scores each login attempt from eight contextual signals such as IP reputation, travel velocity, device novelty and behavioural deviation. The score selects one of three responses: allow with a device token, a push/TOTP challenge, or a strong challenge (TOTP plus biometric or a FIDO2 key). We evaluate the framework on a synthetic dataset of 20,000 login events that contains 1,577 account-takeover attempts, and compare it with password-only, static TOTP and rule-based adaptive baselines. On the held-out test set RA-MFA blocked 95.8% of attacks, which is comparable to static MFA (94.7%), while it challenged only 22.9% of legitimate logins and reduced the average added login time from 11.0 s to 2.2 s. We also discuss the limits of simulated evaluation and describe how the framework can be validated on real traffic.**

***Keywords*— adaptive authentication, cloud security, multi-factor authentication, risk scoring, zero trust.**

I. INTRODUCTION

Moving workloads to the cloud shifts the security perimeter from the office network to the identity of the person logging in. A tenant's storage buckets, virtual machines and management consoles are all reachable from the public Internet, and the only thing standing between an attacker and that data is often a username and a password. Password reuse, phishing kits and large credential dumps make this a weak barrier, which is why cloud providers and regulators now treat MFA as a baseline control [1], [2].

MFA is not free, however. A second factor requested at every login adds seconds to a task that users repeat many times a day, and it generates support tickets for lost phones and expired codes. Practical deployments therefore tend to drift towards exemptions, long "remember this device" windows or weaker factors such as SMS. At the other extreme, a policy that is too lax lets an attacker with a leaked password walk straight in. What is needed is a mechanism that asks for more proof only when the situation looks risky.

Risk-adaptive authentication aims at exactly that. Recent work has proposed adaptive designs [1], authenticator-selection logic combined with intrusion detection [2] and machine-learning-assisted MFA for financial services [3]. Most of these, however, either describe the idea without a reproducible scoring model or evaluate individual checks without measuring how many attacks actually get through and how much friction honest users face.

This paper makes the following contributions:

1) RA-MFA, a three-step cloud login framework in which a logistic-regression risk score decides whether the user receives no challenge, a medium-strength challenge or a strong challenge.

2) A reproducible evaluation set-up with a synthetic login dataset, four authentication policies and metrics that capture both security (attack success rate) and usability (prompt rate and added login time).

3) An analysis of threshold trade-offs, feature influence and the mitigation of common cloud-authentication threats, together with an honest account of what a simulation can and cannot show.

The remainder of the paper is organised as follows. Section II reviews related work, Section III gives background on authentication factors and the threat model, Section IV describes RA-MFA, Section V explains the experimental set-up, Section VI reports results, Section VII discusses security and limitations, and Section VIII concludes.

II. RELATED WORK

Prakash and Singh [1] describe a multi-factor framework in which a risk score built from location, device, login time and history chooses between direct access, a push or SMS challenge, and an OTP-plus-fingerprint challenge. The design is easy to follow, but it is presented as pseudocode and a flowchart; the weights and thresholds are left open, so the scoring step cannot be reproduced. Pell Reddy [2] proposes an authenticator-strategy sorter that rotates among methods according to a user's recent behaviour, adds intrusion checks on IP address and geofence, and protects stored credentials with a ciphertext-policy attribute-based homomorphic scheme. The evaluation reports execution time and false-positive and false-negative rates for 50 to 1,000 users, yet some of the proposed factors, such as IMEI numbers and thermal-image recognition, are hard to collect from an ordinary browser or a modern phone.

Aburbeian and Fernández-Veiga [3] combine MFA with machine-learning classifiers to secure online financial transactions and stress the trade-off between customer reluctance and protection. Okeke and Orimadike [4] study application-based MFA for cloud communications; as summarised in [1], they also point out deployment problems such as client software that stops working on newer operating systems and access failures when users connect through VPNs. On the protocol side, Han *et al.* [5] design a lightweight multi-factor key-agreement protocol for industrial IoT using symmetric primitives, hashing and XOR, and Shukla and Patel [6] present a provably secure ECC-based multi-factor protocol for multi-server clouds. Konwar *et al.* [7] focus on a new OTP-generation algorithm for two-factor authentication of cloud applications.

