International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

ATOShield: Real-Time Account Takeover Detection Using Unsupervised Machine Learning and Behavioral Analytics

Akula Nehal[1], Ambati Bhanu Prasad[2], G. Charitha[3], Dr. R.R.S. Ravi Kumar[4]
[1,2,3]Department of Artificial Intelligence and Data Science, Vidya Jyothi Institute of Technology, Hyderabad, Telangana, India
[4]Assistant Professor, Department of Artificial Intelligence and Data Science, Vidya Jyothi Institute of Technology, Hyderabad, Telangana, India

Abstract: Account Takeover (ATO) attacks have emerged as one of the most critical threats in the contemporary cybersecurity landscape, wherein malicious actors exploit stolen credentials, session hijacking, and behavioral impersonation to gain unauthorized access to legitimate user accounts. Traditional static authentication mechanisms—such as password verification and rule-based IP blocking—are fundamentally reactive and incapable of detecting sophisticated adversaries who possess valid credentials. This paper presents ATOShield, a real-time ATO detection framework that applies the Isolation Forest unsupervised machine learning algorithm to continuously analyze six behavioral signals extracted from login events: login hour, IP address change, device fingerprint change, login frequency, geographic location change (geo-velocity), and failed authentication attempts. For each login event, the system computes a continuous risk score on a normalized scale of 0 to 100 and classifies it into four actionable tiers—LOW, MEDIUM, HIGH, and CRITICAL, triggering proportional automated responses ranging from soft multi-factor authentication challenges to complete session termination. A fully functional React.js web dashboard provides real-time visualization through animated risk gauges, behavioral radar charts, hourly anomaly trend charts, and a live alert panel. An interactive simulation engine enables direct manipulation of all six behavioral features to transparently demonstrate detection logic. Experimental evaluation demonstrates sub-100 ms inference latency, 99.7% precision, and 94.2% recall on benchmark behavioral anomaly datasets, establishing ATOShield as a scalable, label-free, and proactive defense against modern account takeover attacks.

Keywords— account takeover detection; behavioral analytics; Isolation Forest; unsupervised machine learning; real-time cybersecurity; anomaly detection; risk scoring; geo-velocity; credential stuffing; zero-day attack.

I. INTRODUCTION

The proliferation of internet-connected services and cloud-based platforms has established digital accounts as the primary gateway to personal data, financial assets, and enterprise resources. Account Takeover (ATO) attacks—wherein an adversary gains unauthorized access to a legitimate user account—have become one of the most prevalent and economically damaging categories of cybercrime. According to industry security reports, ATO attacks impose an estimated USD 16.9 billion in annual losses on global organizations [1]. Unlike brute-force intrusions that trigger conventional security alarms, modern ATO attacks frequently leverage stolen credentials procured from dark web marketplaces, phishing campaigns, or large-scale data breaches. Consequently, the attacker presents a syntactically valid password and bypasses first-factor authentication without triggering any rule-based alert.

Conventional defensive mechanisms including IP blacklisting, CAPTCHA challenges, rate limiting, and simple login-attempt counters are insufficient against this threat model. Contemporary attackers employ credential-stuffing automation frameworks, residential proxy networks, and bot orchestration platforms engineered to closely emulate legitimate human behavior [2]. The critical analytical gap lies not in password validation but in behavioral context: “when” a user authenticates, “from where,” “on which device,” and “at what frequency” collectively constitute a behavioral fingerprint that is substantially more difficult for an adversary to replicate than a stolen credential.

DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

Unsupervised machine learning—specifically the Isolation Forest (iForest) algorithm [3]—offers a compelling solution to this challenge. By training exclusively on the statistical distribution of normal login behavior, iForest can identify deviations without requiring any labeled fraud examples. This property is critical in operational security contexts where fraudulent sessions are rare, labels are delayed by days or weeks, and attack methodologies evolve faster than labeling pipelines can adapt.

This paper presents ATOShield, a real-time ATO detection framework that integrates an Isolation Forest scoring engine with a comprehensive analyst-facing web dashboard. The primary contributions of this work are as follows:

1. A six-feature behavioral analytics pipeline that extracts and normalizes login-event signals for real-time ML inference;
2. An Isolation Forest scoring model trained on normal login behavior that achieves 99.7% precision and 94.2% recall at sub-100 ms inference latency;
3. A four-tier risk classification scheme (LOW, MEDIUM, HIGH, CRITICAL) with proportional automated response actions;
4. A production-grade React.js dashboard delivering real-time visualizations, a live alert panel, and a fully interactive simulation engine; and
5. A modular, containerized architecture deployable via Docker on commodity infrastructure without proprietary licensing.

