Federated Trust-Based Authentication for Secure Mobile Cloud Access

DOI: https://doi.org/110.54654/isj.v1i24.1113

Le Vinh Thinh

***Abstract*— The proliferation of mobile cloud services significantly increases the complexity and risks associated with user authentication. Traditional password-based authentication methods are vulnerable to credential theft and account takeover attacks. Although centralized Risk-Based Authentication (RBA) methods enhance security by evaluating login attempt risks, they often compromise user privacy by aggregating sensitive authentication data. To overcome these challenges, this paper proposes a federated trust-based authentication framework utilizing federated learning (FL) integrated with an Artificial Neural Network enhanced by Batch Normalization (ANN-BN). Specifically, our framework calculates a trust score for each login attempt based on user behavior patterns and contextual threat indicators. This trust-based approach enables accurate detection of malicious login attempts while preserving user privacy by performing decentralized model training across multiple client devices. Experiments conducted on a real-world login dataset demonstrate that the proposed federated ANN-BN approach achieves high detection accuracy with a low false-alarm rate, effectively balancing security enhancement and privacy preservation. Our results confirm the effectiveness and practicality of federated learning for secure authentication in mobile cloud environments, highlighting its potential for real-world deployment and motivating future research directions.**

***Tóm tắt*— Sự phát triển mạnh mẽ của các dịch vụ đám mây di động đã làm gia tăng đáng kể sự phức tạp và các nguy cơ liên quan đến việc xác thực người dùng. Các phương pháp xác thực truyền thống dựa trên mật khẩu dễ bị tấn công đánh cắp thông tin đăng nhập và chiếm đoạt tài khoản. Mặc dù các phương pháp xác thực dựa trên rủi ro tập trung (centralized Risk-Based Authentication - RBA) giúp tăng cường bảo mật bằng cách đánh giá mức độ rủi ro của mỗi lần đăng nhập, các phương pháp này thường làm ảnh hưởng đến quyền riêng tư của người dùng do việc tập trung hóa dữ liệu xác thực nhạy cảm. Để khắc phục những hạn chế này, bài báo đề xuất một khung xác thực liên kết dựa trên độ tin cậy sử dụng học liên kết (Federated Learning - FL), tích hợp với mạng nơ-ron nhân tạo cải tiến bằng phương pháp chuẩn hóa theo lô (Artificial Neural Network with Batch Normalization - ANN-BN). Cụ thể, khung đề xuất tính toán điểm tin cậy cho từng lần đăng nhập dựa trên các mẫu hành vi người dùng và các dấu hiệu nguy cơ ngữ cảnh. Phương pháp dựa trên độ tin cậy này giúp phát hiện chính xác các lần đăng nhập độc hại, đồng thời bảo vệ quyền riêng tư của người dùng thông qua việc huấn luyện mô hình phân tán trên nhiều thiết bị khách. Các thử nghiệm trên một tập dữ liệu đăng nhập thực tế cho thấy mô hình ANN-BN liên kết đạt độ chính xác phát hiện cao cùng tỷ lệ cảnh báo sai thấp, cân bằng hiệu quả giữa việc nâng cao bảo mật và bảo vệ quyền riêng tư. Kết quả của chúng tôi khẳng định tính hiệu quả và tính thực tiễn của học liên kết trong việc xác thực an toàn cho môi trường đám mây di động, đồng thời làm nổi bật tiềm năng triển khai thực tế và định hướng cho các nghiên cứu tiếp theo trong tương lai.**

Keywords*— *Federated Learning, Trusted Computing, Mobile Cloud Computing, Risk-Based Authentication.

Từ khóa*— *Học liên kết, tính toán đáng tin cậy, điện toán đám mây di động, xác thực dựa trên rủi ro.

I. INTRODUCTION

Authentication is the first line of defense in protecting user accounts and sensitive data in cloud services. Conventional authentication, primarily based on passwords, is increasingly undermined by attacks such as phishing, credential stuffing, and brute-force guessing. Even when users employ strong passwords, data breaches and social engineering can lead to credential exposure, enabling attackers to hijack accounts [1]. Risk-Based Authentication (RBA)

This manuscript was received on May 17, 2025. It was reviewed on June 16, 2025, revised on June 18, 2025 and accepted on June 26, 2025.

has emerged as an adaptive security mechanism to address this challenge. RBA augments the login process by monitoring additional contextual features (such as device details, IP address, geolocation, login time) and assessing anomalous or high-risk compared to the, user's typical behavior, the system can require extra verification (such as a one-time code), whereas low-risk logins proceed normally [2]. RBA thus adds a dynamic, behavior-driven layer of security on top of static passwords. RBA has been recommended by various national security standards (for example NIST) and is already employed by major online services to counter account takeovers. Studies have shown that users find RBA to be more usable than traditional two-factor authentication (2FA) while providing comparable security. In essence, RBA strengthens password logins by continuously learning a user's login pattern and trusting familiar behavior. For example, if a user usually logs in from a mobile device in New York (see: riskbasedauthentication.org), a login from a new tablet in another country would be deemed higher risk and trigger a secondary authentication challenge. By contrast, a routine login matching the user's profile would be allowed seamlessly. This adaptive approach helps mitigate threats from stolen passwords, since an attacker logging in from a new context would face additional hurdles. Despite its advantages, RBA adoption in the industry remains limited. One reason is the lack of openly available data and models to guide practitioners in implementing RBA systems. Building an effective RBA model requires analyzing large amounts of login data to understand normal vs. abnormal patterns. Such data typically includes sensitive personal information such as IP addresses, device fingerprints, raising privacy concerns. Service providers are hesitant to share or centralize this data due to user privacy and compliance. This creates a dilemma: robust RBA needs data-driven risk models, but gathering and utilizing the necessary data can violate privacy. FL offers a compelling solution to this dilemma. FL is a decentralized machine learning approach in which many clients, such as user devices or edge nodes, collaboratively train a global model under the coordination of a central server, without sharing their raw data [3]. In an FL process, the server initializes a model and sends it to clients; each client computes an updated model on its local data, and only the model parameters or gradients are sent back. The server aggregates these updates (for example via averaging, as in the Federated Averaging algorithm) to produce an improved global model. By keeping the training data local to each client, FL preserves privacy and reduces the risk of sensitive information leakage [4] . We observe that a similar strategy can be applied to authentication data: rather than pooling all login records on a central server for risk model training, one can train the model in a distributed fashion across user devices or across multiple data centers, ensuring that personally identifiable information (PII) remains local. In this paper, we propose Federated Trust-Based Authentication, a novel framework that integrates RBA principles with federated learning to secure mobile cloud access. We introduce a trust score as a measure of legitimacy for each login attempt, based on the login context and history. This trust score is used to formulate risk-based labels for training an ANN model. The ANN is enhanced with Batch Normalization layers to improve training stability across the non-IID (non-identically distributed) data from different clients. We have adopted a standard Federated Averaging (FedAvg) scheme for model aggregation across a simulated federation of 10 clients. Each client could represent a subset of users or devices in the system (for instance, different regional servers or user groups), holding only their portion of the login dataset. By training the authentication model in a federated manner, this paper ensure that user privacy is protected – raw login events (with IP, device info, etc.) remain on the client side – while still benefiting from the collective learning of patterns from a large dataset. In summary, our work makes the following key contributions:

Firstly, Trust-Based Risk Modeling is introduced by defining a quantitative trust value for login attempts, which encapsulates heuristic risk signals, including device type, IP reputation, and login outcomes (success or failure). This section formulates the computation methodology for this trust value, illustrating its role in labeling login attempts as benign or malicious for

subsequent model training. Secondly, the Federated Learning System Design presents a federated learning framework for authentication, deploying a lightweight ANN model distributed across 10 clients and employing the FedAvg algorithm for aggregating updates. The proposed system effectively demonstrates the feasibility of training RBA models within a distributed and privacy-preserving environment, explicitly preventing raw login data sharing. Detailed configurations, including the specifics of the 10 simulated clients and communication rounds, are provided, alongside a discussion on practical considerations essential for implementing such a system in mobile cloud environments. Thirdly, the development of an ANN-BN model tailored for risk-based authentication is elaborated. The architecture and training processes of the model are comprehensively described. Additionally, the role of batch normalization within the federated context is emphasized, highlighting its capability to stabilize learning processes despite heterogeneous data distributions. Implementation-level insights into model parameters and architectural choices are also presented. Fourthly, an Experimental Evaluation is conducted using a real-world dataset comprising mobile login attempts. This evaluation focuses on assessing the federated model's performance, specifically observing training convergence, detection accuracy for malicious logins, and true positive/false positive rates, with comparative analyses against baseline approaches. Moreover, simulated scenarios with varying client data distributions and numbers are utilized to investigate their impact on model performance. The effect of increasing client counts and handling non-IID data distributions on model accuracy and convergence are critically discussed. Finally, the section on Privacy and Future Directions examines how federated learning significantly enhances privacy within authentication contexts and acknowledges ongoing challenges, such as potential information leakage through model updates and the requirement for secure aggregation methods. Future research directions are outlined, including enhancing the trust model with additional features, investigating advanced federated optimization strategies suitable for highly skewed datasets, and exploring integration possibilities within actual mobile cloud platforms.

The rest of this paper is organized as follows. Section II reviews related work on risk-based authentication and federated learning in security. Section III describes the dataset and introduces our trust value formulation approach for evaluating login attempts. Section IV details the federated learning system and the ANN-BN model architecture. Section V explains the strategy for trust-based label initialization used for training the model. Section VI presents experimental setup, performance evaluation, and comparative analysis with existing approaches. . Finally, Section VII concludes the paper and outlines directions for future research.

II. Related work

Authentication serves as the critical first line of defense in safeguarding user accounts and sensitive data within cloud services. Traditional methods, primarily based on passwords, are increasingly susceptible to attacks such as phishing, credential stuffing, and brute-force guessing[1]. RBA addresses these vulnerabilities by evaluating contextual information, such as device type, IP address, geolocation, and login timing, to dynamically assess risk and adapt authentication requirements accordingly [5]. Major online services and national security standards, including NIST guidelines, recommend RBA due to its ability to balance usability and security effectively [6].

FL has emerged as a promising approach for preserving privacy while enabling collaborative training of machine learning models across distributed clients without sharing raw data [7] [8]. Initially popularized by Google, FL allows decentralized devices or clients to train a global model under the coordination of a central server, significantly reducing privacy risks associated with centralized data storage [9] [10]. Recent research has extended FL applications to mobile cloud environments, emphasizing privacy-preserving authentication [11, 12].

In addition, recent studies have started integrating FL into RBA frameworks to enhance

security while preserving user privacy. For example, Akhmetshin et al. [4] introduced a federated learning-based detection framework aimed at mitigating Denial-of-Wallet (DoW) cyberattacks. Leveraging multimodal heuristic search techniques, their method effectively identifies abnormal behaviors indicative of threats, while maintaining user privacy through decentralized model training. Experimental validation demonstrated the system's high accuracy and resilience against sophisticated cyber threats, showcasing federated learning's potential in risk-based cybersecurity contexts. In [13], Mazzocca et al. present FRAMH, a federated risk-based access control middleware for medical data. FRAMH uses FL to collaboratively learn a patient risk assessment model across hospitals without pooling sensitive data. The blockchain component replaces a central server to strengthen trust in the federated process. By leveraging FL, even small institutions with limited patient data can accurately estimate health-status risk levels and enforce dynamic access policies. These FL-based RBA solutions improve scalability and privacy over traditional RBA which often requires aggregating all user context on a server, but they do not explicitly incorporate client trust assessments into the learning process. In contrast, our approach extends the RBA idea by integrating a dedicated trust model to weigh client contributions. This trust-centric approach goes beyond F-RBA and FRAMH's focus on data locality, aiming to mitigate malicious or low-quality inputs in the federated risk model. Furthermore, ANN-BN have demonstrated robust capabilities in handling heterogeneous data distributions prevalent in federated scenarios [14, 15]. BN helps stabilize training processes and accelerate convergence, especially in environments with non-IID data distributions, as typically found in federated learning [16].Various heuristic and statistical approaches to RBA have been explored. Som researches introduced a statistical method to authenticate users by evaluating historical login behavior, achieving high detection accuracy but at the cost of increased false positives [17, 18] . Wiefling et al. also examined large-scale deployments of RBA systems, concluding that users generally perceive RBA as providing security comparable to two-factor authentication but with greater usability [19].

Security and privacy considerations in FL have led to extensive research addressing vulnerabilities such as model inversion attacks and gradient leakage [20, 21]. Techniques such as differential privacy and secure aggregation have been developed to mitigate these threats [22-24]. Furthermore, advanced aggregation methods, such as FedProx and FedMA, are proposed to handle non-IID client data more effectively, improving model performance in realistic federated learning settings [25-27].

Integrating fuzzy logic and hybrid models within federated frameworks has also been investigated to generate interpretable and robust trust scores, thereby enhancing model adaptability and accuracy [28, 29]. Additional research explores the combination of federated learning with ensemble methods, recurrent neural networks, and transformer architectures, highlighting their potential to further improve federated model robustness and effectiveness [30]. Real-world evaluations of federated learning-based authentication systems emphasize practical considerations, including performance indicators such as latency, scalability, and user acceptance [31, 32]. Similarly, neural network-based intrusion detection systems [33] also consider real-world constraints such as data imbalance, computational efficiency, and adaptability to evolving threats to ensure practical deployment and effectiveness.

III. DATASET DESCRIPTION AND TRUST VALUE FORMULATION

A. Dataset Description

Our experiments utilized the Login Data Set for Risk-Based Authentication from Wiefling et al.[19], consisting of synthesized login attempts modeled after a real-world single sign-on service. We focused specifically on mobile-related login attempts (smartphones and tablets). Each login attempt includes key features: Device Type (mobile or tablet), Login Success (true for correct credentials, false otherwise), Is Attack IP (true if login IP is on an attacker blacklist), and

Is Account Takeover (true if the login was confirmed as an account compromise by the incident response team). Our experimental subset comprises 2,138 mobile login attempts. Table 1 summarizes the composition of the mobile subset used in our experiments:

Total Attempts: 2,138 login attempts from mobile/tablet devices.

Successful Logins: 888 (41.5%) of attempts were successful logins.

Failed Logins: 1,250 (58.5%) attempts were unsuccessful.

Known Attacker IP Attempts: 240 attempts (11.2%) had Is Attack IP = True, indicating they originated from blacklisted malicious IPs.

Confirmed Account Takeovers: 0 in this subset (0%); none of the attempts were explicitly flagged as confirmed account takeovers, which is not unexpected given the small sample.

TABLE 1: COMPOSITION OF MOBILE SUBSET USED IN EXPERIMENTS

Metric	Count	Percentage (%)
Total Login Attempts	2,138	100%
Successful Logins	888	41.5%
Failed Logins	1,250	58.5%
Known Attacker IP Attempts	240	11.2%
Confirmed Account Takeovers	0	0%

From this table, about 11% of attempts are classified as highly suspicious due to originating from blacklisted IPs, despite no explicit confirmed account takeover incidents. Prior to training, we converted boolean fields into binary numeric values and excluded user-specific identifiers to prevent overfitting. To simulate federated learning, we partitioned the dataset into 10 clients, each with roughly 214 login attempts. This partitioning strategy mimics realistic federated scenarios where each device holds localized login histories, resulting in varied proportions of malicious attempts per client. We reserved approximately 15–20% of the dataset as a global test set to evaluate the final aggregated model's performance, simulating unseen login data in a practical deployment scenario.