Table I summarises how these studies differ from the present work. The gap we address is the combination of three properties in one design: a transparent, trainable scoring model, a clear mapping from score to factor strength, and an

evaluation that reports attack success and user friction against several baselines.

TABLE I
COMPARISON WITH RECENT RELATED WORK

Study	Main idea	Open issue addressed here
[1] 2025	Risk score selects push or OTP+fingerprint	Weights, thresholds and metrics not specified
[2] 2024	Authenticator sorter, IDS checks, CP-ABHE	Hard-to-collect factors; no attack-success baseline
[3] 2024	MFA with ML for financial transactions	Targets payments, not cloud login policy
[5] 2024	Lightweight IIoT key agreement	Protocol-level; no contextual risk
[6] 2024	ECC-based multi-server MFA protocol	Static factors for every login
Ours	**Logistic risk model + three-tier policy**	**Open, reproducible, friction measured**

III. BACKGROUND AND THREAT MODEL

A. Authentication Factors

MFA combines factors from different categories: something the user knows (password, PIN), something the user has (phone, hardware key) and something the user is (fingerprint, face). Contextual signals such as location and time are sometimes treated as additional factors. Each factor has its own strengths and weaknesses, summarised in Table II. The key point for design is that factors differ greatly in phishing resistance and in how much effort they demand from the user, so a policy that always applies the same factor wastes either security or convenience.

TABLE II
COMMON AUTHENTICATION FACTORS

Factor	Strength	Typical weakness
Password / PIN	Cheap, universal	Phishing, reuse, credential stuffing
SMS / e-mail OTP	No extra app needed	SIM swap, interception, relay phishing
TOTP app	Works offline, standard	Real-time phishing relay, seed theft
Push approval	One tap	Prompt bombing (MFA fatigue)
Biometric	Convenient, hard to share	Spoofing; template cannot be changed
FIDO2 / hardware key	Phishing-resistant	Cost; account recovery logistics

B. Threat Model

We assume an attacker who already holds a valid username and password for a cloud account, for example from a breach, a phishing page or credential stuffing, and who tries to log in from a different environment than the legitimate user. Some attackers are careful: they route traffic through residential proxies and try to resemble the victim. We further assume that an attacker can sometimes defeat a medium-strength second factor through phishing relay or push-prompt fatigue, but only rarely defeats a phishing-resistant factor. The attacker does not control the authentication server, and the user's registered device is not compromised.

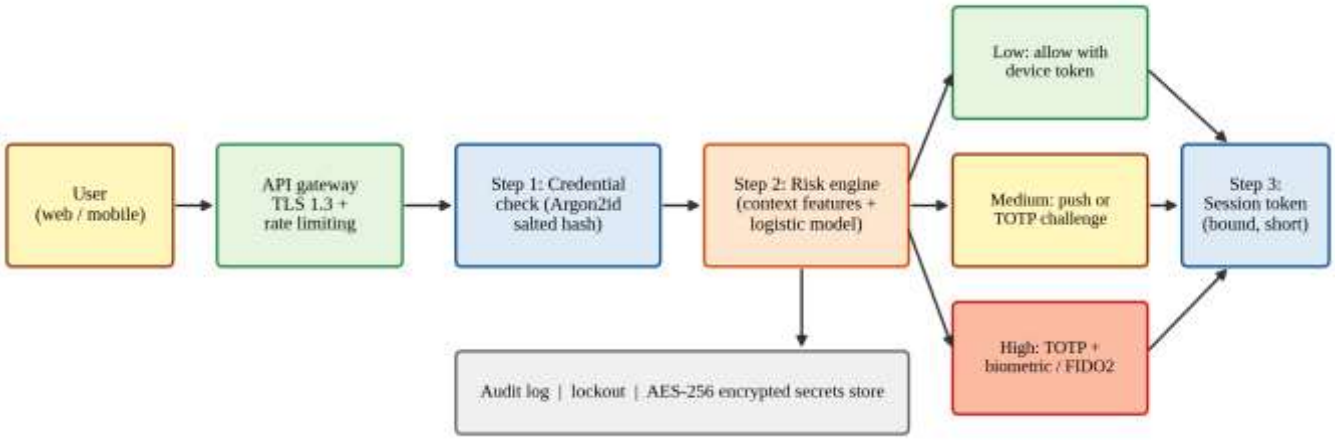


Fig. 1. Overall architecture of the proposed RA-MFA framework.

