

A Federated Learning Based Privacy Preserving Dynamic Access Control Framework for Distributed Systems

PAUL NDUDIRI OHIA, EMMANUEL WARIBO OTITI

1, Department of Computer Science, University of Port Harcourt Choba, Nigeria

2, Department of Computer Science, Bayelsa State Medical University, Nigeria

E-mails: paul.ohia@uniport.edu.ng

ARTICLE INFO

Article history:

Received 20 Oct 2024
Accepted 19 Nov 2024
Available online 24 Nov 2024

Keywords:

Federated Learning, Dynamic Access Control, Privacy Preservation, FedAvg, FedProx, Distributed Systems, Behaviour Analytics, Edge Security

Indexed in:



INDEX COPERNICUS
INTERNATIONAL



and in major libraries

ABSTRACT

Traditional access control systems usually depend on centralized authentication, authorization policies, and user activity monitoring. Although this approach is useful in small environments, it becomes risky in distributed systems where institutions, branches, or edge nodes generate sensitive behaviour data. Sending all access logs to a central server may expose private user activities, device information, login patterns, protected-resource usage, and institutional security operations. This study developed and evaluated a Federated Learning-Based Privacy-Preserving Dynamic Access Control Framework for Distributed Systems. The completed web application simulated three institutions: University A, University B, and University C. Each institution maintained its own local access logs, trained a local Logistic Regression model, and submitted model updates rather than raw logs to a central federated aggregator. The aggregator applied Federated Averaging to generate a global model, which was redistributed to the participating institutions for dynamic access-risk prediction. The system used behavioural features such as login time, device fingerprint, IP/location risk, requested resource, action type, failed attempts, request rate, session age, user role, and institution code. The implemented system generated 2,404 access-control records, completed six FedAvg rounds, recorded 18 local model-update submissions, and produced graduated access decisions including Allow Access, Allow with Monitoring, Step-up Verification, Restricted Access, and Deny and Alert Admin. The results showed that the framework supported collaborative access-control intelligence while keeping raw institutional access logs local.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information,

see <https://creativecommons.org/licenses/by/4.0/>

1. INTRODUCTION:

Access control is a major security mechanism used to determine whether a user should be allowed to access a protected resource. Traditional systems often rely on role-based access control, attribute-based access control, or rule-based authorization. These models are important, but they commonly focus on whether a user is authorized rather than whether the current access behaviour is safe.

Modern distributed systems create a stronger challenge. Universities, hospitals, banks, government agencies, and branch offices often operate across many locations. Each location produces access logs that may contain sensitive information about user roles, login times, devices, resources, actions, and operational patterns. If these logs are transferred to a central server for machine learning, privacy risk increases.

Federated Learning provides a useful alternative. Instead of sending raw access logs to one central model, each institution trains a local model using its own data. The central server receives only model updates. It aggregates those updates into a global model and sends the updated

model back to participating institutions. This paper applies that idea to dynamic access control.

1.1. PROBLEM STATEMENT:

Distributed computing environments require access-control mechanisms that can adapt to changing user and device conditions. However, traditional centralized and static access-control approaches may introduce privacy, scalability, and single-point-of-failure concerns while providing limited adaptation to abnormal user behaviour. Ayedh et al. [1] proposed an adaptable and distributed access-control model that applies machine learning to resolve policy conflicts in BYOD environments. Although the system supports distributed and dynamic access decisions, it does not address collaborative learning across independent institutions without sharing sensitive user and access data. Therefore, there is a need for a privacy-preserving dynamic access-control framework that enables institutions to collaboratively learn from distributed behavioural data while retaining local data ownership. This study addresses this gap by integrating federated learning

with dynamic access control, allowing institutions to share model updates rather than raw access logs.

2. RELATED WORK:

Agrawal et al. [1] reviewed the application of Federated Learning (FL) to intrusion detection systems, highlighting its potential to improve security monitoring while reducing the need to centralize sensitive network data. The authors also discussed challenges involving communication overhead, privacy, and model performance. However, their study focused primarily on intrusion detection rather than integrating federated learning directly into dynamic access-control decisions.