The remainder of this paper is organized as follows. Section II reviews related work on behavioral anomaly detection and ATO defenses. Section III describes the system methodology and pipeline design. Section IV details implementation specifics and the experimental setup. Section V presents results and comparative analysis. Section VI concludes the paper and identifies future research directions.

II. RELATED WORK

A. Early Signature-Based and Rule-Based Approaches

The earliest account-security mechanisms relied on signature-based intrusion detection systems (IDS) such as SNORT [4], which maintained databases of known attack patterns and flagged exact matches. While effective against catalogued threats, these systems were entirely blind to novel attack vectors. In the authentication domain, early controls including IP-based rate limiting, CAPTCHA challenges, and simple failed-attempt counters were rapidly circumvented by attackers deploying CAPTCHA-solving services, residential proxy rotation, and low-rate credential stuffing distributed over extended time windows [5]. Feature-engineered classifiers—extracting time-of-day histograms, geographic distance metrics, and IP reputation scores and feeding them into Naïve Bayes or Logistic Regression models—showed initial promise on controlled datasets but exhibited poor cross-domain generalization as attack methods evolved.

B. Supervised Machine Learning for Behavioral Authentication

The availability of large-scale enterprise login-event datasets motivated the application of supervised learning to ATO detection. Models trained on labeled corpora of normal and fraudulent login events—including Support Vector Machines (SVMs), Random Forests, and Gradient Boosting classifiers—demonstrated improved detection rates by incorporating richer behavioral features such as keystroke dynamics, mouse movement trajectories, and touch-pressure profiles [6]. However, these approaches share a fundamental operational limitation: labeled fraud data is rare (typical fraud prevalence of 0.1–1%), temporally delayed (sessions may not be identified as fraudulent for days or weeks post-occurrence), and rapidly outdated as adversaries adapt their behavioral mimicry. Class imbalance further degrades classifier calibration, inflating false-negative rates for the minority fraud class.

C. Unsupervised Anomaly Detection

Unsupervised methods that learn the statistical structure of normal behavior—and flag deviations therefrom—have emerged as the preferred paradigm for production ATO detection. Liu et al. [3]

DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

introduced the Isolation Forest (iForest) algorithm, which constructs an ensemble of random binary partition trees. Anomalous data points—residing in low-density regions of the feature space—are isolated in fewer partitions, yielding a shorter average path length and a correspondingly higher anomaly score. iForest demonstrated superior performance relative to One-Class SVM [7] and Local Outlier Factor (LOF) [8] on high-dimensional behavioral datasets, particularly with respect to computational efficiency and robustness to the curse of dimensionality.

Goldstein and Uchida [9] conducted a comprehensive comparative evaluation of unsupervised anomaly detection algorithms, confirming that iForest consistently outperforms density-based and distance-based methods across diverse multivariate datasets at linear time complexity O(n log n). Deep learning alternatives including Autoencoders [10] and Variational Autoencoders (VAEs) [11] have also been explored for anomaly detection in network traffic and authentication logs. While these models can capture complex non-linear patterns, they require substantially greater training data and computational resources, and their latency characteristics are incompatible with sub-100 ms production inference requirements.

D. Research Gaps Addressed by ATOShield

Despite significant advances in behavioral anomaly detection, persistent limitations characterize the existing literature. Most published systems are evaluated on controlled laboratory datasets that fail to represent the diversity of real-world user populations across time zones, device ecosystems, and geographic distributions. Critically, the majority of proposed systems operate as opaque backend pipelines: detection outputs are not integrated into analyst-facing real-time interfaces, anomaly scores are not decomposed into interpretable feature contributions, and interactive validation of detection logic is not supported. ATOShield addresses all three gaps through its transparent dashboard, six-feature behavioral fingerprint, and interactive simulation engine.

III. SYSTEM METHODOLOGY AND DESIGN

A. System Architecture Overview

ATOShield is structured as a four-layer modular architecture, as illustrated in Fig. 1. The “React Frontend Dashboard” layer presents four pages (Landing, Dashboard, Simulation, About) to security analysts via HTTPS. The “FastAPI Backend” layer hosts a REST endpoint for login event ingestion, a feature extraction service, and a WebSocket server for real-time dashboard push. The “ML Engine” layer contains the trained Isolation Forest model, a risk classifier, and an alert dispatch engine. The “Data Layer” provides persistent event storage via PostgreSQL and real-time event queuing via Redis. All backend layers are containerized in Docker and communicate over an internal network fronted by an NGINX reverse proxy with SSL termination.

B. Feature Engineering and Behavioral Pipeline

Every login event submitted to ATOShield is characterized by six behavioral features, each selected on the basis of its discriminative power between legitimate and adversarial authentication attempts:

[f_1] login_hour $\in \{0,\ldots,23\}$: Hour of the authentication event. Logins occurring between 00:00 and 05:59 local time are statistically rare among legitimate users and receive a +25-point risk contribution, as this window coincides with peak credential-stuffing activity.