IV. THE PROPOSED RA-MFA FRAMEWORK

A. Overview

RA-MFA sits between the cloud application and its identity store (Fig. 1). A login passes through three steps. Step 1 verifies the password. Step 2 gathers context and computes a risk score. Step 3 issues a session token once the factor demanded by the score has been satisfied. All traffic is carried over TLS 1.3, secrets such as TOTP seeds are stored encrypted with AES-256, and every decision is written to an audit log. Fig. 2 shows the decision flow.

B. Credential Stage

Passwords are stored as salted Argon2id hashes, a slow memory-hard function that makes offline cracking expensive even if the user table leaks. The gateway applies per-account and per-IP rate limits. A wrong password increments a failed-attempt counter that later feeds the risk model, and repeated failures lock the account temporarily.

C. Contextual Features

When the password is correct, the platform collects eight signals that are cheap to compute at login time (Table III). None of them requires special hardware, and all are available to a standard web or mobile client.

TABLE III
FEATURES USED BY THE RISK ENGINE

Feature	Description	Range
geo_dist	Distance from user's usual location (scaled)	0–6
ip_reputation	Blocklist / abuse score of source IP	0–1
new_device	Device fingerprint unseen before	0/1
hour_dev	Hours away from usual login time	0–12
failed_att	Failed attempts in the last hour	0–10
travel_vel	Impossible-travel score vs. last login	0–1
anon_net	Tor / open proxy / hosting range	0/1
behav_dev	Typing-rhythm deviation from profile	0–1

D. Risk Model

Let x be the standardised feature vector of a login attempt. The risk score is the output of a logistic function:

$$r = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b) = 1 / (1 + e^{-(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)}) \quad (1)$$

where w and b are learned from labelled historical logins using L2-regularised logistic regression with class weighting, because attacks are rare. We chose logistic regression deliberately: it is fast enough to run inside the login request, its coefficients can be inspected by a security analyst, and it can be re-trained regularly from the audit log. Fig. 3 shows the learned coefficients on our data.

E. Adaptive Policy

Two thresholds τ_1 and τ_2 divide the score range into three tiers:

$$action(r) = \text{ALLOW if } r < \tau_1;\ \text{STEP-UP if } \tau_1 \le r < \tau_2;\ \text{STRONG if } r \ge \tau_2 \quad (2)$$

In the ALLOW tier the user receives a device-bound token without a second factor. In the STEP-UP tier the user approves a push notification or enters a TOTP code; push prompts are rate-limited and use number matching to blunt fatigue attacks. In the STRONG tier the user must present a TOTP code together with a biometric check, or a FIDO2 security key, which is bound to the site origin and therefore resists phishing relay. We use $\tau_1 = 0.15$ and $\tau_2 = 0.60$; Section VI studies the sensitivity of τ_1.

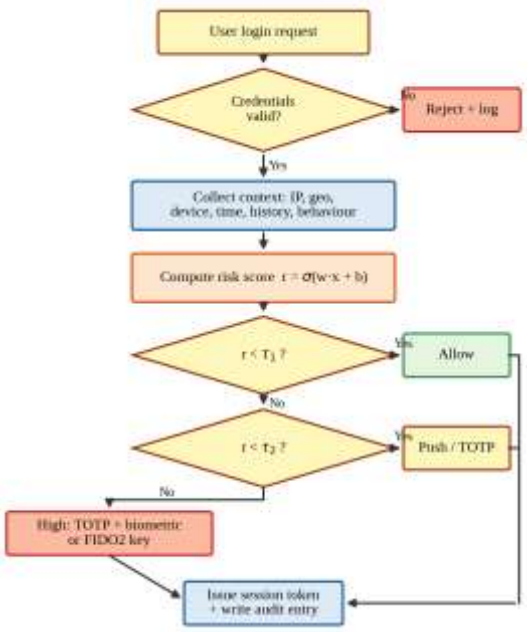


Fig. 2. Login decision flow of RA-MFA.