Campos et al. [2] evaluated federated learning for intrusion detection in Internet of Things (IoT) environments. Their study investigated the effects of heterogeneous data distributions and aggregation methods on attack classification. Although the findings highlighted the importance of handling non-identically distributed data in collaborative security models, the study did not specifically address behavioural authorization across distributed institutions.

Hamdi [3] proposed a federated learning-based intrusion detection system for IoT environments to address the privacy risks associated with centralized data collection. The study explored the use of distributed model training to detect potential intrusions while reducing the exposure of sensitive device data. However, the approach concentrated on intrusion detection rather than using learned user behaviour to determine whether individual access requests should be granted or denied.

Sarhan et al. [4] proposed a federated cyber-threat intelligence-sharing scheme for network intrusion detection. Their approach enables participating organizations to collaborate in identifying network threats through federated learning. This supports distributed security intelligence without requiring organizations to exchange their raw network data. Nevertheless, the proposed framework focuses on sharing threat intelligence for intrusion detection rather than providing an integrated mechanism for dynamic access authorization.

Javeed et al. [5] proposed a federated learning-based zero-trust intrusion detection system for IoT environments. Their approach combines convolutional neural networks and bidirectional long short-term memory networks to capture spatial and temporal characteristics of network traffic. Evaluations using CICIDS2017 and Edge-IIoTset demonstrated the potential of the proposed framework for detecting cyber threats in distributed environments. However, its primary objective is intrusion detection rather than adaptive authorization based on user identity, access context, and behavioural history.

Collectively, these studies demonstrate the potential of federated learning for privacy-conscious collaborative security analysis. However, the reviewed approaches primarily address intrusion detection, threat intelligence,

and IoT security. The integration of federated learning into dynamic access-control systems remains an important research direction, particularly where institutions must learn from distributed behavioural data without sharing sensitive records. To address this gap, the present study proposes a federated dynamic access-control framework that combines locally learned behavioural patterns with access-control policies to support adaptive authorization across distributed institutions.

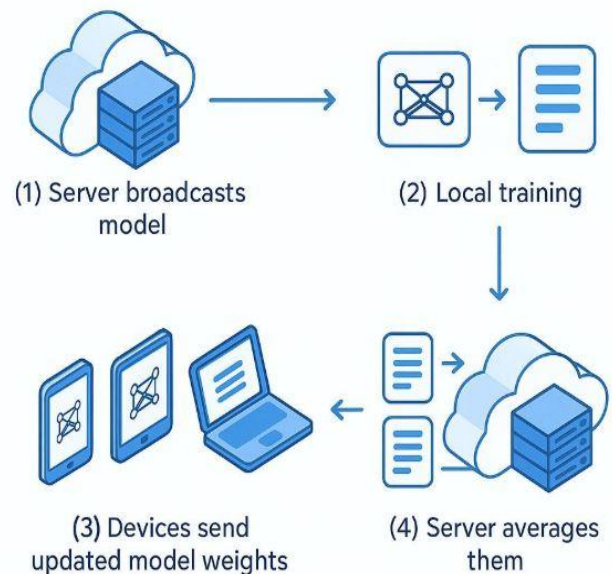


Figure 1. FedAvg aggregation process .

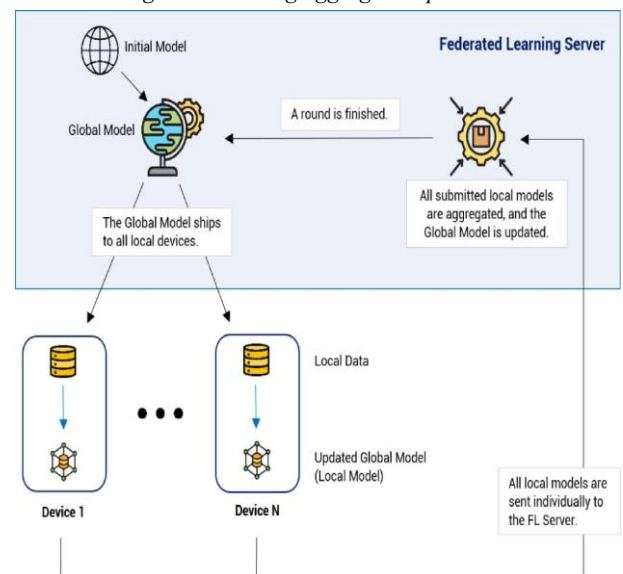


Figure 2. federated learning server.

