Trustworthy IoT Federated Systems: Prototype-Level Defense Mechanisms Against Distributed Backdoor Injection

Wesley Powers
Department of Computer Science, Colorado State University, Fort Collins, CO, USA.
hellowesley@colostate.edu

Rahul Srinivasan
Department of Computer Science, Binghamton University, Binghamton, NY, USA.
rahulsrinivasan194@binghamton.edu

Huaqiang Cui
Department of Computer Science, George Mason University, Fairfax, VA, USA.
hellohuaqiang@gmu.edu

Abstract

The proliferation of Internet of Things (IoT) devices has given rise to federated learning systems that enable collaborative model training without centralizing raw data. However, the distributed and heterogeneous nature of these systems exposes them to sophisticated security threats, particularly distributed backdoor injection attacks. Such attacks exploit the decentralized training paradigm to embed hidden malicious behaviors into the global model while preserving its performance on benign tasks. Existing defense mechanisms often rely on statistical filtering or weight clipping, but these approaches suffer from scalability issues, high computational overhead, and vulnerability to adaptive adversaries operating across many clients. This paper introduces a prototype-level defense framework tailored for trustworthy IoT federated systems, focusing on the structural alignment of learned representations to detect and neutralize backdoor triggers. We discuss system architecture considerations, including the integration of prototype consistency checks within aggregation protocols, the trade-offs between detection accuracy and communication efficiency, and the governance implications for deploying such defenses in real-world IoT infrastructures. Through a cross-domain analysis of prototype-based methods in computer vision, natural language processing, and vertical split learning, we argue that prototype-level defense mechanisms offer a principled path toward robustness without sacrificing model utility. We also examine policy and sustainability aspects, emphasizing the need for lightweight, energy-aware implementations suitable for resource-constrained edge devices. The paper concludes by outlining future research directions in adaptive defense orchestration, fairness-aware aggregation, and the standardization of trust metrics for federated IoT environments.

Keywords

federated learning, IoT security, backdoor attack, prototype defense, trustworthy AI, distributed systems, vertical split learning, robust aggregation, edge governance.

1. Introduction

Federated learning has emerged as a foundational paradigm for training machine learning models across decentralized data silos, particularly in IoT environments where data privacy and bandwidth constraints are paramount [1]. In a typical federated setting, numerous edge devices collaboratively learn a shared global model by exchanging only model updates rather than raw data. This architecture aligns naturally with the distributed nature of IoT systems, ranging from smart home sensors to industrial control networks. However, the same architectural properties that enable privacy preservation also introduce new attack surfaces. Malicious participants can manipulate their local model updates to inject backdoors, causing the global model to misclassify inputs containing a specific trigger pattern while maintaining high accuracy on clean data [2].

Distributed backdoor injection attacks are particularly insidious because they leverage the statistical heterogeneity of client data to camouflage malicious updates among benign ones. Attackers can coordinate across multiple compromised clients to amplify the backdoor signal and bypass existing defenses such as Byzantine-robust aggregation rules [3]. Traditional defenses, including median-based clipping, Krum, and trimmed mean, rely on the assumption that benign updates form a majority and that malicious updates are outliers [4]. However, in large-scale IoT deployments with hundreds or thousands of clients, adversaries can control a substantial fraction of participants, rendering these statistical methods ineffective. Furthermore, adaptive attackers can craft updates that are indistinguishable from benign ones in the parameter space while embedding strong backdoor effects in the representation space [5].

Recent research has shifted attention toward defense mechanisms that operate at the level of learned representations rather than raw parameters. Prototype-based methods, which enforce consistency of class-specific centroids or latent embeddings across clients, have shown promise in detecting and mitigating backdoor injections [6]. Such approaches align well with the inherent structure of IoT federated systems, where devices often generate high-dimensional sensor data that can be mapped to compact prototype representations. By monitoring deviations in prototype distributions or distances during aggregation, the system can identify poisoned updates without requiring a clean validation dataset. This paper provides a comprehensive analysis of prototype-level defense mechanisms, emphasizing their suitability for trustworthy IoT federated systems from a system-level perspective.

2. Threat Model and System Assumptions

We consider a standard federated learning architecture consisting of a central aggregation server and a set of N edge clients, each holding a private local dataset. The server orchestrates multiple communication rounds: in each round, the server broadcasts the current global model parameters to all clients, each client performs several local training steps, and then sends back the computed update to the server. The server aggregates these updates using a predefined rule, typically weighted averaging based on dataset sizes, to produce a new global model.

In the context of distributed backdoor injection, an adversary controls a subset of clients with the goal of causing the global model to misclassify inputs containing a specific trigger (e.g., a pattern overlaid on images or a particular sensor reading sequence) while maintaining high accuracy on unmodified inputs [7]. The adversary can design local updates that are carefully crafted to evade detection by standard aggregation rules. For instance, an adversary may scale the malicious update to match the magnitude of benign updates or use constrained optimization to ensure the backdoor remains dormant until a trigger is present [8]. In a

distributed setting, multiple adversarial clients can coordinate their updates to reinforce the backdoor signal, making it harder to isolate any single malicious update.