F. Login Algorithm

Algorithm 1 gives the complete login procedure.

```
Algorithm 1: RA-MFA login
Input : username u, password p, request context c
Output: GRANTED or DENIED
1  if not VerifyArgon2id(u, p): Log(u,c); return DENIED
2  x <- ExtractFeatures(u, c)
3  r <- Sigmoid(w . Standardise(x) + b)
4  if r < T1:  level <- NONE
5  elif r < T2: level <- STEPUP (push/TOTP)
6  else:        level <- STRONG (TOTP+bio / FIDO2)
7  if level != NONE and not Challenge(u, level):
8      Log(u, c, r, FAIL); LockoutIfNeeded(u)
9      return DENIED
10 t <- IssueSessionToken(u, deviceID, ttl)
11 Log(u, c, r, level, OK); return GRANTED
```

G. Protecting Data and Sessions

Session tokens are short-lived and bound to the device identifier that passed the check, so a stolen cookie is of little use elsewhere. TOTP seeds and biometric templates are kept encrypted at rest and never leave the identity store. Biometric matching is done on the user's device where possible, and only a signed assertion is sent to the server. Behavioural features such as typing rhythm are stored as numeric profiles instead of raw keystrokes to reduce privacy exposure.

V. EXPERIMENTAL SET-UP

A. Dataset

No public dataset provides labelled cloud-login attempts with all of the contextual signals we need, and real login logs are confidential. We therefore generated a synthetic dataset of 20,000 login events with a seeded random generator so that every number in this paper can be reproduced. Eight percent of events (1,577) are account-takeover attempts using valid credentials; the rest are genuine logins. To avoid an unrealistically easy problem we added two kinds of overlap: about a quarter of the attackers are "stealthy" and imitate normal behaviour (similar location, few failures), and about 10% of genuine logins occur in unusual conditions such as travel or a new phone. The data were split 70/30 in a stratified way, giving 14,000 training events and a test set of 6,000 events with 473 attacks. The feature distributions are assumptions chosen to be plausible, not measurements of a real service, so the absolute numbers below should be read as an illustration of relative behaviour.

B. Compared Policies

We compare four policies on the same test events: (i) *password only*; (ii) *static MFA*, which always asks for a TOTP code; (iii) *rule-based adaptive*, which uses a fixed hand-weighted sum of the normalised features with the same three-tier structure (thresholds 0.30 and 0.55); and (iv) the proposed *RA-MFA* with the learned logistic score.

C. Challenge Model and Metrics

Because we cannot run real users, the outcome of each challenge is drawn from the assumed probabilities in Table IV. They encode the idea that medium-strength factors can occasionally be bypassed through phishing relay or prompt fatigue, while a TOTP-plus-biometric or FIDO2 challenge is rarely bypassed but is slightly more likely to fail for honest users. The metrics are: attack success rate (attackers who obtain a session), legitimate rejection rate, prompt rate (share of genuine logins that receive a challenge) and mean added login time for genuine users.

TABLE IV
ASSUMED CHALLENGE PARAMETERS

Challenge	Attacker passes	Genuine passes	User time (s)
None	100%	100%	0
Push	8%	98.5%	6
TOTP	6%	97.5%	11
TOTP + bio / FIDO2	1%	95%	17

VI. RESULTS AND DISCUSSION

A. Overall Comparison

Table V reports the outcome on the 6,000 test events. Password-only authentication lets every attacker in. Static TOTP stops most attackers (attack success 5.3%) but challenges everyone, costing 11 s per login and rejecting 2.5% of genuine users at the first attempt. The rule-based adaptive policy is much cheaper for users but lets 15.4% of attackers through, because fixed weights cannot separate stealthy attackers from users in unusual contexts. RA-MFA reached the lowest attack success rate (4.2%) while challenging 22.9% of genuine logins and adding 2.2 s on average, which is roughly 80% less user effort than static MFA.