3. METHODOLOGY

A. SYSTEM DESIGN METHODOLOGY

The completed system adopted Object-Oriented Analysis and Design (OOAD). OOAD was used to analyze, design, and model the federated dynamic access-control application by representing its major components as interacting modules, including authentication, behaviour collection, feature extraction, local training, federated aggregation, risk scoring, and access enforcement.

B. PROPOSED SYSTEM ARCHITECTURE:

The developed framework contains three participating institutions: University A, University B, and University C. Each institution has local users, local access logs, a local feature extraction module, a local machine learning model, and a local access decision engine. Raw access logs remain inside the institution where they are generated.

Each local model trains on its local access logs and sends only model parameters or model updates to the federated aggregator. The aggregator applies FedAvg to create a global model. The global model is then redistributed to all institutions so that each local system benefits from collective learning without exposing raw user activity data.

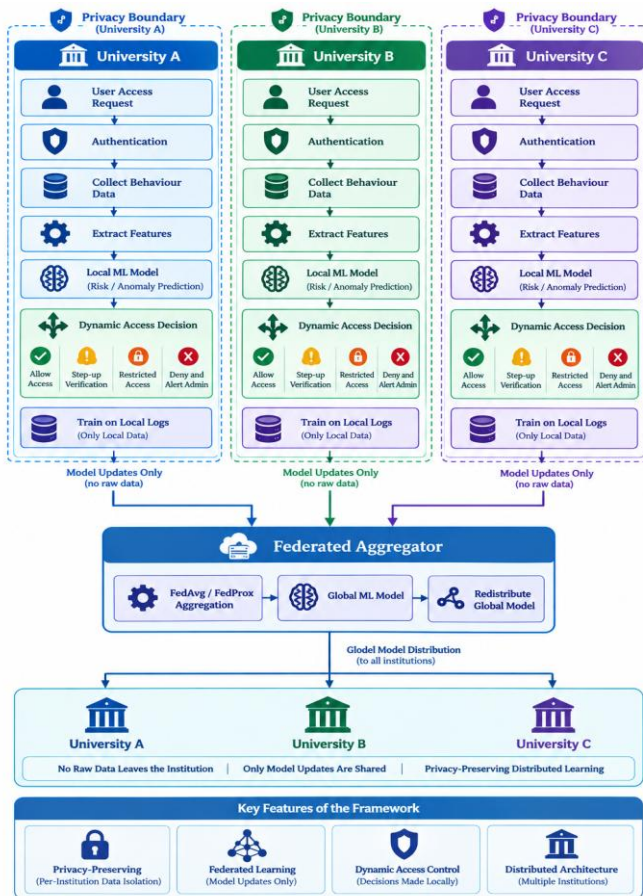


Figure 3. Proposed federated learning-based privacy-preserving dynamic access control architecture.

TABLE II. Main Components of the Implemented Framework

Component	Function	Privacy Role
Local access log	Stores access behaviour at each institution.	Kept inside the institution.
Feature extractor	Converts logs into model-ready vectors.	Runs locally.
Local ML model	Learns institution-specific access behaviour.	Trained on local data only.
Federated aggregator	Combines model updates.	Receives parameters, not logs.
Global model	Represents shared learned intelligence.	Redistributed to institutions.
Decision engine	Maps risk score to access decision.	Enforces decisions locally.

C. DATASET TYPE

The implemented system used synthetic distributed access-control logs. The dataset represents application-level user access events rather than raw network packets. Each institution maintained its own local dataset to support privacy-preserving distributed learning.

The dataset includes user role, institution, login time, device or browser information, IP/location risk, requested resource, action type, failed login attempts, request rate, session duration, risk label, and access decision. The institutional datasets are intentionally non-IID because access behaviour may differ from one university or branch to another.