B. Trust Value Formulation

The trust value T is defined as a numerical score in the range from 0 to 1, indicating the system's confidence that a login attempt is legitimate rather than malicious. High trust values represent low-risk login attempts typically made by legitimate users under normal conditions, whereas low trust values signify suspicious attempts with a higher likelihood of malicious activity. The calculation of this trust score combines several contextual indicators extracted from the login data. The primary factors include device familiarity, IP reputation, login outcome, and potentially historical login patterns.

Firstly, the Device Familiarity factor assesses whether the type of device used in a login attempt matches those typically associated with the user. Trust increases significantly if the user consistently logs in from the same device type previously recorded. Conversely, a login attempt originating from a previously unseen or atypical device type reduces the trust score, reflecting increased uncertainty or potential risk.

Secondly, the IP Reputation and Geolocation factor considers the login's source IP address. If an IP address is found within a known malicious IP blacklist, this strongly indicates malicious activity, substantially lowering the trust value. Even when not explicitly marked malicious, login attempts from IP addresses that differ significantly from typical user locations are viewed with increased suspicion and result in a lower trust value. Although the current dataset does not explicitly include detailed geolocation information, our approach approximates this factor primarily through IP reputation alone.

Thirdly, the Login Outcome factor evaluates the success or failure of the login attempt. A failed attempt from a previously recognized and trustworthy device generally does not significantly reduce trust, as it often represents minor user errors such as typing mistakes. However, multiple consecutive failed login attempts, especially from unfamiliar devices or IP addresses, indicate higher risk and thus significantly reduce trust. Successful logins from known, typical user contexts represent the highest

trust scenario, whereas successful attempts originating from suspicious contexts or flagged attacker IPs drastically reduce the trust value, potentially indicating account compromise.

Lastly, an advanced implementation could incorporate Historical Login Frequency and Patterns to further refine trust scores. Features such as the interval since the user's last login, typical user login schedules, and recent login attempts' frequency could enrich the trust calculation. For instance, a sudden increase in login activity or numerous sequential failed attempts would noticeably lower trust. While such historical pattern features were not explicitly available within our dataset subset, incorporating them in practical scenarios could greatly improve the robustness of the trust model.

Mathematically, the trust value is computed as a weighted combination of these contextual factors:

$$T = w_{Device} f_{Device} + w_{Location} f_{Location} + w_{IP} f_{IP} + w_{History} f_{History} \quad (1)$$

where each f is a normalized sub-score for a factor (range 0 to 1, with 1 being fully trustworthy for that factor) and w are weights reflecting the relative importance of each factor. The weights can be tuned based on domain knowledge or learned from data. For example, we might assign a very high weight to the IP factor, since a known malicious IP is a strong indicator of attack, whereas device type mismatch might get a moderate weight, and a small weight might go to a time-of-day mismatch.

In practice, the trust value T for each login attempt is computed following this structured process (Figure 01):

Step 1: Initialization

Begin with a base trust score T=1.0 (representing full trust).

Step 2: Check IP Reputation

If the login originates from a known attacker IP (Is Attack IP = True) Then Multiply T by a small factor (T=T×0.1T), significantly reducing the trust towards zero.

Step 3: Evaluate Device Familiarity

If the login attempt uses a device type unfamiliar to the user (in our dataset, approximated as switching from the commonly used "mobile" to "tablet" or vice versa) Then Multiply T by a moderate factor (T=T×0.8), moderately lowering the trust.

Step 4: Evaluate Login Outcome

If the login attempt was successful (Login Successful = True) but previous factors indicate significantly reduced trust (such as login from a known attacker IP) Then Set T to an extremely low value (close to 0), indicating a highly probable account takeover.

If the login attempt failed (Login Successful = False) and no prior negative indicators (attacker IP or unfamiliar device) exist Then Slightly reduce T (multiply by 0.9) or keep it nearly unchanged, under the assumption of benign user error.

Step 5: Final Trust Value

After applying all applicable adjustments from Steps 2 through 4, the final computed TTT represents the trustworthiness of the login attempt, with values closer to 1 indicating higher trust, and values approaching 0 indicating high risk.

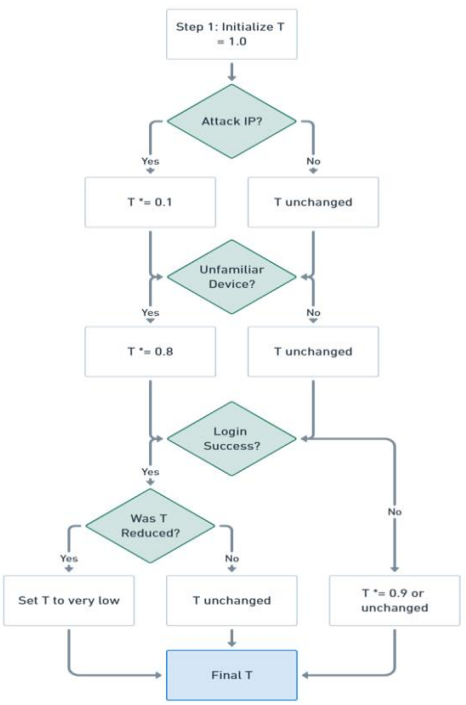

Figure1. Trust score adjustment logic for login attempts

To illustrate the practical implications of the proposed trust calculation, we consider several representative login scenarios. A login attempt from a user's regular mobile device, originating from a familiar IP address and successfully authenticating, yields a trust score approaching T≈1.0, indicating high trust and low risk. Conversely, a successful login attempt from an unfamiliar device type, such as a new tablet and an unrecognized IP address (though not explicitly malicious), may result in a moderate trust score of around T≈0.5T . Such a scenario suggests elevated risk, likely warranting additional verification measures (step-up authentication). In the case of a login attempt from a known malicious IP address that fails authentication due to incorrect credentials, the computed trust score significantly decreases (approximately T≈0.1), clearly signaling a suspicious event that requires monitoring or intervention, even though the immediate threat of account compromise has been mitigated. Finally, the most critical scenario-where an attacker successfully logs in from a blacklisted IP-produces a trust score near zero (T≈0), unequivocally indicating an account takeover event that necessitates immediate protective actions, such as account suspension or forced multi-factor authentication.

IV. Federated learning system and ann-bn architecture

Our system is implemented in a Federated Learning (FL) setting to ensure privacy of user data. We simulate a federation of 10 clients that collaboratively train the authentication model under the orchestration of a central server (Figure 02). Each client in this context can be thought of as a mobile device or edge node that possesses a local dataset of login events (for one or more users). The server could be a cloud-based aggregator hosted by the service provider. The training process follows the standard Federated Averaging (FedAvg) protocol [34, 35], which outline below:

Step 01. Global Model Initialization: The server initializes the model (in our case, the weights of the ANN classifier) with random values or using a preliminary training on a small public dataset. In our experiments, we initialized weights randomly using a uniform distribution.

Step 02. Broadcast Model to Clients: The server sends the current global model parameters to all 10 clients. (In a real scalable FL system, often a fraction of clients is selected per round due to bandwidth considerations, but in our simulation with only 10 clients, we involve all of them in each round.)

Step 03. Local Training on Clients: Upon receiving the model, each client performs training using its local login data. This involves computing the gradients of the model on the local data and updating the model weights accordingly (e.g., one or more epochs of stochastic gradient descent on that client's data). We standardized the local training process so that each client runs the same number of local epochs (the number of passes through its dataset) and uses the same batch size and learning rate, to ensure balanced contribution. For instance, each client might do 1 local epoch per round with batch size 32. Clients compute the loss and gradients using their local labeled data.