TABLE V
PERFORMANCE ON THE TEST SET (N = 6,000)

Policy	Attack succ. (%)	Attacks blocked (%)	Genuine rejected (%)	Prompt rate (%)	Added time (s)
Password only	100.0	0.0	0.00	0.0	0.0
Static MFA (TOTP)	5.3	94.7	2.51	100.0	11.0
Rule-based adaptive	15.4	84.6	0.11	10.4	0.7
Proposed RA-MFA	**4.2**	**95.8**	**0.45**	**22.9**	**2.2**

A caution is needed when reading the 1.1-point gap between static MFA and RA-MFA. With 473 attackers in the test set the standard error of an attack success rate near 5% is about one percentage point, so we regard the two as equivalent in security. The real advantage of RA-MFA is that it reaches this protection with a fraction of the friction.

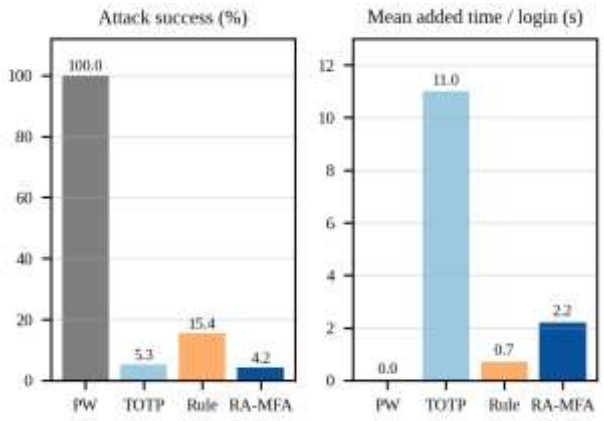


Fig. 3. Attack success rate and mean added login time for the four policies (PW = password only).

B. Risk Score Quality

The learned model separates the two populations well, with an area under the ROC curve of 0.969 compared with 0.963 for the hand-weighted score (Fig. 4). The AUC gap is small, but it translates into a clearer practical difference at the chosen operating points, as the lower attack success rate in Table V shows. Fig. 5 shows the score distributions together with the two thresholds.

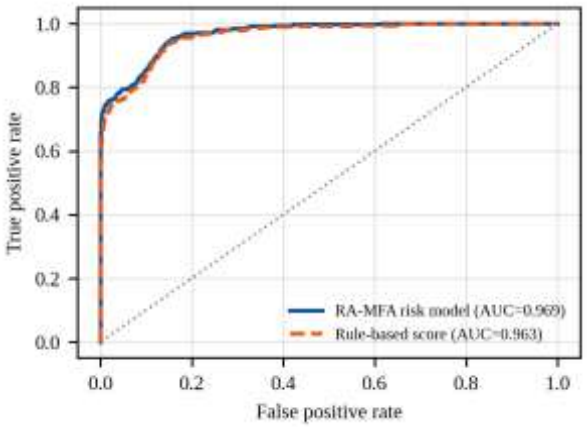


Fig. 4. ROC curves of the learned and the rule-based risk scores.

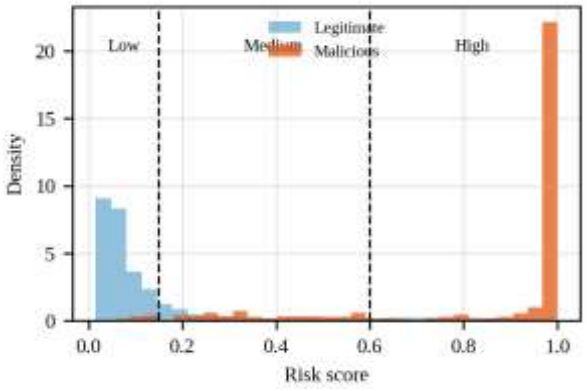


Fig. 5. Distribution of risk scores for genuine and malicious logins; dashed lines mark τ_1 and τ_2.