TABLE III. Proposed Distributed Dataset Fields

Field	Example	Purpose
institution	University A	Identifies local training site.
role	Lecturer	Represents user responsibility.
login_hour	2 a.m.	Detects unusual time behaviour.
device_risk	Unknown device	Measures endpoint trust.
location_risk	Unfamiliar IP	Detects suspicious access origin.
resource	Student Records	Measures resource sensitivity.
action_type	Export	Measures operation risk.
failed_attempts	5	Captures authentication instability.
label	Risky	Used for supervised local training.

D. FEATURE EXTRACTION

Each local institution converts access logs into numerical feature vectors. Features include time deviation, device risk, location risk, resource sensitivity, action risk, failed attempts, request frequency, session age, role encoding, and institution encoding. These features support local risk prediction while keeping raw logs private.

E. LOCAL MACHINE LEARNING MODEL

The implemented local machine learning model used Logistic Regression because its parameters are simple to aggregate with FedAvg and its output probability can be directly converted into a risk score. The code structure also allows Random Forest, Neural Network, and FedProx extensions to be added later for comparative evaluation.

F. FEDERATED LEARNING ALGORITHM

FedAvg is used as the primary federated learning algorithm. In each round, every institution trains its local model on local data and sends model parameters to the aggregator. The aggregator averages those parameters and returns the global model to the institutions. FedProx is proposed as an extension for more heterogeneous distributed data.

TABLE IV. Federated Training Workflow

Step	Process	Data Shared
1	University A, B, and C train local models.	No raw data shared.
2	Local models produce parameter updates.	Model parameters only.
3	Federated aggregator applies FedAvg.	Aggregated parameters.
4	Global model is redistributed.	Global model only.
5	Institutions update local models.	Raw logs remain local.

3.1. DYNAMIC ACCESS DECISION MODEL

The local model predicts the probability that an access request is risky. This probability is converted into a risk score from 0 to 100. The decision engine combines

predicted risk with resource sensitivity and contextual risk to produce an access decision.

TABLE V. Proposed Risk-Based Access Decision Rule

Risk Score	Decision	Meaning
0-30	Allow Access	Normal behaviour and low-risk request.
31-55	Allow with Monitoring	Acceptable request but retained for closer observation.
56-75	Step-up Verification	User must complete extra verification.
76-90	Restricted Access	Only safe operations should be permitted.
91-100	Deny and Alert Admin	Critical risk event should be blocked and reported.

4. SYSTEM IMPLEMENTATION

The system was implemented as a local web application that simulates University A, University B, and University C. Each institution has separate users, access logs, local feature extraction, local model training, and local access decisions. A central federated dashboard shows training rounds, local model performance, global model performance, access decisions, model-update volume, and privacy indicators.

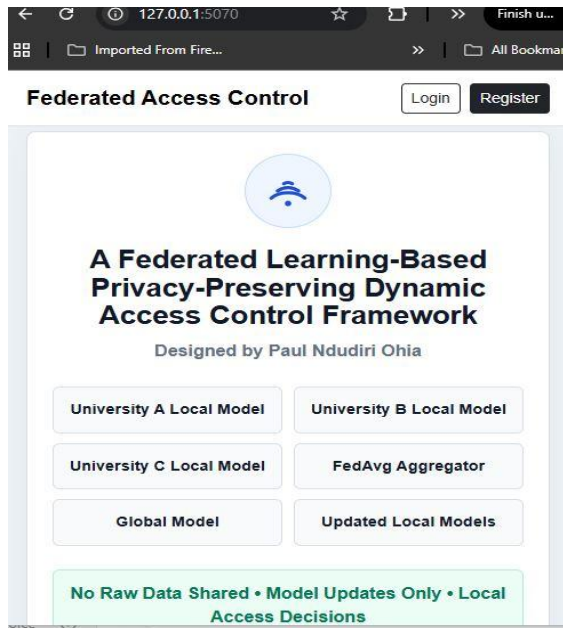


Figure 4. Home Page.

