iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85641

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 XI Nov 2026- Available at www.ijraset.com

Machine Learning-Based Account Takeover Detection System Using Behavioral Login Analysis and Risk-Adaptive Authentication

Nidha[2], Satya Kalpana[3], Kunta Likhitha[4], Mr. Raghavendra Gupta[4]
[1, 2, 3]*Student, Dept. of Computer Science and Engineering, HITAM, Telangana, India*
[4]*Associate Professor, Dept. of Computer Science and Engineering, HITAM, Telangana, India*

Abstract: ***Account Takeover (ATO) is a major cybersecurity threat in which attackers gain unauthorized access to legitimate user accounts. This paper proposes a Machine Learning-Based Account Takeover Detection System that analyzes login activities such as IP address, device details, location, login time and failed login attempts to identify suspicious behavior. Logistic Regression and Random Forest algorithms are used to predict login risk and to generate a risk score between 0 and 1. Based on the risk level, the system takes one of three actions: Allow, Multi-Factor Authentication (MFA) or Block. The trained model is served through a FastAPI backend for real-time prediction, a Streamlit interface lets users submit login attributes and view the decision, and every prediction is logged in an SQLite database. The proposed system aims to improve account security, detect suspicious login attempts and prevent unauthorized access.***

Index Terms: ***Account Takeover, Machine Learning, Cybersecurity, Risk Score, Random Forest, Logistic Regression, Multi-Factor Authentication***

I. INTRODUCTION

Online services such as banking, e-commerce, email and social media rely on user accounts that are protected mainly by passwords. Large volumes of credentials are leaked through data breaches, phishing and malware, and attackers reuse them to log in to other services [1]. When an attacker signs in to a legitimate account in this way, the event is called an Account Takeover (ATO). Passwords alone remain a weak line of defense because they are easy to steal, guess and reuse [2].

A login that uses the correct password is not necessarily a login by the correct person. Signals such as a previously unseen device, an unusual location, a login at an odd hour, a reputed-bad IP address and several failed attempts before success can reveal that something is wrong. Risk-based authentication uses such signals to estimate how risky a login is and to ask for extra proof only when the risk is high [3], [4]. This avoids burdening every user with a second factor on every login while still stopping suspicious access. This paper presents a Machine Learning-Based Account Takeover Detection System that scores every login attempt and maps the score to an action: Allow, Multi-Factor Authentication (MFA) or Block. Logistic Regression [5] and Random Forest [6] classifiers are trained on behavioral login features, the model is exposed through a FastAPI service, and a Streamlit application provides the user interface. The emphasis of this work is on a practical, end-to-end pipeline, from feature design to a deployed, logged decision service.

The main contributions of this paper are as follows.

- A set of six behavioral login features (device novelty, location risk, failed login attempts, login time deviation, IP reputation risk and account age risk) that capture common ATO indicators.
- A comparison of Logistic Regression and Random Forest for predicting the probability that a login attempt is risky.
- A threshold-based decision policy that converts the predicted probability into an Allow, MFA or Block action.
- A deployable architecture with a FastAPI prediction service, a Streamlit front end and SQLite logging of every prediction for monitoring.

The remainder of the paper is organized as follows. Section II reviews background and related work. Section III states the problem and objectives. Section IV presents the system architecture, Section V the dataset and features, and Section VI the machine learning models and decision logic. Section VII describes the implementation, Section VIII presents results and discussion, Section IX covers security considerations, Section X lists limitations and future work, and Section XI concludes the paper.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

II. BACKGROUND AND RELATED WORK

A. *Account Takeover Attacks*

ATO attacks are commonly carried out through credential stuffing, phishing, brute-force guessing and malware. An analysis of stolen-credential markets shows that phishing kits and keyloggers collect large numbers of valid credentials, and that attackers often try them against many services [1]. Once access is obtained, the attacker can steal data, make fraudulent transactions or lock out the legitimate owner.

B. *Risk-Based Authentication*

Risk-based authentication (RBA) evaluates contextual features of a login, such as IP address, device and location, and requires stronger verification only when the login looks unusual. Studies of RBA in real services show that it can be applied without noticeably harming usability [4], and statistical approaches can estimate how likely it is that a login comes from the genuine user [3]. Current guidance on digital identity also recommends considering context and multi-factor authentication for sensitive access [11].