The tier breakdown in Table VI confirms the intended behaviour. Of the 5,527 genuine logins in the test set, 77.1% fall in the low tier, 15.4% in the medium tier and 7.5% in the high tier. Of the 473 attacks, 81.2% reach the high tier, 16.1% the medium tier and only 2.7% (13 attempts) fall in the low tier and enter without a challenge.

TABLE VI
RISK-TIER OUTCOME OF RA-MFA ON THE TEST SET

Tier	Genuine logins	Attacks	Challenge
Low ($r < 0.15$)	4,260	13	None
Medium	850	76	Push / TOTP
High ($r \geq 0.60$)	417	384	TOTP+bio / FIDO2
Total	5,527	473	—

C. Influence of Features

Travel velocity (coefficient 1.12) and failed attempts (0.89) carry the most weight, followed by behavioural deviation (0.52), anonymising networks (0.40) and IP reputation (0.38). New-device status and hour deviation contribute less. Raw geographic distance receives an almost zero weight (Fig. 6) because it is strongly correlated with travel velocity, so the model needs only one of the two. These coefficients are a property of our synthetic distributions and will differ on real traffic, but the ability to inspect them is a practical advantage over opaque models.

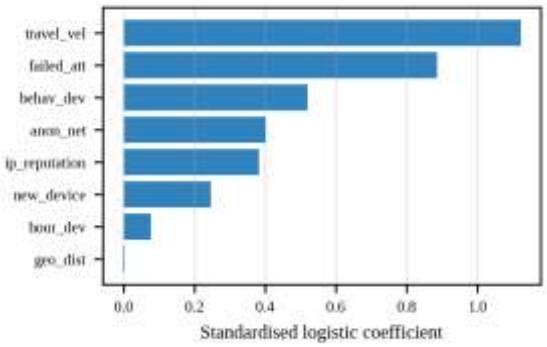


Fig. 6. Standardised logistic coefficients of the eight features.

D. Threshold Sensitivity

Fig. 7 varies the low-risk threshold τ_1 with τ_2 fixed. Lowering τ_1 to 0.05 blocks 98.1% of attacks but challenges 67.8% of genuine logins, which approaches the friction of static MFA. Raising it to 0.35 reduces prompts to 12.0% but lets attack success grow to 11.6%. The value 0.15 is a reasonable compromise for this data, and an operator can move along the curve according to the sensitivity of the application, for example by choosing a lower τ_1 for an administrator console than for a read-only dashboard.

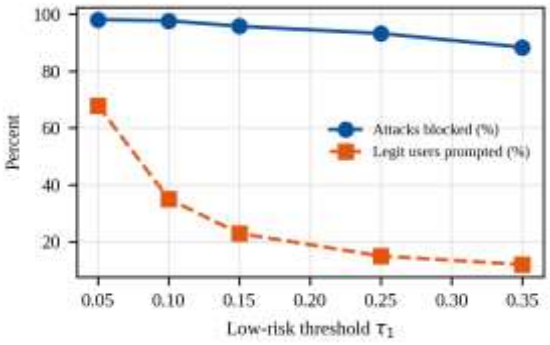


Fig. 7. Effect of the low-risk threshold τ_1 on security and user friction.

VII. SECURITY ANALYSIS AND LIMITATIONS

A. Threat Mitigation

Table VII relates common cloud-authentication threats to the mechanisms in RA-MFA. These mappings come from the design of the framework; only the contextual-risk effect on account takeover is measured in our simulation.

TABLE VII
THREATS AND CORRESPONDING RA-MFA MECHANISMS

Threat	Mitigation in RA-MFA
Credential stuffing	Velocity, IP-reputation and anonymiser signals raise the tier; strong factor demanded
Brute force / dictionary	Argon2id hashing, rate limits, lockout, failed-attempt feature
Phishing / OTP relay	FIDO2 origin-bound key in the high tier
MFA fatigue	Push rate limit, number matching, escalation to strong tier
Replay	Single-use challenges and short token lifetime
Man-in-the-middle	TLS 1.3 for all channels
Session theft	Token bound to device, short TTL
Database leak	Salted hashes; AES-256 for seeds and templates

B. Limitations

The most important limitation is that the evaluation is simulated. The dataset and the bypass probabilities in Table IV are assumptions, and real attackers adapt: an attacker who learns that a particular country or device type is treated as low-risk can try to imitate it. The 2.7% of attacks that enter without a challenge show that the low tier is the weak point, and it should be combined with continuous session monitoring. Behavioural features raise privacy questions, so an operator must obtain consent and follow data-protection regulation. Finally, a risk model needs retraining as user behaviour changes, and a mistaken high score can lock out a genuine user at an inconvenient moment, so a well-designed recovery path is essential.