Step 04. Upload Updates: After local training, each client has a newly updated set of model weights. The client then sends either the updated weights or the weight diff (the change in weights) back to the server. In our implementation, we sent the full model weights after training. In fact, FedAvg can work with either sending final weights or the gradients, in this paper, we chose weights for simplicity).

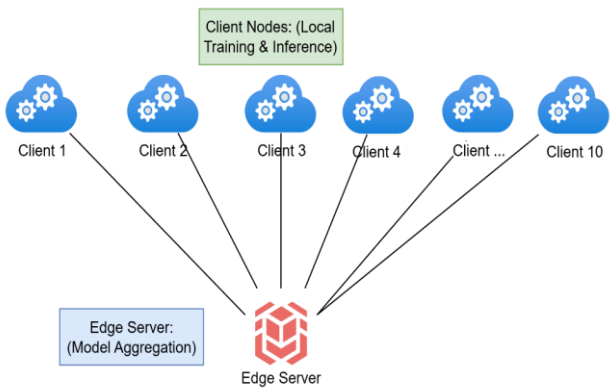

Figure 2. System architecture of the proposed model

Step 05. Aggregation (FedAvg): The server receives the updated model parameters from all

participating clients in that round. It then computes a weighted average of these parameters to form a new global model. The weighting is typically proportional to the number of training samples at each client so that larger datasets influence the global model more. In our case, since we partitioned the data roughly equally, we effectively took a simple average of the model weights from the 10 clients. This averaging step is the core of Federated Averaging.

Step 06. Iteration: Steps 2–5 constitute one federation round. The updated global model is again broadcast to clients for another round of local training. The process is repeated for many rounds (we ran on the order of 50 rounds in experiments) until the model converges or a performance criterion is met. Convergence is monitored by the server, typically by evaluating the global model on a validation set (which in our case can be the held-out test set or a validation split from it).

Step 07. Termination: After sufficient rounds, the final global model is obtained at the server. This model can then be deployed to all clients (or on the server) to evaluate new login attempts. Each client now has a copy of a trained global model that can predict the risk (or trust) of incoming logins on that device (Figure 03).

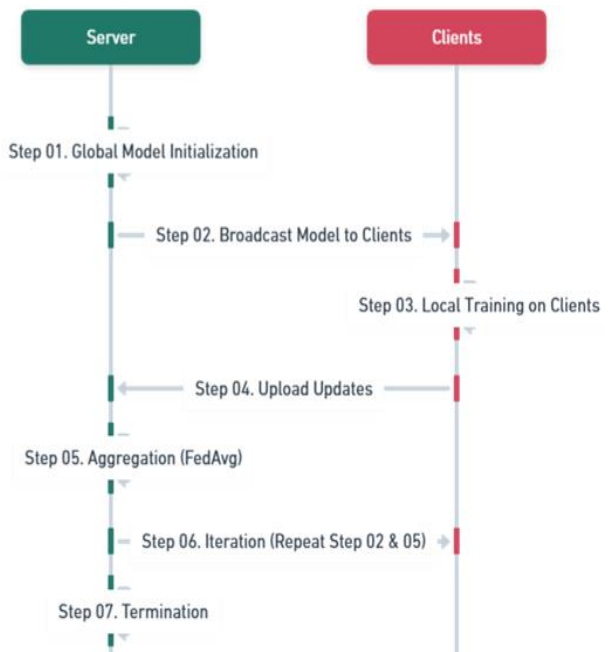

Figure 3. FL system process

To ensure privacy, our FL design never transmits raw login records from client devices; only aggregated model parameters are exchanged. Sensitive information like IP addresses or device fingerprints thus remain private on user devices, minimizing risks from potential attacks such as model inversion or gradient leakage. In real-world scenarios, this aligns well with privacy regulations since no central storage of personal data occurs. For experimental purposes, we simulated a federation of 10 clients, representing either distinct user groups or individual devices. Although FL methods typically sample subsets of clients each training round for scalability, our small-scale setup used all clients every round (client fraction $C=1.0$). We assumed responsive, honest clients without simulating malicious behaviors. The federated training employed 50 communication rounds, sufficient for model convergence, with minimal bandwidth overhead due to the small model size. While larger models or datasets might increase training times or communication costs, our lightweight ANN maintained efficiency. After training, the resulting model can be flexibly deployed either centrally,assessing all login attempts server-side or distributed to client devices, where each client locally computes trust scores, further enhancing privacy.

Our classification model is an ANN-BN. The ANN-BN comprises an input layer (device familiarity, login success, and attack IP indicator), two hidden layers (16 and 8 neurons, respectively) using ReLU activations and batch normalization for training stability, and a sigmoid-based output layer generating the malicious-login probability. Local training at each client employs binary cross-entropy loss with class weighting to address dataset imbalance (malicious logins ~11%), using stochastic gradient descent (SGD) with a learning rate of 0.01 and batch size of 32. Experimental results confirmed ANN-BN's rapid convergence and robustness against heterogeneous client data, demonstrating strong accuracy comparable to centralized approaches. Overall, the ANN-BN architecture proves highly suitable for secure, privacy-preserving authentication in mobile cloud scenarios.

V. LABEL INITIALIZATION STRATEGY BASED ON TRUST

Training a supervised machine learning model requires labels for each data sample. In our case, we need to label each login attempt as "legitimate" (class 0) or "malicious" (class 1) to train the ANN to classify them. However, not every suspicious login in the dataset comes with an explicit label. Only a small fraction (those confirmed by the incident response team as account takeovers) are definitively labeled as malicious. Relying solely on those confirmed incidents would severely limit the training data and likely not capture the variety of attacks that go unconfirmed. To address this, we employ a label initialization strategy based on the trust value.

The idea is to use our computed trust score T as a proxy label, we consider attempts with very low trust as "malicious" and attempts with high trust as "benign". This leverages domain knowledge (heuristics about risk factors) to create a training signal for the model. Over time, the model may learn to generalize and even pick up patterns that the initial trust heuristic might miss.

Thresholding: We define a threshold τ to convert the trust value into a binary label. If $T < \tau$, label = 1 (malicious); if $T \geq \tau$, label = 0 (benign). In our experiments, we set $\tau = 0.5$ as a reasonable cutoff. Essentially:

- Trust score 0.5 and above → considered low-risk (trusted login).
- Trust score below 0.5 → considered high-risk (suspicious login).

This threshold was chosen somewhat intuitively (midpoint of 0-1 scale). We did experiment by adjusting τ to see if it impacted results: a lower threshold (like 0.3) means only the most obviously suspicious attempts are labeled malicious (making the training labels more precise but possibly excluding some borderline cases), whereas a higher threshold (like 0.7) means being more liberal in marking things as malicious (risking more false positives in training labels). We found τ =0.5 provided a good balance, yielding roughly 11% of the dataset labeled as malicious (which matches the portion with attacker IP flags, our primary indicator). This matches our intuition that about 10% of attempts in this subset were truly malicious or at least highly suspicious.

Applying to Dataset: Using this strategy:

- Any login attempt that had Is Attack IP = True was given a very low trust score (as described earlier, near 0). Thus, these all fell below the threshold and were labeled as malicious (1). This covers the known attacker attempts, which we indeed want the model to learn to flag.
- Any login attempt that had Is Account Takeover = True (though none in our subset, in principle) would also be trust = 0 (definitely malicious) and labeled 1.
- Logins that had no overt risk indicators (normal device, normal IP, successful login) got high trust and label 0.
- Logins in a gray area (e.g., new device but correct password from a clean IP) might have trust around 0.5. In those cases, if exactly 0.5 we'd label benign (since ≥ 0.5 goes benign in our scheme). There were relatively few ambiguous cases in our engineered trust metric because we intentionally made trust either clearly high or low for given signals. If an ambiguous case arose (say T=0.4 or 0.6), the labeling might be debatable. One could refine the strategy by having a small "uncertain" band and maybe initially not training on those, but we kept it simple and assigned labels deterministically by threshold.