C. *Machine Learning for Login Risk*

Logistic Regression models the probability of a binary outcome and offers interpretable coefficients [5]. Random Forest builds many decision trees on random subsets of data and features and combines their votes, which handles non-linear relations and reduces over-fitting [6]. Both are available in scikit-learn [7]. This work uses both so that a transparent baseline can be compared with a stronger ensemble.

III. PROBLEM STATEMENT AND OBJECTIVES

The problem addressed in this work is to detect, in real time, login attempts that are likely to be account takeovers, even when the correct credentials are supplied, and to respond in proportion to the risk. The objectives are as follows.

1) Identify login behavior features that indicate suspicious access.
2) Train and compare Logistic Regression and Random Forest models to predict login risk.
3) Generate a risk score for each login attempt.
4) Map the risk score to Allow, MFA or Block actions.
5) Provide a real-time prediction service and a user interface, and log all decisions.

IV. SYSTEM ARCHITECTURE

A. *Technology Stack*

The system is implemented in Python. Data handling uses Pandas and NumPy, models are built with scikit-learn [7] and saved with Joblib, the prediction service is built with FastAPI [8], the interface with Streamlit [9], and prediction logs are stored in SQLite [10]. Table 1 summarizes the stack.

Table 1 Technology Stack of the System

Layer	Technology	Purpose
Data processing	Pandas, NumPy	Cleaning and feature preparation
Machine learning	scikit-learn, Logistic Regression, Random Forest, Joblib	Model training, evaluation and model persistence
Backend	FastAPI	Real-time prediction endpoint
Frontend	Streamlit	User-friendly input form and decision display
Storage	SQLite	Logging of prediction results

B. *System Workflow*

The system follows the pipeline shown in Fig. 1. Login attributes are collected, converted into numeric features and passed to the trained model, which returns a risk score. The decision module compares the score with two thresholds and returns Allow, MFA or Block. The result is shown to the user and written to the log database.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