C. Computational Overhead

Scoring a login requires one dot product over eight standardised features followed by a sigmoid, a constant-time operation that adds negligible server cost compared with password hashing. The dominant latency in RA-MFA is therefore the human response to a challenge, which is why we measure user time rather than CPU time. Feature extraction, in particular IP-reputation lookup and device-fingerprint matching, is the part that depends on external services and should be cached to keep the login path fast.

D. Practical Deployment

RA-MFA can be integrated with identity platforms through a pre-authentication hook that returns the required assurance level. Running first in "shadow mode", where the score is recorded but not enforced, lets an organisation collect real labelled data, calibrate thresholds and measure false challenges before switching the policy on.

VIII. CONCLUSION AND FUTURE WORK

This paper presented RA-MFA, a risk-adaptive multi-factor authentication framework that uses a transparent logistic-regression score to decide how much proof a cloud login requires. In a controlled simulation with 20,000 synthetic login events, it matched the protection of static TOTP (4.2% versus 5.3% attack success, a difference within sampling error) while challenging fewer than a quarter of genuine logins and cutting added login time by about 80%. It also clearly outperformed a hand-weighted rule-based baseline, which let 15.4% of attacks through.

Future work will proceed along four lines: validating the framework on anonymised real login logs in shadow mode; replacing logistic regression with gradient-boosted or sequence models and comparing their interpretability; adding continuous authentication during the session; and studying the privacy cost of behavioural features, possibly with on-device scoring.

REFERENCES

[1] M. P. D. Prakash and A. Singh, "Maximizing cloud security: Synergistic approaches to multi-factor authentication deployment," in *Proc. Int. Conf. Recent Advancement and Modernization in Sustainable Intelligent Technologies & Applications (RAMSITA-2025)*, Adv. Intell. Syst. Res., vol. 192, 2025, pp. 945–955, doi: 10.2991/978-94-6463-716-8_70.

[2] R. R. Pell Reddy, "Empowering cloud security: Pioneering an interactive multi-factor authentication framework for cloud user verification," *Int. J. Comput. Trends Technol.*, vol. 72, no. 6, pp. 168–178, Jun. 2024, doi: 10.14445/22312803/IJCTT-V72I6P122.

[3] A. H. M. Aburbeian and M. Fernández-Veiga, "Secure internet financial transactions: A framework integrating multi-factor authentication and machine learning," *AI*, vol. 5, no. 1, pp. 177–194, 2024, doi: 10.3390/ai5010010.

[4] R. O. Okeke and S. O. Orimadike, "Enhanced cloud computing security using application-based multi-factor authentication (MFA) for communication systems," *Eur. J. Electr. Eng. Comput. Sci.*, vol. 8, no. 2, pp. 1–8, 2024, doi: 10.24018/ejece.2024.8.2.593.

[5] Y. Han, H. Guo, J. Liu, B. B. Ehui, Y. Wu, and S. Li, "An enhanced multifactor authentication and key agreement protocol in industrial Internet of Things," *IEEE Internet Things J.*, vol. 11, pp. 16243–16254, 2024, doi: 10.1109/JIOT.2024.3355228.

[6] S. Shukla and S. J. Patel, "A design of provably secure multi-factor ECC-based authentication protocol in multi-server cloud architecture," *Cluster Comput.*, vol. 27, pp. 1559–1580, 2024.

[7] R. Konwar, D. Jha, R. Agrawal, R. Purkayastha, and I. Banerjee, "A two-factor authentication mechanism using a novel OTP generation algorithm for cloud applications," in *Proc. 14th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, 2024, pp. 245–250, doi: 10.1109/Confluence60223.2024.10463309.