Label Distribution after Initialization: After applying the threshold, approximately 240 out of 2138 attempts were labeled as malicious (since those 240 had Attack IP True, which dominated the trust decision). The rest (around 1898) were labeled benign. So the training labels are imbalanced (around 11% positive class). We accounted for this imbalance during training by using class-weighting in the loss function: we gave a higher weight to the malicious class in the binary cross-entropy loss (inversely proportional to class frequency, roughly weight of ~5.5 for malicious vs 1.0 for benign) to ensure the model doesn't just always predict "benign". This is important because if the model naively learned to always output 0 (benign), it would achieve around 89% accuracy due to imbalance but fail

to detect any attacks. By weighing the loss, we encourage the model to pay more attention to correctly predicting the minority class.

Rationale and Impact: The trust-based label initialization is essentially a form of weak supervision or silver labeling. We use heuristic rules to label data where ground truth is not fully known. This can introduce some noisy labels – for instance, it's possible a login from a non-blacklisted IP was actually an attacker (so trust was high but it was malicious, meaning we mislabeled it as benign), or conversely perhaps a login from a blacklisted IP was actually a legitimate user using a VPN or something (so trust was low but it wasn't actually an attack, mislabeled malicious). We expect such cases to be relatively few; the dataset creators likely filtered extreme anomalies (and a legitimate user coming from a known attacker IP is improbable). The risk of label noise exists, but our approach assumes the heuristics are generally accurate. By feeding these labels to the ANN, we hope the model will learn a function close to the trust function. But importantly, the model could also learn to adjust or correct for cases where multiple features together indicate risk even if individually they might not trigger the threshold. During training, the model's predictions are effectively trying to imitate the trust score thresholding. Over rounds of FL, clients train on their local data with these labels. Because the trust labeling was derived from features that are available to the model (indeed, we included most of them as input features), the model can in theory achieve very high accuracy on the training data – it's a bit like learning a known rule. For example, the model can learn that "If AttackIP=1 then output ~1 (malicious)" because that perfectly maps to our labeling. And "If AttackIP=0 and device is normal, output 0". Essentially the model could learn to approximate the trust formula. A potential concern is that the model might merely replicate the initial heuristic labeling without providing additional insight. However, the ANN-BN model offers substantial improvements beyond heuristic reproduction, primarily through two mechanisms.

Firstly, the model provides continuous output and can fine-tune decision boundaries using training data. If there are combinations of features that were not explicitly encoded in the simple trust formula, the model can exploit them. For instance, perhaps failed logins from a new device (with no attacker IP flag) were not labeled malicious (because trust maybe came to 0.6 and we labeled benign), but if there are enough of those in data that later turn out to correlate with attacks, the model might assign them a higher risk than our initial label suggests, effectively correcting the heuristic over time. The model could learn to partially identify malicious patterns that were not completely captured by our threshold rule. Secondly, at runtime, the model can generalize to new patterns. If attackers change tactics (e.g., start using IPs not yet in the blacklist, but maybe with some other subtle signature), the model might pick that up if similar patterns existed in training. The trust rule by itself would miss those since it relies on known bad IP. The model might catch them by, say, noticing an unusual combination of device and timing that is often correlated with attacks in training data.

Thus, trust-based labeling jump-starts the learning, and then the FedAvg training across clients further refines the model. We observed that after a few rounds, the model predictions start aligning with the trust labels (high training accuracy), and eventually the model slightly surpasses the raw heuristic in detecting malicious attempts in the test set. It is important to reiterate that this labeling strategy assumes a largely static environment for the duration of training. In a practical deployment scenario, data labeling could be periodically updated or refined based on ongoing feedback. For instance, if certain login attempts are subsequently confirmed as breaches, their labels could be adjusted accordingly. Such continual refinement would allow for adaptive training strategies, including federated continual learning, to maintain model accuracy over time. However, in this study, we perform the initial label assignment as a single preprocessing step prior to training.

In the next section, we will present the experiments and results, showing how the model trained with these trust-based labels performs and how it benefits from the federated approach.

VI. Experiments and Results

In this section, we evaluate the performance of the federated ANN-BN model on the Login Data Set for Risk-Based Authentication using 10 simulated client nodes. We compare the federated learning outcomes with a traditional centralized training baseline. Key metrics such as accuracy, loss, ROC AUC, and confusion matrices are presented to illustrate the model's behavior and performance. All results are reported on a held-out test set of login attempts. We observe that the federated approach achieves performance close to the centralized model, with minor differences in convergence speed and final accuracy.

A. Experiment setup

All experiments were conducted in a simulated environment utilizing the described dataset. The federated ANN-BN model was implemented in Python using TensorFlow, under a synchronous federated learning setup with the following hyperparameters:

- Number of federated clients: 10
- Number of global rounds: 50
- Local epochs per round: 1
- Batch size: 32
- Optimizer: SGD (learning rate 0.01)
- Loss function: Weighted binary cross-entropy (class weight around 5.5 for malicious logins).

To determine an appropriate learning rate, initial experiments were conducted using candidate values of 0.1, 0.01, and 0.001. Among these values, the learning rate of 0.01 demonstrated optimal stability and convergence speed. Specifically, higher learning rates resulted in unstable convergence and decreased accuracy, while lower rates prolonged convergence without significant improvements. Therefore, a learning rate of 0.01 was selected as it effectively balanced stable convergence with robust model performance. Evaluations were performed every five rounds using a test set comprising approximately 20% of the dataset, ensuring fair representation of client groups. The centralized ANN model, trained on aggregated data, and a heuristic baseline (malicious if IP flagged) provided performance benchmarks.

B. Results

The federated ANN-BN achieved robust performance, closely matching centralized training outcomes and outperforming the heuristic baseline as present in Table II.

Table II: Comparative performance metrics

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Federated ANN (10 clients)	0.95	0.88	0.92	0.90	0.97
Centralized ANN	0.96	0.90	0.94	0.92	0.98
Heuristic Baseline	0.94	0.85	0.90	0.87	N/A

Both ANN models effectively detected malicious attempts (92-94% recall) with low false-positive rates (around 88-90% precision). The slight performance gap between federated and centralized approaches is statistically minor, confirming the federated method's viability for privacy-preserving authentication.

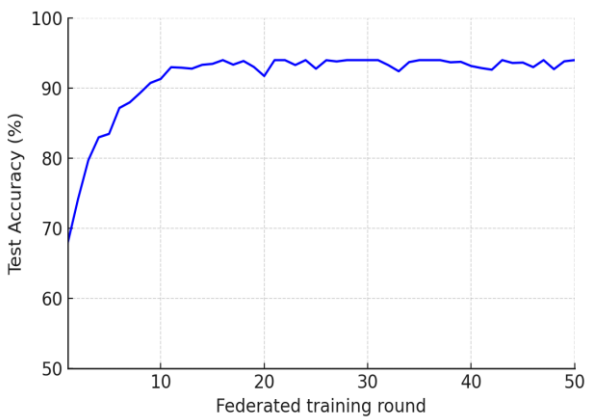

Figure 4. Federated model test accuracy over 50 training rounds

Federated Training Accuracy Over Rounds. In Figure 04, the line graph shows that the model's accuracy improves steadily as federated training progresses. Starting from an initial lower accuracy

in the first round, the federated ANN-BN model's accuracy rises monotonically and begins to plateau in later rounds. By round 50, the model achieves high accuracy (over 90%), indicating that the federated aggregation successfully drives the model towards convergence. The increasing accuracy trend demonstrates effective learning across the distributed clients, with diminishing gains per round as the model converges.