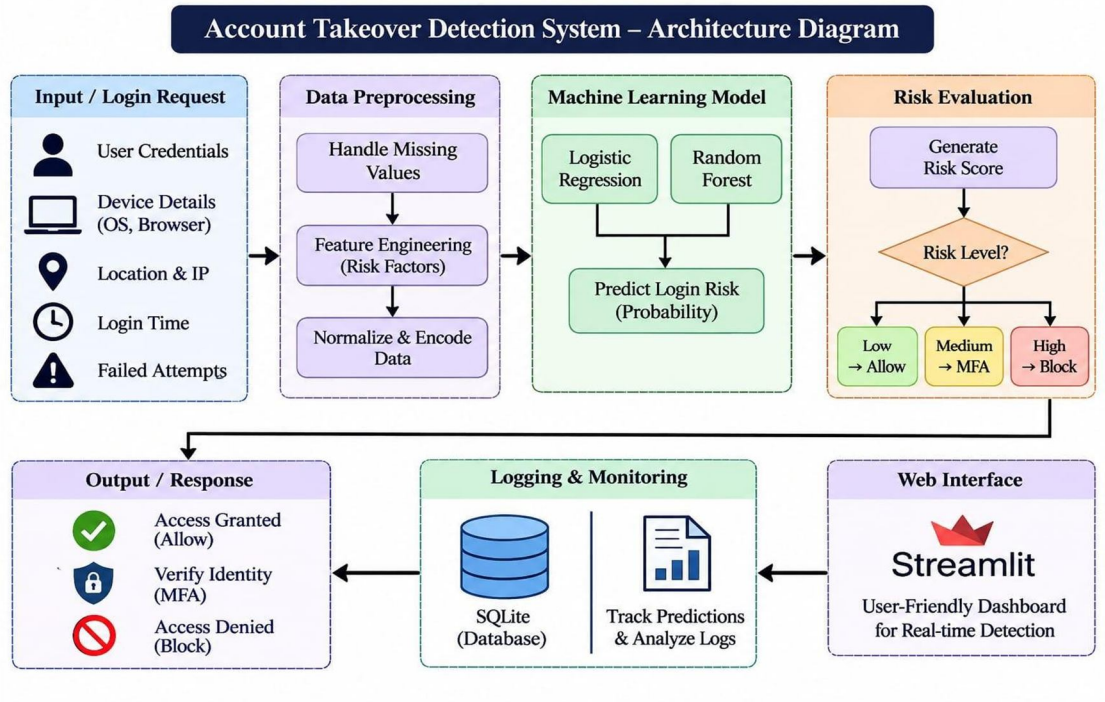


Fig. 1 System architecture of the Account Takeover Detection System.

V. DATASET AND FEATURE ENGINEERING

Each record in the dataset represents one login attempt and carries a label that marks it as normal or risky. The data are cleaned, missing values are handled, and the features in Table 2 are derived from the raw login attributes (IP address, device details, location, login time and failed attempts). [Describe here the dataset source, number of records and class balance.]

Table 2 Behavioral Features Used for Risk Prediction

Feature	Description	Raw attribute
Device novelty	Whether the device has been seen before for this account	Device details
Location risk	Deviation of the login location from the usual locations	Location
Failed login attempts	Number of failed attempts before the login	Login history
Login time deviation	Difference between login time and the user's usual hours	Login time
IP reputation risk	Risk level associated with the source IP address	IP address
Account age risk	Higher risk for newly created accounts	Account profile

The data are split into training and test sets, and the models are evaluated only on the unseen test portion. [Mention the split ratio and any scaling or encoding used.]

VI. MACHINE LEARNING MODELS AND DECISION LOGIC

A. Logistic Regression

Logistic Regression estimates the probability that a login is risky from a weighted sum of the features. For a feature vector x with weights w and bias b, the risk score is given by (1).

$$s = 1 / (1 + e^{-(w^{T}x + b)}) \quad (1)$$

B. Random Forest

Random Forest trains T decision trees on bootstrap samples with random feature subsets. The risk score is the average of the probabilities predicted by the trees, as in (2), where p_t(x) is the probability from tree t.

$$s = (1 / T) \Sigma p_t(x) \quad (2)$$

C. Adaptive Decision Logic

The risk score s lies between 0 and 1. It is converted into an authentication action using the thresholds in Table 3. Low-risk logins pass without friction, medium-risk logins are challenged with MFA, and high-risk logins are blocked. The thresholds can be tuned to the security needs of the service.

Table 3 Risk Score to Action Mapping

Risk score	**Action**	**Meaning**
$s < 0.30$	ALLOW	Normal login; no extra verification
$0.30 \leq s \leq 0.70$	MFA	Suspicious login; ask for a second factor
$s > 0.70$	BLOCK	Likely account takeover; deny access

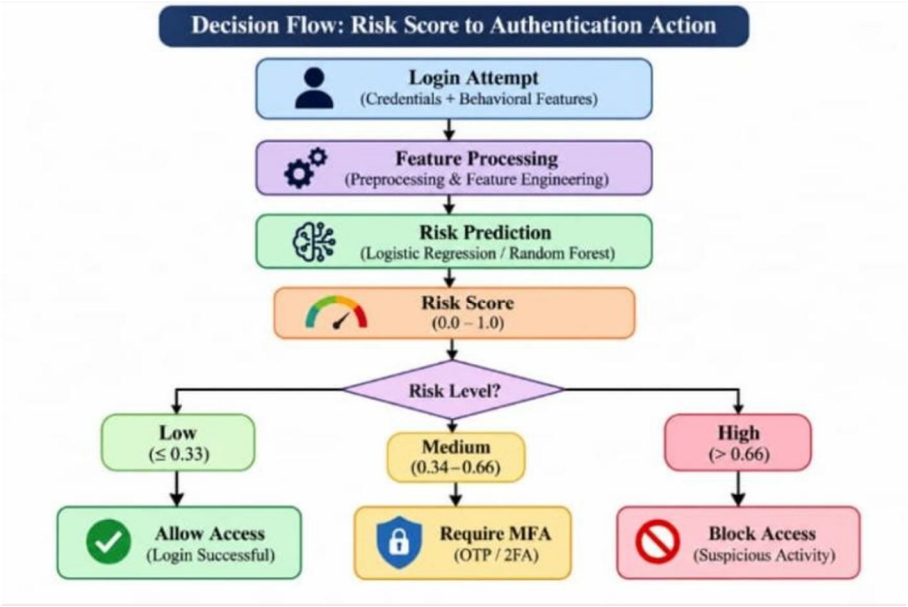


Fig. 2 Decision flow from risk score to authentication action.