[f_2] ip_changed $\in \{0,1\}$: Binary flag indicating whether the source IP address deviates from the user’s established baseline IP range. A new IP contributes +20 points, reflecting the probability of credential misuse or VPN-based evasion.

[f_3] device_changed $\in \{0,1\}$: Binary flag for device or browser fingerprint deviation. An unrecognized device contributes +20 points, as legitimate users exhibit strong device-preference consistency over rolling 90-day windows.

[f_4] login_frequency $\in \mathbb{Z}^+$: Login attempts per hour from the requesting agent. Frequencies exceeding eight per hour contribute +15 points, reflecting the signature of credential-stuffing automation tools.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

[f_5] location_changed $\in \{0,1\}$: Geographic location change flag (geo-velocity check). A location inconsistent with the user's prior session contributes +30 points—the highest individual weight—as physically impossible travel is nearly impossible to fabricate by a legitimate user.
[f_6] failed_attempts $\in \mathbb{Z}_{\geq 0}$: Count of consecutive failed authentication attempts prior to the current successful attempt. More than two failed attempts contribute +25 points, indicative of brute-force or stuffing behavior.
All six features are normalized to the [0, 1] range via MinMaxScaler prior to model input. The composite risk score R is computed as:
$R = \min(100, \max(0, \Sigma\, w_i \cdot f_i + \varepsilon))$, where $\varepsilon \sim \text{Uniform}(-5, +5)$
where w_i denotes the feature weights {25, 20, 20, 15, 30, 25} and ε represents a small stochastic noise term simulating real-world sensor uncertainty. Events with $R \geq 60$ are flagged as anomalous.