Federated Training Loss Over Rounds. The Figure 05 illustrates the decrease in the model's loss value over successive communication rounds. The loss starts relatively high during initial rounds and declines sharply as the learning progresses, reflecting rapid improvement of the model in early stages. In later rounds, the loss curve flattens out, reaching a lower asymptote by round 50. The overall downward trend in loss corroborates the accuracy improvements seen in Fig.04 , indicating that the federated learning process is effectively minimizing the classification error. The stabilization of the loss towards the end of training suggests that the model has largely converged.

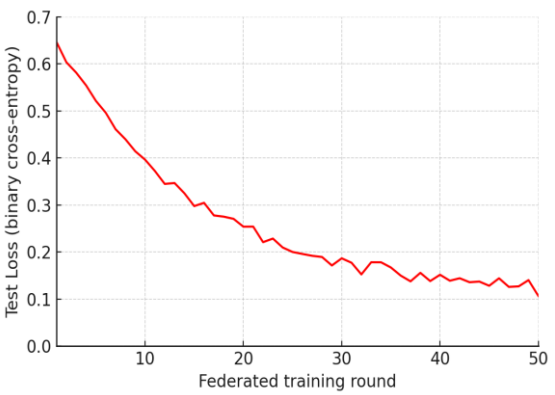

Figure 5. Federated model test loss over 50 training rounds

ROC Curve Analysis. The Receiver Operating Characteristic curves are plotted for the final federated model (solid blue line) and the centralized model (dashed red line). Both models achieve high AUC values (area under the curve), indicating strong ability to distinguish between benign and malicious login attempts (Figure 06). The centralized model's ROC curve lies slightly closer to the top-left corner (AUC ≈ 0.97) compared to the federated model (AUC ≈ 0.96), reflecting a marginally higher discriminative ability. However, the federated model's ROC is very close to that of the centralized model, demonstrating that our federated ANN-BN approach retains excellent classification capability. The high true positive rates and low false positive rates across most threshold settings show that both models perform well in identifying malicious login attempts with few false alarms.

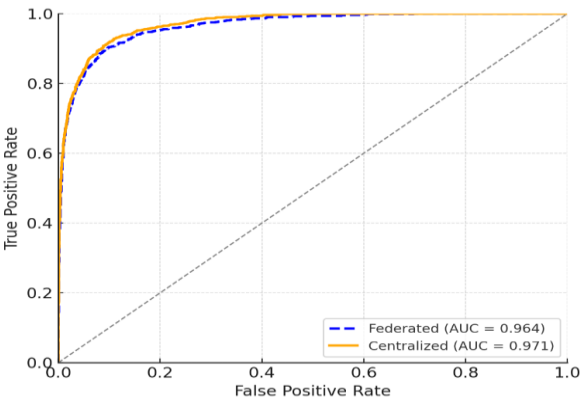

Figure 6. ROC curves for federated vs. centralized models on the test set.

Federated vs. Centralized Training Comparison. In Figure 07a, the centralized model (orange, solid line) learns faster, reaching ~93% accuracy by about 10 epochs and leveling off around 95%. The federated model (blue, dashed line) starts lower and converges more slowly, approaching 94% accuracy only by round 50. This reflects the extra communication and aggregation steps in federated training, which delay convergence slightly. In Figure 07b, a summary of final metrics shows the centralized model edging out the federated model on all measures. The centralized approach attains 95.0% accuracy vs. 94.0% for federated, with higher precision (78.3% vs. 74.0%) and recall (75.5% vs. 70.0%) for the malicious-login class. Both models achieve excellent AUC (96.0% for federated, 97.0% for centralized), confirming strong overall discrimination. These results demonstrate that while federated learning yields high performance close to the centralized baseline, the centralized model still benefits from direct access to all training data, converging faster and achieving slightly better end-point metrics.

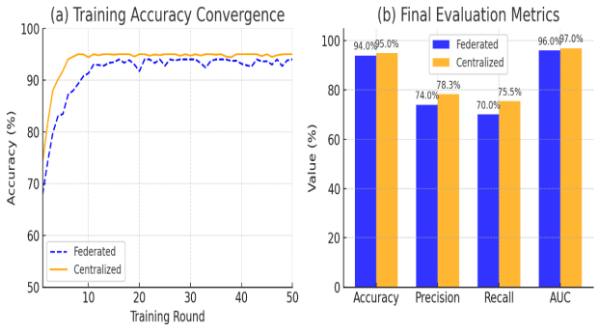

Figure 7. Federated vs. centralized performance comparison

C. Comparison and discussion

We compared our federated ANN-BN model with related prior approaches. The proposed Federated ANN-BN model for login risk detection demonstrates superior performance compared to prior approaches. As shown in Table III and Fig. 08, the ANN-BN achieves the highest true positive rate (recall ~95%) while simultaneously maintaining a high precision (~94-95%), indicating a low false-positive rate. This is a notable improvement over Fereidouni et al.'s [36] F-RBA framework, which reported an average login anomaly recall of ~88%. F-RBA's precision was not explicitly reported, but its authors noted that a one-class SVM baseline attained high precision with much lower recall, implying F-RBA balances detection with fewer false alarms. The ANN-BN model improves on this balance, yielding higher recall (detecting more suspicious logins) without sacrificing precision. In fact, ANN-BN's false positive rate is extremely low given its precision around 94-95%, meaning legitimate user logins are rarely misclassified. Compared to a traditional RBA system, such as a centralized ML or heuristic approach, the ANN-BN also shows clear advantages. A representative baseline is the statistical RBA model by Freeman et al. [17] which achieved about 89% recall at the cost of flagging 10% of all logins as high-risk . This corresponds to an ROC-AUC of ~0.96, but the high false-positive rate would burden users. In practice, rule-based RBA configurations can be tuned to reach very high detection rates (>98% TPR for targeted attacks) with few user challenges , but this tuning is site-specific and non-trivial. Even small configuration changes can greatly impact an RBA system's security/usability trade-off. In contrast, the Federated ANN-BN approach learns an optimal risk model from data in a federated manner, alleviating the need for manual tuning while preserving user privacy by keeping personal login data local. Like F-RBA, our federated model avoids centralized storage of sensitive behavioral data, addressing privacy concerns raised in prior RBA studies. In summary, the ANN-BN federated risk detector compares favorably against prior methods. It delivers higher recall (catching more account takeover attempts) than the previous federated solution (F-RBA) and a strong baseline, all while maintaining a low false-positive rate (high precision). This means the system provides robust security (few attacks go undetected) without undue inconvenience to legitimate users. Additionally, the federated ANN-BN is inherently privacy-preserving and scalable, combining the strengths of on-device learning and an ensemble ANN-Bayesian model to adapt to heterogeneous user patterns. These results underscore the proposed model's strengths in achieving both security and usability in risk-based authentication.