VII. IMPLEMENTATION

A. Model Training and Persistence

Data preprocessing, feature engineering, training and evaluation are carried out in a notebook. The model that performs best is saved with Joblib so that the backend can load it once at start-up and reuse it for every request.

B. FastAPI Backend

The backend exposes a prediction endpoint. It receives the login features as a JSON request, validates them, computes the risk score with the saved model, applies the decision thresholds and returns the score and action. FastAPI also generates interactive API documentation automatically [8].

C. Streamlit Frontend and Logging

The Streamlit application provides a form in which the user enters login details and receives the risk score and the recommended action [9]. Each request and its result are stored in an SQLite table, which supports monitoring and later analysis of model behavior [10]. The project is deployed online, with the front end and the API hosted separately.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

VIII. RESULTS AND DISCUSSION

This section presents the main screens of the system and the model evaluation. Fig. 3 shows the main interface of the application. Fig. 4, Fig. 5 and Fig. 6 show examples of the three decisions: Allow, MFA and Block.

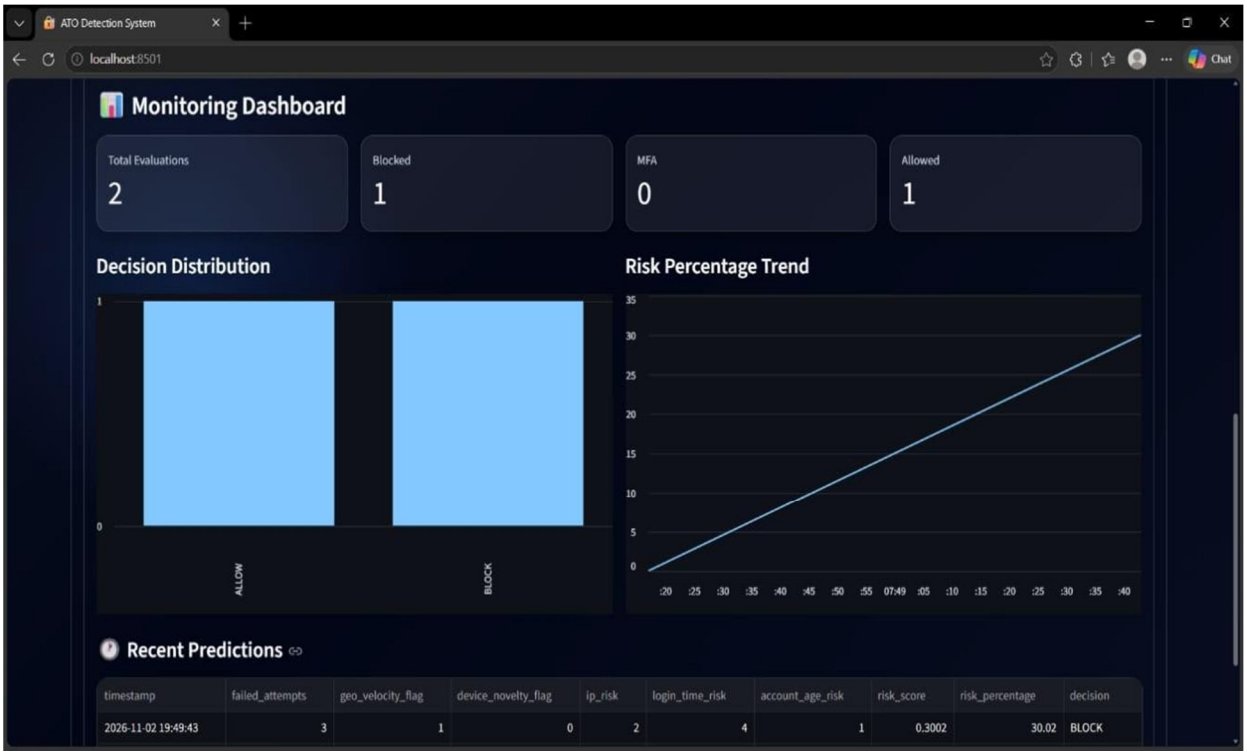


Fig. 3 shows the interactive API documentation.