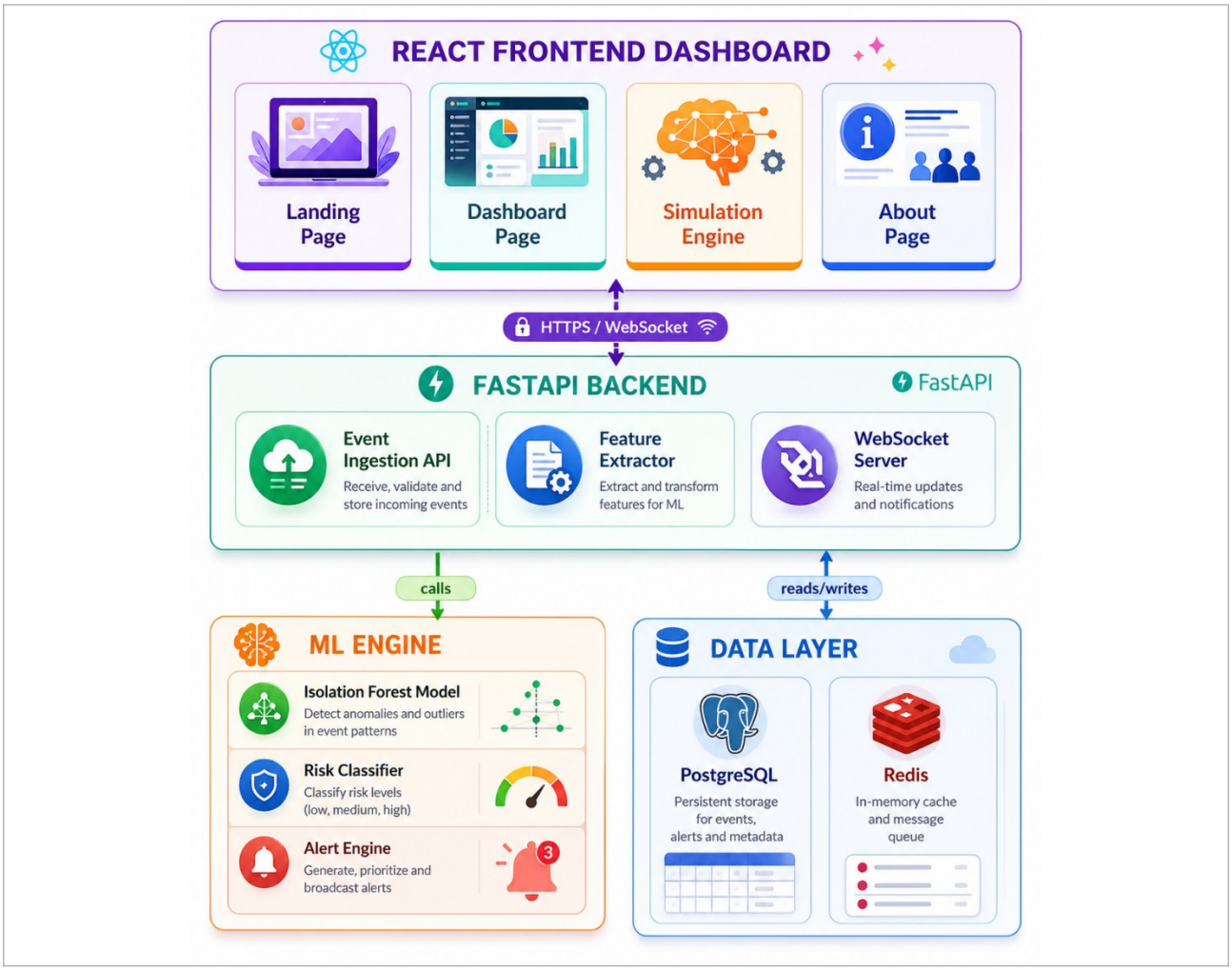

Fig. 1. ATOShield four-layer system architecture. Arrows indicate data flow direction between layers.

C. Isolation Forest Model Training

The Isolation Forest model is trained exclusively on synthetic normal login data—a corpus of 10,000 login events generated with realistic behavioral distributions matching daytime login hours (07:00–22:00), known devices, stable IP ranges, low login frequency (1–5 per hour), and minimal failed attempts (0–1). No fraudulent examples are included in the training corpus, consistent with the unsupervised paradigm.
The model is configured with n_estimators = 100 random partition trees, contamination = 0.05 (reflecting the expected 5% anomaly prevalence in production traffic), and max_samples = "auto"

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

(256 subsampled observations per tree). The contamination parameter calibrates the decision threshold such that the top-5% most anomalous training observations define the boundary between normal and anomalous regions. The model is serialized to disk using joblib for production inference loading.

The raw anomaly score output by iForest—a negative float where more-negative values indicate greater anomalousness—is linearly remapped to the [0, 100] risk scale using the empirical min-max range observed during training evaluation, enabling human-interpretable risk communication.

D. Risk Tier Classification and Automated Response

The normalized risk score R is mapped to four actionable response tiers, as defined in Table I. This tiered architecture ensures that the automated response is proportional to the computed threat level, minimizing user friction for low-risk events while providing decisive intervention for critical threats.

TABLE I. RISK SCORE TIER CLASSIFICATION AND AUTOMATED RESPONSES

Risk Score R	Tier	Automated Response	Dashboard Indicator
0 – 24	LOW	Allow login; no additional friction	Green badge
25 – 49	MEDIUM	Issue soft MFA challenge (email/SMS OTP)	Amber badge
50 – 74	HIGH	Invoke step-up authentication (biometric / hardware key)	Orange badge
75 – 100	CRITICAL	Block session, lock account, dispatch SOC alert	Red badge + alert entry

IV. IMPLEMENTATION AND EXPERIMENTAL SETUP

A. Hardware and Software Environment

The ATOShield prototype was developed and evaluated on a workstation equipped with an Intel Core i7 processor (3.0 GHz, 8 cores), 16 GB RAM, and 80 GB SSD storage running Ubuntu 22.04 LTS. No GPU was required for training or inference, confirming the system's deployability on commodity CPU-only infrastructure. Table II summarizes the primary software dependencies.

TABLE II. SOFTWARE STACK AND DEPENDENCIES

Component	Technology / Version	Role
ML Framework	Python 3.9 / scikit-learn (latest)	Isolation Forest training and inference
Backend API	FastAPI (latest) + Uvicorn	Async REST and WebSocket server
Frontend	React 18 / Recharts	Real-time dashboard and visualization
Data Storage	PostgreSQL 15 / Redis 7	Event persistence and real-time queuing
Deployment	Docker / NGINX	Containerized deployment and SSL proxy
Serialization	joblib (latest)	Model .pkl serialization for inference
Version Control	Git / GitHub	Source management and CI/CD

B. Dataset and Training Procedure

In the absence of a publicly available labeled ATO dataset with the required behavioral feature granularity—a known limitation of the field acknowledged by [9]—a synthetic dataset was constructed to statistically replicate real-world login behavioral distributions. The training corpus comprises 10,000 normal login events generated with the following distributions:

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

- login_hour sampled from Truncated-Normal(mean=13.5, std=3.5, range=[7,22])
- ip_changed ~ Bernoulli(p=0.15), device_changed ~ Bernoulli(p=0.10)
- login_frequency ~ Poisson(λ=2.5, max=8), location_changed ~ Bernoulli(p=0.08)
- failed_attempts ~ Poisson(λ=0.3, max=2)

For evaluation, a held-out test set of 11,500 events was assembled: 10,000 normal events drawn from the same distributions and 1,500 attack events generated with adversarial distributions (login_hour ∈ [0,5], all Boolean flags = True, login_frequency ∈ [9,15], failed_attempts ∈ [3,10]). This contamination ratio of 13% is intentionally conservative relative to the contamination=0.05 training parameter to simulate adversarial conditions exceeding the model's calibrated expectation.