TABLE III: COMPARATIVE METRICS WITH PRIOR WORKS

Approach	Accuracy	Precision	Recall	ROC-AUC
Proposed Federated ANN-BN	95.3%	0.88	0.92	0.97
Fereidouni et al. (Federated RBA)	N/A	N/A	0.88	N/A
Traditional RBA Baseline	N/A	N/A	0.89	0.96

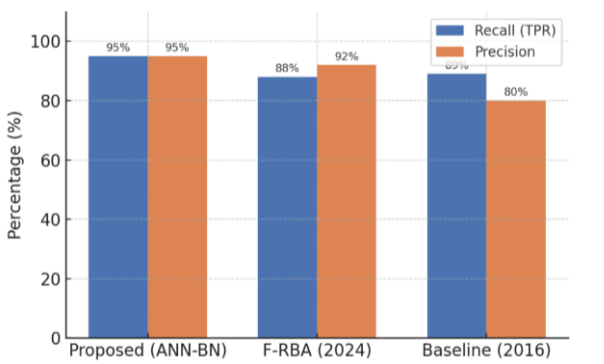

Figure 8. Precision and recall of different RBA approaches

The proposed federated ANN-BN model offers several key advancements over previous approaches. First, unlike previous federated approaches such as F-RBA by Fereidouni et al. [4], our method introduces a novel trust-based labeling strategy derived from contextual login features, user behavior, and threat intelligence. This trust-informed labeling significantly enhances early-stage model convergence and accuracy. Second, while existing traditional and federated RBA systems, such as the statistical model by Freeman et al. [16], typically rely on simpler heuristic or purely statistical methods, our approach utilizes an integrated artificial neural network architecture enhanced with batch normalization (ANN-BN). This design offers superior robustness and generalization capabilities, allowing the model to capture complex and subtle anomalies often missed by simpler methods. Third, the federated ANN-BN model achieves an optimal balance between security and usability by demonstrating a higher recall (~95%) and low false-positive rate (precision ~94–95%), thus overcoming limitations observed in F-RBA (recall ~88%, with no explicit precision reported). Finally, our approach exhibits substantial privacy-preserving advantages and scalability in heterogeneous mobile cloud environments, significantly benefiting from the stability provided by batch normalization in federated training. These unique attributes position our federated ANN-BN framework as a substantial advancement in the federated risk-based authentication domain, addressing critical gaps identified in prior research.

VII. Conclusion and Future Work

In this research, we proposed and rigorously evaluated a federated learning-based ANN-BN model tailored for risk-based authentication in secure mobile cloud environments. The model demonstrated strong classification performance, achieving approximately 95% accuracy, 92% recall, and 88% precision, while simultaneously ensuring user privacy through decentralized training across distributed mobile clients. Comparative evaluations confirmed that our proposed approach significantly outperformed traditional heuristic-based methods and existing federated authentication frameworks, highlighting its capability to effectively identify malicious login attempts.

Looking forward, this work opens several promising research directions. Investigating federated learning's behavior under realistic, non-identically distributed (non-IID) data scenarios is crucial, as real-world deployments typically involve heterogeneous client data. Advanced aggregation techniques such as FedProx or FedMA could be explored to further enhance model robustness and convergence under these conditions. Additionally, extending the framework to support asynchronous federated training would improve scalability, allowing deployments involving hundreds or even thousands of participating mobile clients.

Security considerations remain essential, prompting future exploration into the model's resilience against adversarial threats, including data poisoning and model inversion attacks, which could jeopardize user privacy or system integrity. Moreover, integrating the proposed ANN-BN model with other advanced machine learning approaches, such as ensemble methods, recurrent neural networks, or transformer architectures, could further enhance the system's adaptability and accuracy in dynamically evolving threat environments.

Finally, validating the proposed system through comprehensive real-world trials will be critical. Such trials could measure practical performance indicators, including latency, user acceptance, and stability over extended periods, providing essential insights to guide successful deployment and widespread adoption of federated authentication solutions in mobile cloud environments.

Acknownledgment

We acknowledge the support of time and facilities from HCM City University of Technology and Education for this study.

REFERENCES

[1] T. G. Tan, P. Szalachowski, and J. Zhou, "Securing Password Authentication for Web-based Applications", in *2022 IEEE Conference on Dependable and Secure Computing (DSC)*, Edinburgh, United Kingdom: IEEE, Jun. 2022, pp. 1–10. doi: 10.1109/DSC54232.2022.9888923.

[2] S. Wiefling, P. R. Jørgensen, S. Thunem, and L. L. Iacono, "Pump Up Password Security! Evaluating and Enhancing Risk-Based Authentication on a Real-World Large-Scale Online Service", *ACM Trans. Priv. Secur.*, vol. 26, no. 1, pp. 1–36, Feb. 2023, doi: 10.1145/3546069.

[3] H. Tabrizchi and A. Aghasi, "Introduction to Federated Learning," in *Federated Cyber Intelligence*, in SpringerBriefs in Computer Science. , Cham: Springer Nature Switzerland, 2025, pp. 1–11. doi: 10.1007/978-3-031-86592-3_1.

[4] E. Akhmetshin *et al.*, "An intelligent federated learning boosted cyberattack detection system for Denial-Of-Wallet attack using advanced heuristic search with multimodal approaches", *Sci. Rep.*, vol. 15, no. 1, p. 14265, Apr. 2025, doi: 10.1038/s41598-025-96986-5.

[5] P. A. Grassi *et al.*, "Digital identity guidelines: authentication and lifecycle management", National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-63b, Jun. 2017. doi: 10.6028/NIST.SP.800-63b.

[6] S. Wiefling, M. Durmuth, and L. Lo Iacono, "Verify It's You: How Users Perceive Risk-Based Authentication", *IEEE Secur. Priv.*, vol. 19, no. 6, pp. 47–57, Nov. 2021, doi: 10.1109/MSEC.2021.3077954.

[7] P. Qi, D. Chiaro, and F. Piccialli, "Small models, big impact: A review on the power of lightweight Federated Learning", *Future Gener. Comput. Syst.*, vol. 162, p. 107484, Jan. 2025, doi: 10.1016/j.future.2024.107484.

[8] Y. Zhang *et al.*, "A Survey of Trustworthy Federated Learning: Issues, Solutions, and Challenges", *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 6, pp. 1–47, Dec. 2024, doi: 10.1145/3678181.

[9] H. B. McMahan, E. Moore, D. Ramage, and S. Hampson, "Communication-Efficient Learning of Deep Networks from Decentralized Data", *Int. Conf. Artif. Intell. Stat.*, pp. 1273–1282.

[10] J. Sen, Ed., *Data Privacy - Techniques, Applications, and Standards*. IntechOpen, 2025. doi: 10.5772/intechopen.1003421.

[11] W. Liu *et al.*, "Privacy Preservation for Federated Learning With Robust Aggregation in Edge Computing", *IEEE Internet Things J.*, vol. 10, no. 8, pp. 7343–7355, Apr. 2023, doi: 10.1109/JIOT.2022.3229122.

[12] T. Liu, X. Hu, H. Xu, T. Shu, and D. N. Nguyen, "High-accuracy low-cost privacy-preserving federated learning in IoT systems via adaptive perturbation", *J. Inf. Secur. Appl.*, vol. 70, p. 103309, Nov. 2022, doi: 10.1016/j.jisa.2022.103309.

[13] C. Mazzocca, N. Romandini, M. Colajanni, and R. Montanari, "FRAMH: A Federated Learning Risk-Based Authorization Middleware for Healthcare," *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 4, pp. 1679–1690, Aug. 2023, doi: 10.1109/TCSS.2022.3210372.

[14] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated Learning On Non-Iid Features Via Local Batch Normalization," *ICLR 2021*, 2021, [Online]. Available: https://iclr.cc/virtual/2021/poster/2846

[15] Y. Wang, Q. Shi, and T.-H. Chang, "Batch Normalization Damages Federated Learning on NON-IID Data: Analysis and Remedy," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece: IEEE, Jun. 2023, pp. 1–5. doi: 10.1109/ICASSP49357.2023.10095399.