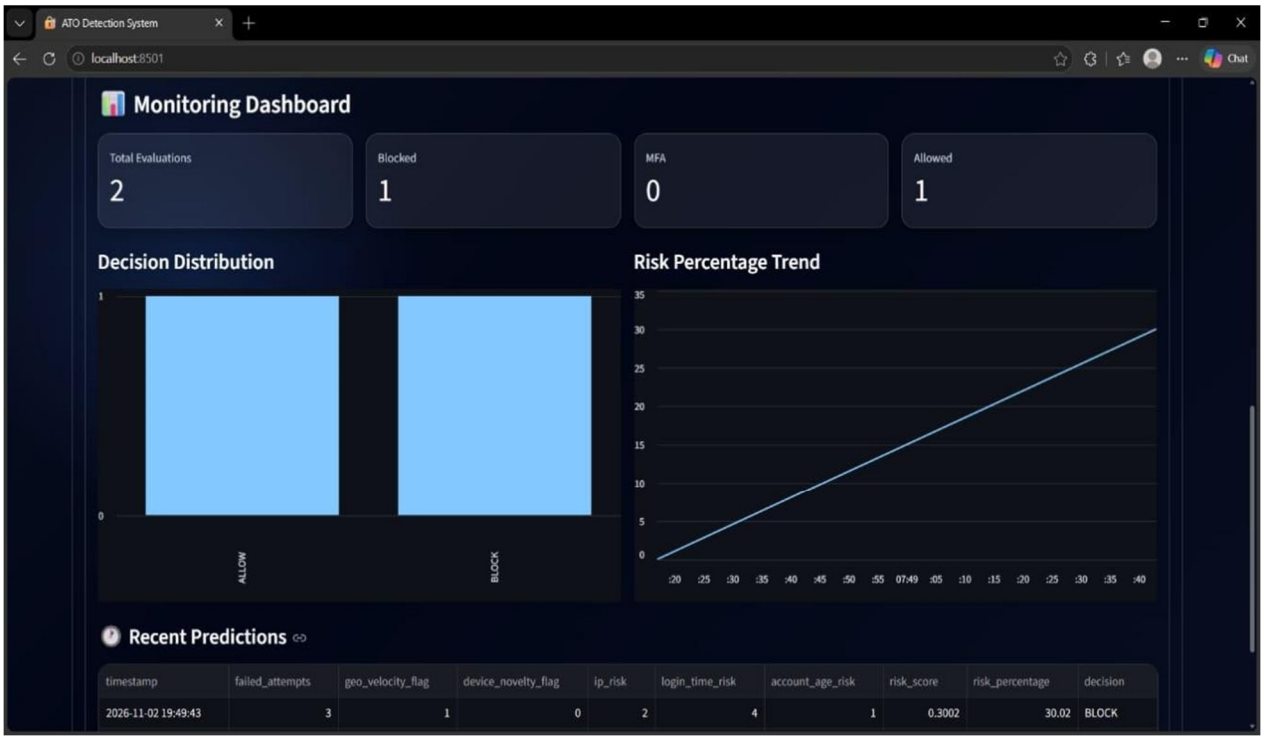


Fig. 4 Low-risk login classified as ALLOW

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

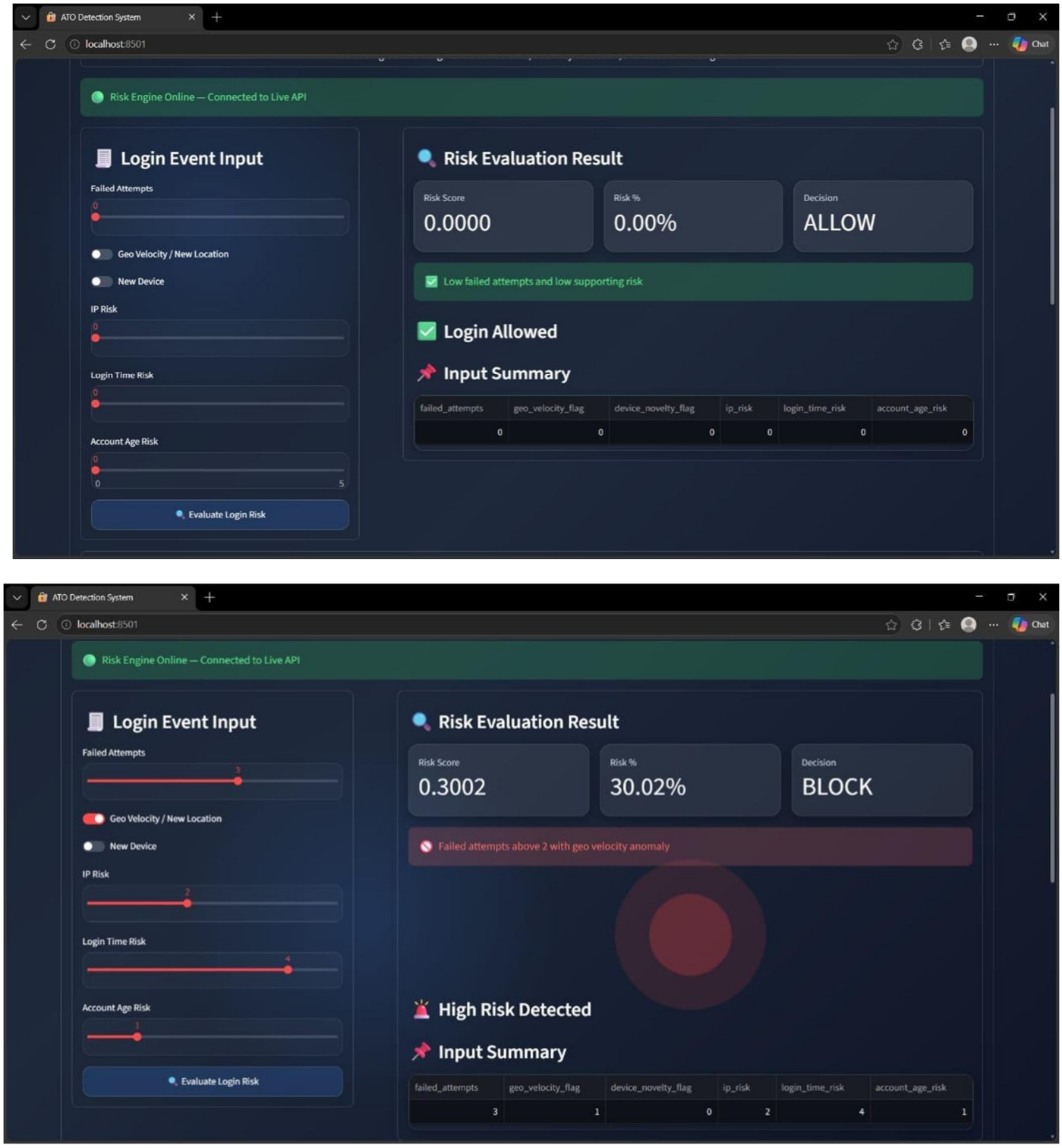


Fig 5 High-risk login classified as BLOCK

A. Model Performance

Table 4 compares the two models on the test data using accuracy, precision, recall, F1-score and ROC-AUC. Recall is especially important because a missed takeover is more costly than an extra MFA prompt.