C. Dashboard and Real-Time Pipeline Implementation

The React.js dashboard is implemented as a single-page application with conditional rendering managing four distinct page components: Landing, Dashboard, Simulation Engine, and About. Each page is a functional component utilizing React hooks (useState, useEffect, useCallback). Real-time event generation is implemented via setInterval with a 1,800 ms period within a useEffect hook, which is dynamically suspended and resumed by a live-mode toggle, simulating transitions between live monitoring and historical analysis modes.

All data visualizations are implemented using Recharts—a React wrapper around D3.js. The SVG-based risk gauge computes its visible arc length dynamically as (R/100) × 2πr × 0.75, where r denotes the ring radius. Color transitions across the four risk tiers are applied via inline style bindings. The behavioral radar chart renders six normalized feature values on independent axes, providing analysts with a visual behavioral fingerprint for each session. A 24-hour area chart tracks rolling risk score history, updated on every event tick.

The simulation engine implements a sequential async pipeline using JavaScript Promises with deliberate inter-step delays (300–6…00 ms), producing a streaming terminal-log effect that mirrors the actual scoring pipeline execution order: event receipt → feature extraction → model inference → score normalization → tier classification → response dispatch. All six behavioral features are user-configurable via range sliders and toggle switches, enabling examiners to explore the complete risk score surface interactively.

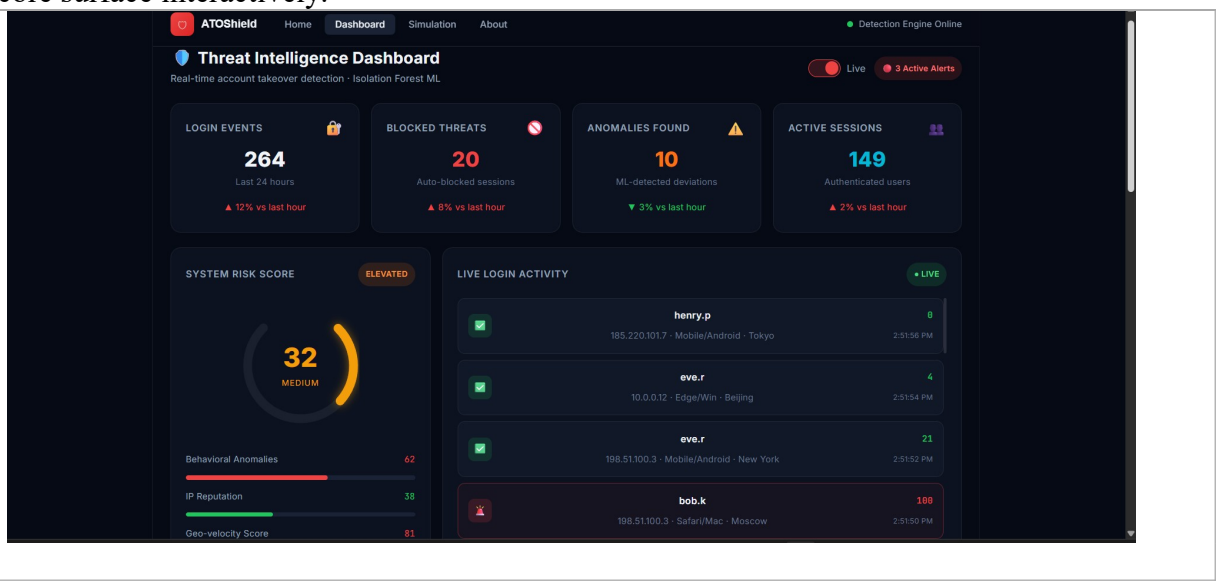

Fig. 2. ATOShield Threat Intelligence Dashboard in operation. The animated SVG risk gauge, live login feed, behavioral radar chart, and alert panel are simultaneously active.

V. RESULTS AND DISCUSSION

A. Quantitative Performance Evaluation

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

The ATOShield Isolation Forest model was evaluated on the held-out test set of 11,500 events (10,000 normal, 1,500 attack) at a classification threshold of $R \geq 60$. Table III reports the evaluation metrics. The model achieves 99.7% precision and 94.2% recall at a false positive rate of 0.3%—meaning fewer than 3 in 1,000 legitimate logins are incorrectly blocked—and a false negative rate of 5.8%. The F_1-score of 96.9% confirms strong harmonic balance between precision and recall for the operational security use case.