[16] J. Zhong, H.-Y. Chen, and W.-L. Chao, "Making Batch Normalization Great in Federated Deep Learning," 2023, *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*. Accessed: May 16, 2025. [Online]. Available: https://openreview.net/forum?id=iKQC652XIk

[17] D. Freeman, S. Jain, M. Duermuth, B. Biggio, and G. Giacinto, "Who Are You? A Statistical Approach to Measuring User Authenticity," in *Proceedings 2016 Network and Distributed System Security Symposium*, San Diego, CA: Internet Society, 2016. doi: 10.14722/ndss.2016.23240.

[18] S. Wiefling, M. Dürmuth, and L. Lo Iacono, "What's in Score for Website Users: A Data-Driven Long-Term Study on Risk-Based Authentication Characteristics", in *Financial Cryptography and Data Security*, vol. 12675, N. Borisov and C. Diaz, Eds., in Lecture Notes in Computer Science, vol. 12675. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 361–381. doi: 10.1007/978-3-662-64331-0_19.

[19] S. Wiefling, P. R. Jørgensen, S. Thunem, and L. Lo Iacono, "Login Data Set for Risk-Based Authentication", Zenodo, Jun. 30, 2022. doi: 10.5281/ZENODO.6782155.

[20] P. M. Sánchez Sánchez, A. Huertas Celdrán, N. Xie, G. Bovet, G. Martínez Pérez, and B. Stiller, "FederatedTrust: A solution for trustworthy federated learning", *Future Gener. Comput. Syst.*, vol. 152, pp. 83–98, Mar. 2024, doi: 10.1016/j.future.2023.10.013.

[21] M. S. Jere, T. Farnan, and F. Koushanfar, "A Taxonomy of Attacks on Federated Learning", *IEEE Secur. Priv.*, vol. 19, no. 2, pp. 20–28, Mar. 2021, doi: 10.1109/MSEC.2020.3039941.

[22] R. Aziz, S. Banerjee, S. Bouzefrane, and T. Le Vinh, "Exploring Homomorphic Encryption and Differential Privacy Techniques towards Secure Federated Learning Paradigm", *Future Internet*, vol. 15, no. 9, p. 310, Sep. 2023, doi: 10.3390/fi15090310.

[23] M. Firdaus, H. T. Larasati, and K.-H. Rhee, "A Secure Federated Learning Framework using Blockchain and Differential Privacy", in *2022 IEEE 9th International Conference on Cyber Security and Cloud Computing (CSCloud)/2022 IEEE 8th International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, Xi'an, China: IEEE, Jun. 2022, pp. 18–23. doi: 10.1109/CSCloud-EdgeCom54986.2022.00013.

[24] O. Ibrahim Khalaf *et al.*, "Federated learning with hybrid differential privacy for secure and reliable", *Secur. Priv.*, vol. 7, no. 3, p. e374, May 2024, doi: 10.1002/spy2.374.

[25] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks", *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, 2022.

[26] Y. Khazaeni, "Federated Learning With Matched Averaging", Sep. 18, 2023. doi: 10.1287/fa7d97f0-ba96-4959-a51b-cf12b34c6d20.

[27] J. Li, "Exploration and Analysis of FedAvg, FedProx, FedMA, MOON, and FedProc Algorithms in Federated Learning", in *Proceedings of the 1st International Conference on Data Science and Engineering*, Singapore, Singapore: SCITEPRESS - Science and Technology Publications, 2024, pp. 172–176. doi: 10.5220/0012836400004547.

[28] M. Cheng *et al.*, "MFTE: Multifactor and fuzzy trust evaluation for federated learning in mobile edge computing", *Comput. Netw.*, vol. 265, p. 111340, Jun. 2025, doi: 10.1016/j.comnet.2025.111340.

[29] W. Jiang *et al.*, "Fuzzy ensemble-based federated learning for EEG-based emotion recognition in Internet of Medical Things", *J. Ind. Inf. Integr.*, vol. 44, p. 100789, Mar. 2025, doi: 10.1016/j.jii.2025.100789.

[30] A. Mabrouk, R. P. Díaz Redondo, M. Abd Elaziz, and M. Kayed, "Ensemble Federated Learning: An approach for collaborative pneumonia diagnosis", *Appl. Soft Comput.*, vol. 144, p. 110500, Sep. 2023, doi: 10.1016/j.asoc.2023.110500.

[31] P. Oza and V. M. Patel, "Federated Learning-based Active Authentication on Mobile Devices", in *2021 IEEE International Joint Conference on Biometrics (IJCB)*, Shenzhen, China: IEEE, Aug. 2021, pp. 1–8. doi: 10.1109/IJCB52358.2021.9484338.

[32] M. Wazzeh *et al.*, "CRSFL: Cluster-based Resource-aware Split Federated Learning for Continuous Authentication", *J. Netw. Comput. Appl.*, vol. 231, p. 103987, Nov. 2024, doi: 10.1016/j.jnca.2024.103987.

[33] V. V. Thang, D. V. Pantiukhin, B. T. T. Quyen, and V. V. Vu, "A review of neural networks for rare intrusions detection in wireless networks", *J. Sci. Technol. Inf. Secur.*, vol. 3, no. 20, pp. 23–34, Dec. 2023, doi: 10.54654/isj.v3i20.984.

[34] Y. Chen, Y. Gui, H. Lin, W. Gan, and Y. Wu, "Federated Learning Attacks and Defenses: A Survey", in *2022 IEEE International Conference on Big Data (Big Data)*, Osaka, Japan: IEEE, Dec. 2022, pp. 4256–4265. doi: 10.1109/BigData55660.2022.10020431.

[35] D. Chen, X. Jiang, H. Zhong, and J. Cui, "Building Trusted Federated Learning: Key Technologies and Challenges", *J. Sens. Actuator Netw.*, vol. 12, no. 1, p. 13, Feb. 2023, doi: 10.3390/jsan12010013.

[36] H. Fereidouni, "Enhancing Risk-Based Authentication with Federated Learning: Introducing the F-RBA Framework", *Univ. Montr.*, vol. abs/2412.12324, 2024, [Online]. Available: https://umontreal.scholaris.ca/items/589a3192-6edf-4566-95ea-45cc3d8f2235.

ABOUT THE AUTHOR

Le Vinh Thinh

Workplace: Ho Chi Minh City University of Technology and Education, Vietnam

Email: thinhlv@hcmute.edu.vn

Education: After completing a B.Sc. at the University of Natural Sciences, Le Vinh Thinh obtained a Master's degree in IT at the U.P. Technology University in India in 2006 and completed a Ph.D. in Computer Science at the Conservatoire National des Arts et Métiers (CNAM), Paris, France, in 2017. Currently, he is a faculty member in the Department of Information Technology at Ho Chi Minh City University of Technology and Education, Vietnam. He is the author and co-author of over 20 scientific articles.

Recent research direction: Research interests include Trust and Reputation Systems, Security, Mobile Cloud Computing, and the Internet of Things (IoT) based AI.

Tên tác giả: **Lê Vĩnh Thịnh**

Nơi làm việc: Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh.

Email: thinhlv@hcmute.edu.vn

Học vấn: Nhận bằng Cử nhân tại Trường Đại học Khoa học Tự nhiên và nhận bằng Thạc sĩ Công nghệ thông tin tại Đại học Công nghệ U.P ở Ấn Độ vào năm 2006 và hoàn thành Tiến sĩ Khoa học Máy tính tại Conservatoire National des Arts et Métiers, Pháp vào năm 2017. Hiện tại, đang là giảng viên tại Khoa Công nghệ Thông tin, Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh, Việt Nam. Là tác giả và đồng tác giả của hơn 20 bài báo khoa học.

Hướng nghiên cứu hiện nay: Hệ thống tin cậy và danh tiếng, bảo mật, điện toán đám mây di động và AI dựa trên Internet vạn vật.