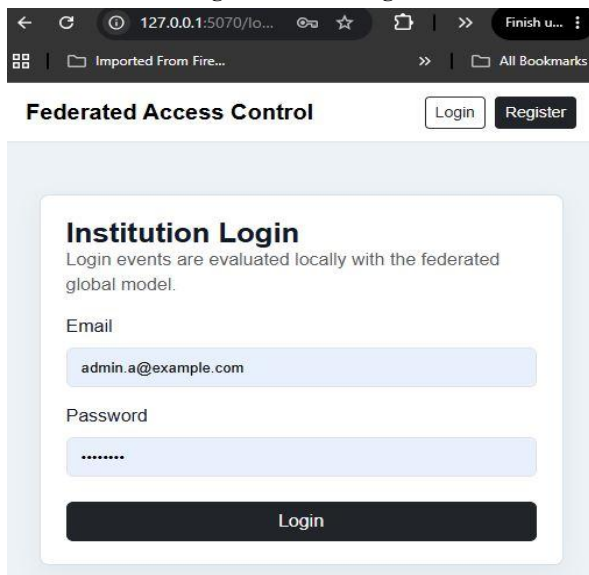


Figure 5. login and institution selection screen.

The completed implementation used Python Flask for the backend, SQLite for local storage, scikit-learn for Logistic Regression, HTML/CSS/Bootstrap-style layouts for the interface, and server-rendered charts for dashboard reporting. The federated aggregator simulated FedAvg by averaging local Logistic Regression parameters across institutions.

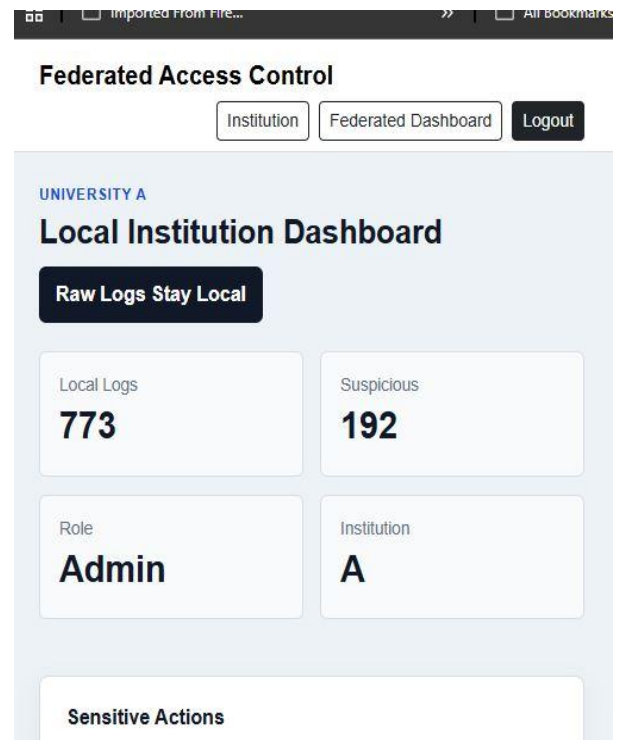


Figure 6. local institution dashboard.

Federated Access Control				
Institution Federated Dashboard Logout				
Local Model Updates				
Round	Institution	Samples	Local Accuracy	Update Status
6	University A	773	100.0%	Model update sent; raw logs retained locally
6	University B	707	100.0%	Model update sent; raw logs retained locally
6	University C	924	100.0%	Model update sent; raw logs retained locally
5	University A	772	96.8%	Model update sent; raw logs retained locally
University				Model update

Figure 7. local access logs and local model results.