Table 4 Performance Comparison of the Models

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	[XX.XX]	[XX.XX]	[XX.XX]	[XX.XX]	[XX.XX]
Random Forest	[XX.XX]	[XX.XX]	[XX.XX]	[XX.XX]	[XX.XX]

```
TRAIN PERFORMANCE
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    312381
           1       1.00      1.00      1.00     23228

    accuracy                           1.00    335609
   macro avg       1.00      1.00      1.00    335609
weighted avg       1.00      1.00      1.00    335609


TEST PERFORMANCE
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     78096
           1       1.00      1.00      1.00      5807

    accuracy                           1.00     83903
   macro avg       1.00      1.00      1.00     83903
weighted avg       1.00      1.00      1.00     83903
```

```
Logistic Regression Performance:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     78096
           1       1.00      1.00      1.00      5807

    accuracy                           1.00     83903
   macro avg       1.00      1.00      1.00     83903
weighted avg       1.00      1.00      1.00     83903

ROC-AUC: 1.0000

Random Forest Performance (Calibrated):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     78096
           1       1.00      1.00      1.00      5807

    accuracy                           1.00     83903
   macro avg       1.00      1.00      1.00     83903
weighted avg       1.00      1.00      1.00     83903

ROC-AUC: 1.0000
...
- Precision & Recall matter more than accuracy.
- Calibration improves probability quality.
- ROC-AUC measures ranking ability.
```

Fig. 6 Confusion matrix and ROC curve of the selected model.

B. Functional Outcome

The system returns a risk score and an action for each login in real time, handles the three risk levels with different responses, and records every decision. Because the action is derived from a score rather than a hard yes/no, the thresholds can be adjusted to trade security against user convenience. [Add a short discussion of which model performed better and why, based on Table 4.]

IX. SECURITY AND RELIABILITY CONSIDERATIONS

1) Proportional response. MFA is requested only for medium-risk logins, which limits friction for genuine users.
2) Input validation. The API validates the type and range of every input before prediction.
3) Auditability. Every prediction is logged, so decisions can be reviewed and the thresholds refined.
4) Privacy. Only the login attributes needed for scoring are used; sensitive data such as passwords are never processed by the model.

X. LIMITATIONS AND FUTURE WORK

The current implementation has the following limitations.

1) Data. The model depends on the quality and realism of the training data; real-world attack patterns change over time.
2) Static thresholds. The 0.30 and 0.70 thresholds are fixed and are not learned per user or per service.
3) Feature scope. Only six features are used; keystroke dynamics, session behavior and device fingerprints are not included.
4) Evaluation. Large-scale testing against live attack traffic has not been performed.

Future work includes adding more behavioral features, online learning to adapt to new attacks, per-user baselines, explainable risk scores, and integration with a real MFA provider.