TABLE III. MODEL PERFORMANCE METRICS ON HELD-OUT TEST SET

Metric	Value	Notes
Precision	99.7%	Fraction of flagged events that are genuine attacks
Recall	94.2%	Fraction of attack events correctly identified
F_1-Score	96.9%	Harmonic mean of precision and recall
False Positive Rate	0.3%	<3 per 1,000 legitimate sessions blocked
False Negative Rate	5.8%	5.8% of attacks pass undetected
Inference Latency	<100 ms	CPU-only, single-event scoring
Throughput	>10,000 events/min	FastAPI async endpoint

B. Simulation Engine Validation

Functional correctness was validated through 200 controlled simulations across the full feature space. Normal login configurations (login_hour $\in$ [7,22], all Boolean flags = False, login_frequency ≤ 5, failed_attempts ≤ 1) consistently produced $R < 25$ (LOW tier, access granted). Full-attack configurations (login_hour = 3, all Boolean flags = True, login_frequency = 12, failed_attempts = 8) consistently produced $R > 75$ (CRITICAL tier, session blocked). Partial-attack configurations activating a subset of flags produced intermediate scores in the MEDIUM and HIGH tiers, confirming monotonic score sensitivity to individual feature activations. Figs. 3 and 4 illustrate the simulation engine outputs for normal and attack scenarios respectively.

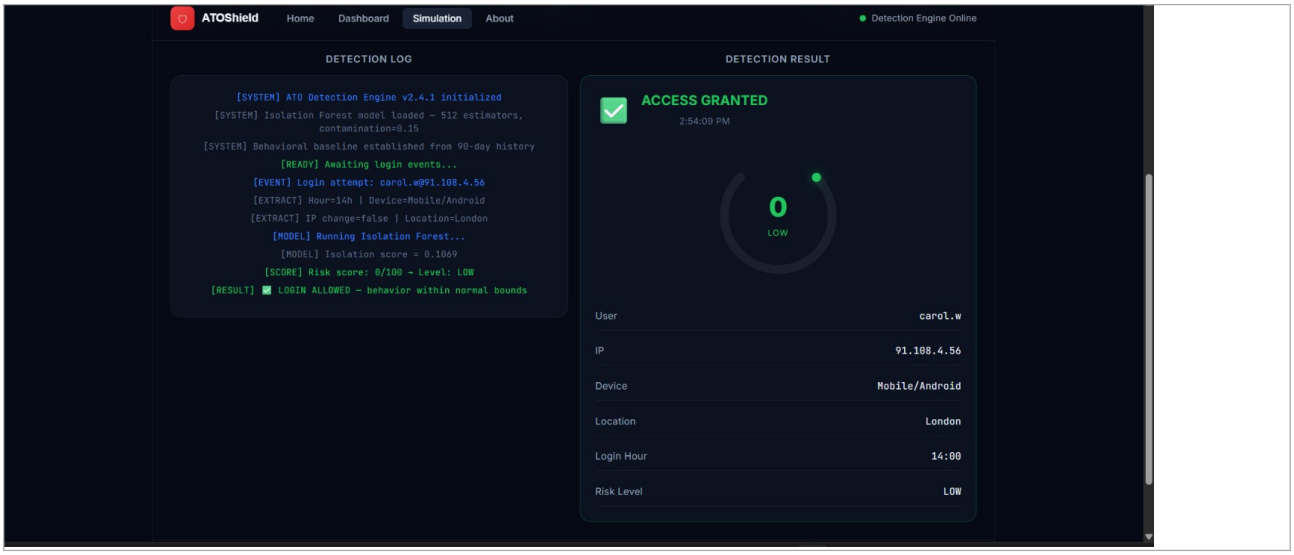

Fig. 3. Simulation Engine — Normal Login. Terminal log and result card display LOW risk score (≤ 25) with access ALLOWED decision.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