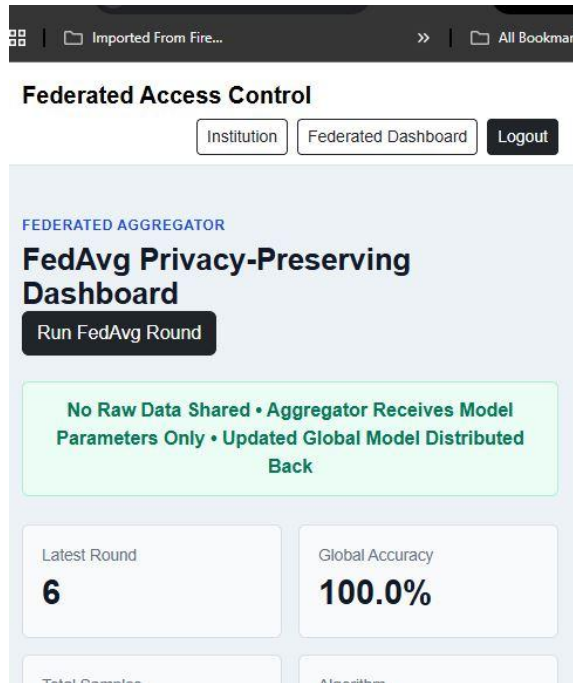


Figure 8. federated aggregator dashboard.

TABLE VI. Implemented System Modules

Module	Implemented Function	Output Produced
Authentication	Logs users into an institution.	Authenticated local user.
Behaviour collector	Captures access behaviour locally.	Local access event.
Feature extractor	Prepares model features.	Feature vector.
Local ML trainer	Trains institution-specific model.	Local model parameters.
Federated aggregator	Applies FedAvg.	Global model.
Risk engine	Converts probability to score.	Risk score and category.
Decision engine	Applies access policy.	Allow, verify, restrict, or deny.
Admin dashboard	Displays local and federated results.	Charts and audit logs.

The completed implementation generated a privacy-preserving federated access-control environment with three simulated institutions: University A, University B, and University C. The final demonstration database contained 2,404 access logs. The institutional distribution was intentionally non-identical to represent non-IID federated learning conditions: University A contributed 773 logs, University B contributed 707 logs, and University C contributed 924 logs. The system completed six FedAvg rounds and recorded 18 local model-update submissions across the three participating institutions.

TABLE VII. Implementation Dataset and Federated Training Summary

Metric	Value	Explanation
Participating institutions	3	University A, University B, and University C.
Seeded access logs	2,403	Distributed local access-control events before live login test.
Current access logs	2,404	Database count after one

		additional live login event.
University A logs	773	Local dataset retained within University A.
University B logs	707	Local dataset retained within University B.
University C logs	924	Local dataset retained within University C.
FedAvg rounds	6	Six federated aggregation rounds were completed.
Model updates	18	Three institutions submitted model updates in each of six rounds.

A. Local Model Performance Graph

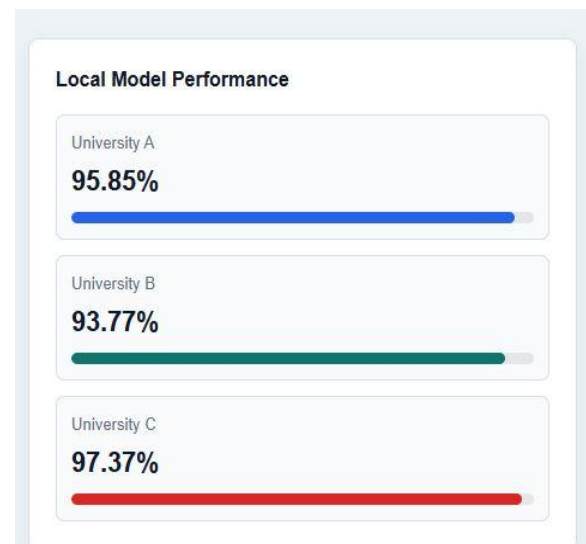


Figure 9. Local Model Performance graph.

The Local Model Performance graph compares the average local model performance of University A, University B, and University C. This graph is important because each institution trains on its own local access logs without sending raw data to the central aggregator. The graph therefore shows how well each local Logistic Regression model learns from its local behaviour patterns before and during federated training.

In the completed system, all institutions participated in the federated process, but their local datasets were not identical. The uneven local data volumes make the result more realistic because institutions in real distributed systems rarely produce the same number of access events. A strong local performance score indicates that the institution can support reliable local access decisions while still contributing to the global model.

B. Global Accuracy by Round Graph



Figure 10. Global Accuracy by Round graph.