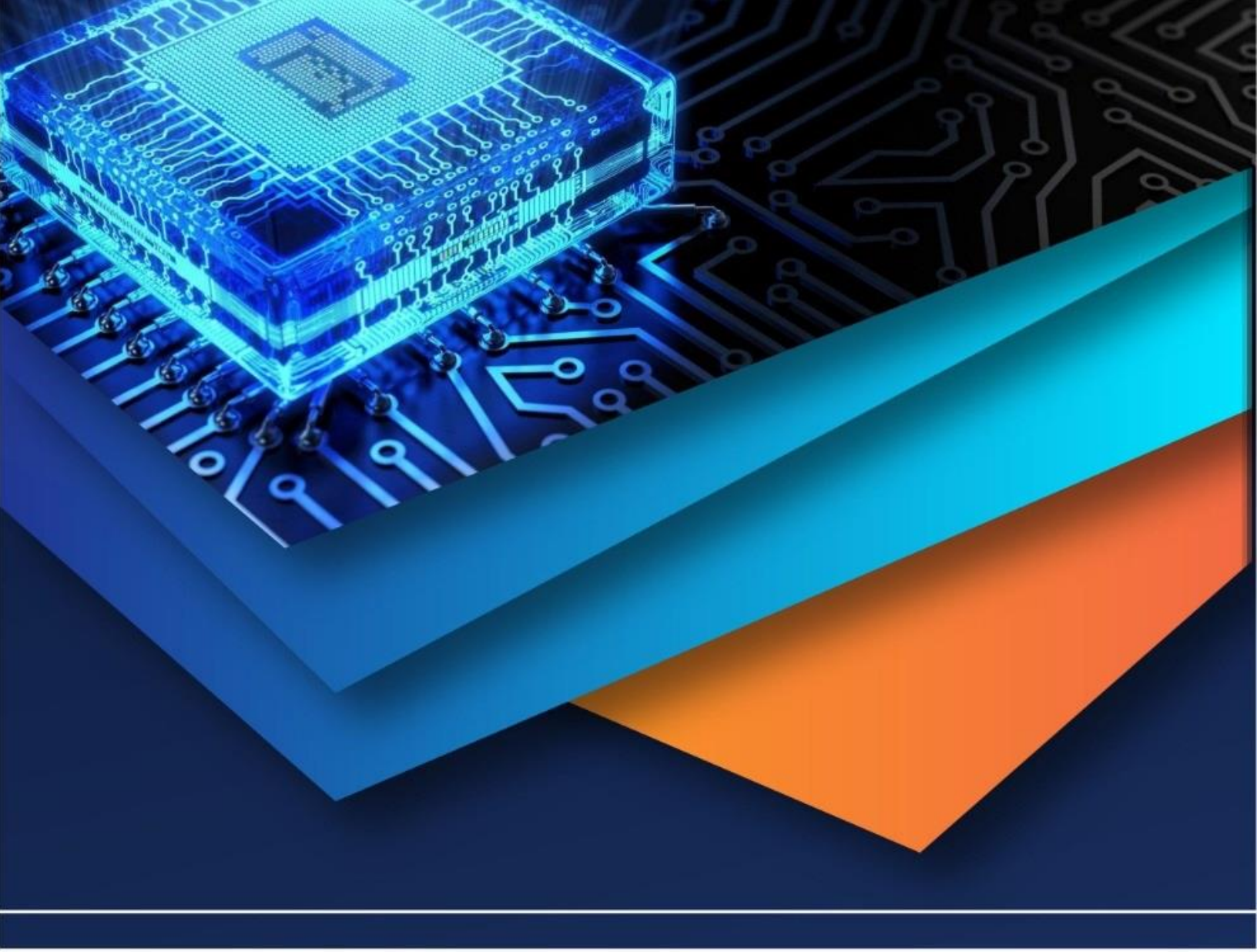
XI. CONCLUSION

This paper described a Machine Learning-Based Account Takeover Detection System that scores login attempts from behavioral features and converts the score into Allow, MFA or Block decisions. Logistic Regression and Random Forest models were used to predict login risk, a FastAPI service provides real-time predictions, a Streamlit interface supports interaction, and SQLite logging supports monitoring. The system shows that risk-adaptive authentication can improve account security and detect suspicious logins while keeping genuine logins smooth. Further work on richer features, adaptive thresholds and large-scale evaluation can make the approach suitable for production use.

REFERENCES

[1] K. Thomas et al., “Data breaches, phishing, or malware? Understanding the risks of stolen credentials,” in Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS), 2017, pp. 1421–1434.
[2] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, “The quest to replace passwords: A framework for comparative evaluation of web authentication schemes,” in Proc. IEEE Symp. Security and Privacy, 2012, pp. 553–567.
[3] D. Freeman, S. Jain, M. Dürmuth, B. Biggio, and G. Giacinto, “Who are you? A statistical approach to measuring user authenticity,” in Proc. Network and Distributed System Security Symp. (NDSS), 2016.
[4] S. Wiefling, L. Lo Iacono, and M. Dürmuth, “Is this really you? An empirical study on risk-based authentication applied in the wild,” in ICT Systems Security and Privacy Protection (IFIP SEC), 2019, pp. 134–148.
[5] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.
[6] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[7] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[8] S. Ramírez, “FastAPI documentation.” [Online]. Available: https://fastapi.tiangolo.com
[9] Streamlit Inc., “Streamlit documentation.” [Online]. Available: https://docs.streamlit.io
[10] SQLite, “SQLite documentation.” [Online]. Available: https://www.sqlite.org/docs.html
[11] P. A. Grassi et al., “Digital identity guidelines: Authentication and lifecycle management,” NIST Special Publication 800-63B, 2017.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)