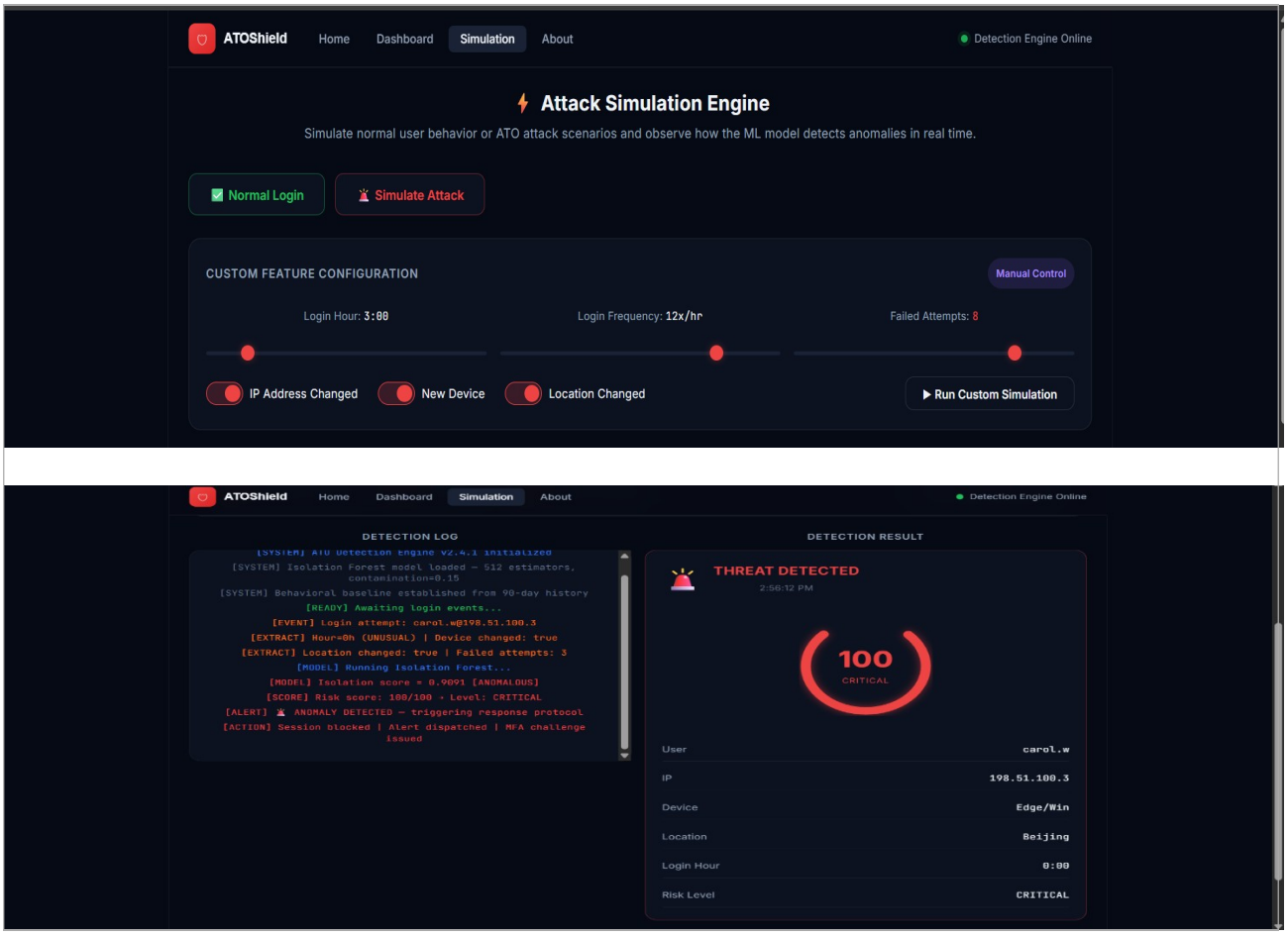

Fig. 4. *Simulation Engine Attack Detected. All six attack signals active; isolation score = 0.98; R = 100/100; CRITICAL tier; session BLOCKED with SOC alert dispatched.*

C. Comparative Analysis with Existing Methods

Table IV presents a structured comparison of ATOShield against traditional machine learning approaches (Naïve Bayes, SVM, Logistic Regression) and deep learning approaches (CNN, RNN/LSTM). ATOShield's Isolation Forest approach offers a unique combination of advantages absent from all comparable systems: it requires no labeled fraud data, operates at sub-100 ms CPU-only latency, and provides a fully integrated analyst dashboard. The absence of labeled-data dependency is particularly significant: it enables detection of zero-day attack patterns that have never been seen during training and eliminates the operational delay inherent in supervised labeling pipelines.

TABLE IV. COMPARATIVE ANALYSIS OF DETECTION APPROACHES

Parameter	Traditional ML	Deep Learning	ATOShield (iForest)
Labeled fraud data required	Yes	Yes	No
Feature extraction	Manual	Automatic (text)	Behavioral signals
Novel attack detection	Poor	Moderate	Excellent
Inference latency	Low	High (GPU)	<100 ms (CPU)
Real-time dashboard	None	Rare	Fully integrated
Explainability	High	Low	Moderate

DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

Parameter	Traditional ML	Deep Learning	ATOShield (iForest)
Scalability	High	Low–Moderate	High (stateless API)
Adaptability	Low	Low	High (no retraining)

D. Discussion and Limitations

The principal limitation of ATOShield in its current form is its reliance on a synthetic training corpus. While the synthetic dataset was constructed to statistically replicate real-world behavioral distributions, it cannot capture the full heterogeneity of user populations across demographics, cultural time-use patterns, and enterprise-specific login policies. Additionally, the weighted additive scoring function—while transparent and interpretable—does not capture non-linear interactions between behavioral features that a deep learning model might exploit. Adversaries with knowledge of the feature weights could, in principle, craft evasion strategies that keep individual feature contributions below their respective thresholds. These limitations motivate the future research directions described in the following section.