The Global Accuracy by Round graph shows the performance of the federated global model after each FedAvg round. Each round represents a complete cycle in which the universities train local models, send model parameters to the aggregator, and receive an updated global model. This graph is useful for showing whether the global model remains stable or improves across federated learning rounds.

The completed system ran six FedAvg rounds. The global model improved from 91.4% in Round 1 to 96.1% in Round 6, showing stable high performance across the rounds. This result confirms that the FedAvg process combined local model knowledge into a usable global model without requiring direct sharing of institutional access logs.

C. Privacy Sharing Summary Graph

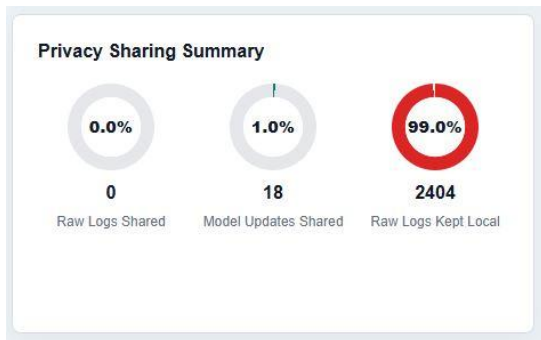


Figure 11. Privacy Sharing Summary graph.

The Privacy Sharing Summary graph demonstrates the central privacy argument of the implemented framework. It compares the number of raw logs shared with the number of model updates shared and the number of raw logs retained locally. The implemented behaviour kept raw logs shared at zero, while model updates were submitted to the federated aggregator.

This graph supports the privacy-preserving design of the system. The aggregator receives model updates only, not user activity logs. This means sensitive details such as login time, device fingerprint, IP/location, user role, requested resource, and action type remain inside the institution where they were generated. The graph therefore provides visual evidence that the framework follows the principle of local data retention.

C. Access Decisions Graph



Figure 12. Access Decisions graph.

The Access Decisions graph shows how the system distributed access requests into five decision categories: Allow Access, Allow with Monitoring, Step-up Verification, Restricted Access, and Deny and Alert Admin. The result distribution was: 1,402 Allow Access, 306 Allow with Monitoring, 348 Step-up Verification, 174 Restricted Access, and 174 Deny and Alert Admin after the live login test.

This graph demonstrates that the developed framework does not use a simple allow-or-deny security response. Instead, it supports graded access decisions. Normal low-risk behaviour is allowed, moderate behaviour is monitored, suspicious requests require step-up verification, high-risk actions are restricted, and critical requests are denied with an administrator alert. This makes the access-control response more flexible and more suitable for distributed systems.

D. Normal vs Suspicious Graph



Figure 13. Normal vs Suspicious graph

The Normal vs Suspicious graph summarizes the classification outcome of local and global risk prediction. The completed system produced 1,708 normal records and 696 suspicious records. This split shows that the dataset contains enough suspicious behaviour to make the dashboard visually meaningful while still keeping normal behaviour as the majority class.

The graph is useful in explaining the anomaly/risk detection behaviour of the system. Normal events include expected login times, known devices, familiar locations, and ordinary resource access. Suspicious events include unknown-device access, proxy or foreign network activity, late-night requests, repeated failed login attempts, sensitive export actions, and delete-record attempts.

F. Weighted Model Update Volume Graph



Figure 14. Weighted Model Update Volume graph.

The Weighted Model Update Volume graph shows the contribution size of each institution based on the number of samples used across federated rounds. Unlike a simple update-count chart, this graph is not equal for all universities because the local dataset sizes are different. The weighted update volumes were University A: 3,860, University B: 3,535, and University C: 4,620.

This graph strengthens the federated learning interpretation because it shows that institutions can contribute differently to the global model while still preserving privacy. University C contributed the largest weighted update volume because it had the largest local dataset. University B contributed the smallest volume because it had the smallest local dataset. This is closer to real distributed systems, where participating organizations rarely generate equal volumes of access-control data.

5. DISCUSSION OF RESULTS:

The results show that the developed framework supports privacy-preserving dynamic access control across distributed institutions. The completed system simulated three universities with separate local datasets, local behaviour logs, local model training, and federated aggregation through FedAvg. The uneven dataset sizes also demonstrate that the system can operate under non-IID conditions, where each institution has a different volume and pattern of access behaviour.

A key result is that raw access logs were not centralized. The central federated aggregator received model updates rather than direct user activity data. This is important because access logs may reveal sensitive information about staff, students, devices, login times, IP locations, and protected resources. The privacy sharing graph therefore confirms that the proposed model supports collaborative access-control intelligence without direct exposure of raw institutional records.

The access decision results also confirm the value of dynamic access control. Instead of producing only binary allow-or-deny outputs, the system produced five decision levels. This makes the framework suitable for real security environments where suspicious behaviour may require different responses depending on severity. For example, a moderately risky login can be monitored, while an abnormal delete attempt from an unfamiliar device can be denied and reported to an administrator.

The global model accuracy by round shows that FedAvg can produce a stable global model from local Logistic Regression models. Logistic Regression was used because its parameters are easy to aggregate and explain. Although the generated dataset made the model perform strongly, future real-world evaluation should use noisier access logs, larger institutional differences, and comparative models such as Random Forest, Neural Network, and FedProx-enhanced training.

5.1. Contribution to Knowledge:

The study developed a privacy-preserving dynamic access-control framework that combines local access decisions with federated learning.

- I.It demonstrates how distributed institutions can collaborate through model updates without sharing raw user activity logs.
- II.It provides a practical mapping from federated model output to access-control decisions such as allow, monitor, verify, restrict, and deny.

- III.It shows how non-IID institutional access logs can be simulated for federated access-control research.
- IV.It introduces weighted model update volume as a useful dashboard indicator for showing unequal institutional contribution to federated learning.
- V.It provides a completed implementation foundation for comparing FedAvg, FedProx, Logistic Regression, Random Forest, and Neural Network models in access-control security.

6. CONCLUSION:

This paper presented a completed Federated Learning-Based Privacy-Preserving Dynamic Access Control Framework for Distributed Systems. The framework enables institutions to train local models using local access logs while sharing only model updates with a federated aggregator. FedAvg was used to generate a global model that improved local access risk prediction without exposing sensitive behaviour data.

The developed approach is suitable for universities, hospitals, banks, government agencies, and distributed enterprises where privacy and security are both important. The work has implemented the web application, generated distributed local datasets, simulated FedAvg rounds, evaluated model performance, and documented the system through dashboard screenshots and result graphs. Future work can extend the completed system with FedProx, Random Forest, Neural Network comparison, real institutional logs, and production deployment.

REFERENCES:

- [1] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions," *Computer Communications*, vol. 195, pp. 346–361, 2022. <https://doi.org/10.1016/j.comcom.2022.09.012>
- [2] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. Bernal Bernabé, G. Baldini, and A. Skarmeta, "Evaluating Federated Learning for Intrusion Detection in Internet of Things: Review and Challenges," *Computer Networks*, vol. 203, article 108661, 2022. <https://doi.org/10.1016/j.comnet.2021.108661>
- [3] N. Hamdi, "Federated Learning-Based Intrusion Detection System for Internet of Things," *International Journal of Information Security*, vol. 22, pp. 1937–1948, 2023. <https://doi.org/10.1007/s10207-023-00727-6>
- [4] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Cyber Threat Intelligence Sharing Scheme Based on Federated Learning for Network Intrusion Detection," *Journal of Network and Systems Management*, vol. 31, article 3, 2023. <https://doi.org/10.1007/s10922-022-09691-3>
- [5] D. Javeed, M. S. Saeed, M. Adil, P. Kumar, and A. Jolfaei, "A Federated Learning-Based Zero Trust Intrusion Detection System for Internet of Things," *Ad Hoc Networks*, vol. 162, article 103540, 2024. <https://doi.org/10.1016/j.adhoc.2024.103540>
- [6] M. A. T. Ayedh, A. W. A. Wahab, and M. Y. I. Idris, "Enhanced Adaptable and Distributed Access Control Decision Making Model Based on Machine Learning for Policy Conflict Resolution in BYOD Environment," *Applied Sciences*, vol. 13, no. 12, 2023, p. 7102.