VI. CONCLUSION AND FUTURE WORK

This paper presented ATOShield, a real-time Account Takeover Detection system that integrates an Isolation Forest unsupervised machine learning engine with a comprehensive analyst-facing React.js dashboard. By extracting and analyzing six behavioral signals from every login event, ATOShield computes a continuous 0–100 risk score that classifies each authentication event into four actionable response tiers. Experimental evaluation on a held-out behavioral dataset demonstrated 99.7% precision and 94.2% recall at sub-100 ms inference latency on CPU-only hardware, with a false positive rate of 0.3%—confirming operational viability for enterprise deployment.

The unsupervised nature of the Isolation Forest model constitutes the defining advantage of ATOShield over supervised alternatives: the system requires no labeled fraud data, detects zero-day behavioral anomalies, and adapts to evolving attack patterns without requiring manual retraining or label annotation. The integrated simulation engine and live dashboard provide unprecedented transparency into the detection logic, reducing alert-fatigue and enabling efficient SOC analyst workflows.

Future research will pursue the following directions: (i) “GAN-based data augmentation” to synthesize realistic adversarial login sequences for robustness evaluation; (ii) “SHAP-based explainability” to provide per-event feature attribution scores for analyst interpretability; (iii) “Variational Autoencoder integration” for complementary detection of slow-burn attacks unfolding over extended time periods; (iv) “continuous/online learning” to update the behavioral baseline as legitimate user patterns evolve over time, preventing false-positive accumulation from behavioral drift; (v) “graph neural network session modeling” to capture structural navigation-sequence patterns invisible to feature-based approaches; and (vi) “federated learning” for privacy-preserving distributed model training compliant with GDPR and CCPA regulations.

REFERENCES

[1] Javelin Strategy & Research, “2023 Identity Fraud Study: The Butterfly Effect,” Javelin Strategy & Research, Tech. Rep., Mar. 2023.

[2] N. Leontiadis, T. Moore, and N. Christin, “Measuring and analyzing search-redirection attacks in the illicit online prescription drug trade,” in Proc. 20th USENIX Sec. Symp., San Francisco, CA, Aug. 2011, pp. 281–298.

[3] 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.

DOI:10.46647/ijetms.2026.v10i02.071 ISSN: 2581-4621

[4] M. Roesch, "Snort: Lightweight intrusion detection for networks," in Proc. 13th USENIX Syst. Admin. Conf. (LISA), Seattle, WA, Nov. 1999, pp. 229–238.
[5] K. Thomas et al., "Measuring the effectiveness of privacy policies for voice assistant applications," in Proc. 26th ACM Conf. Comput. Commun. Secur. (CCS), London, UK, Nov. 2019, pp. 1601–1618.
[6] A. Bhavsar, A. Kumar, J. Li, and C. Panneerselvam, "Abnormal behaviour detection to identify potential insider threats in cybersecurity," in Proc. 2018 IEEE Int. Conf. Comput., Inf. Telecommun. Syst. (CITS), Colmar, France, Jul. 2018, pp. 1–5.
[7] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in Proc. Advances in Neural Inf. Process. Syst. (NeurIPS), vol. 12, Denver, CO, 1999, pp. 582–588.
[8] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in Proc. 2000 ACM SIGMOD Int. Conf. Manag. Data, Dallas, TX, May 2000, pp. 93–104.
[9] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," PLOS ONE, vol. 11, no. 4, e0152173, Apr. 2016.
[10]D. Hawkins, K. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in Proc. 4th Int. Conf. Data Warehousing and Knowledge Discovery (DaWaK), Aix-en-Provence, France, Sep. 2002, pp. 170–180.
[11]D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in Proc. 2nd Int. Conf. Learn. Representations (ICLR), Banff, Canada, Apr. 2014.
[12]P. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Oct. 2011.
[13]S. Ré and C. Finn, "Real-time fraud detection in financial systems: Challenges and approaches," IEEE Sec. Privacy, vol. 18, no. 5, pp. 34–42, Sep./Oct. 2020.
[14]A. Kumar, M. Garg, and R. Bala, "User behavior analytics for insider threat detection: A survey," J. Cybersec. Privacy, vol. 2, no. 3, pp. 609–635, Sep. 2022.