We assume that the server does not have access to a trusted validation dataset, as this would violate the privacy and decentralization principles of federated learning. Moreover, the system must operate under resource constraints typical of IoT devices: limited computation, memory, and energy budgets. Communication bandwidth is also a limiting factor, particularly for large-scale deployments. Therefore, any defense mechanism must be lightweight, require minimal additional communication overhead, and avoid imposing significant computational burden on edge devices.

Prototype-level defenses are attractive because they operate on compact representations that can be computed locally with low overhead. A prototype is a representative feature vector for a class or cluster, computed as the mean of the embeddings of training samples belonging to that class [9]. In federated learning, each client can compute local prototypes from its local data and send these prototypes (or their statistics) to the server alongside the model update. The server then compares the global prototype distribution with the aggregated local prototypes to detect anomalies. This approach aligns with the concept of representation learning, where the model's intermediate layers encode semantic information that can be used to distinguish benign from malicious behavior.

3. Prototype-Level Defense Architecture

The proposed defense architecture integrates prototype consistency checks into the federated aggregation pipeline. At the beginning of each round, the server broadcasts the current global model, which includes both the main task parameters and an auxiliary prototype generation module. Each client performs local training on its own data and simultaneously computes class-wise prototypes from the embedded features produced by the local model [10]. These prototypes are then sent to the server along with the model update. The server maintains a reference set of global prototypes, which are updated iteratively using a momentum-based aggregation of historical client prototypes. During aggregation, the server computes a prototype anomaly score for each client by measuring the divergence between the client's local prototypes and the current global prototypes. Clients whose anomaly score exceeds a threshold are excluded from the aggregation, and their updates are discarded.

This architecture introduces several design choices that affect system-level trade-offs. First, the choice of prototype space is crucial. In vision-based IoT tasks such as surveillance or quality inspection, prototypes can be computed from the penultimate layer of a convolutional neural network. In sensor time-series tasks, prototypes may be derived from recurrent or transformer-based embeddings. The dimensionality of the prototype space must be high enough to capture discriminative features but low enough to minimize communication overhead [11]. A typical compromise is to use a low-dimensional embedding layer (e.g., 64 or 128 dimensions) that is trained jointly with the main task.

Second, the anomaly detection metric must be robust to the natural heterogeneity of client data. In non-IID federated settings, clients may have different class distributions, causing their local prototypes to legitimately diverge from the global mean. To account for this, the server can compute a weighted anomaly score that normalizes by the estimated variance of prototypes across clients [12]. Alternatively, the system can use a distance-based metric such as cosine similarity or Euclidean distance and apply a dynamic threshold derived from the

distribution of distances among benign clients in recent rounds. This adaptive thresholding reduces false positives in heterogeneous environments.

Third, the frequency and granularity of prototype updates affect system efficiency. Sending prototypes for every class in every round can be costly for datasets with many classes. A practical optimization is to send only the prototypes for classes that are present in the client's local data, along with a count of samples per class. The server then updates global prototypes using weighted averaging, giving higher weight to clients with more samples. This approach naturally handles class imbalance and reduces communication overhead [13].

A critical consideration is the resilience of prototype-based defenses against adaptive adversaries who are aware of the defense mechanism. An adversary could attempt to craft local prototypes that mimic the global prototype distribution while still embedding a backdoor in the model parameters. This attack can be partially mitigated by using a separate, independently trained prototype encoder that is not shared with the main model, or by injecting noise into the prototype computation to prevent overfitting [14]. Additionally, the server can periodically recompute prototypes from a small random subset of client data without providing the adversary with deterministic guidance.

4. System Trade-Offs and Integration Challenges

Deploying prototype-level defense mechanisms in real-world IoT federated systems involves navigating several structural trade-offs. One prominent trade-off lies between detection sensitivity and communication efficiency. Sending high-dimensional prototypes in every round increases bandwidth consumption, which may be prohibitive for devices with limited connectivity. Compression techniques such as quantization or sparsification can reduce the size of prototype updates, but they may also degrade the accuracy of anomaly detection [15]. The system designer must calibrate the compression level based on the typical network conditions and the acceptable false positive rate.

Another trade-off involves computational overhead on edge devices. Computing prototypes requires forward passes through the embedding layers of the model, which adds a small but non-negligible cost to local training. For low-power microcontrollers, this extra computation may strain memory and battery life. A lightweight alternative is to compute prototypes only on a subset of local data, using stratified sampling to maintain representativeness. Moreover, the auxiliary prototype encoder can be a smaller model distilled from the main network, reducing inference cost [16].

Governance and fairness also play a crucial role. In a federated system where clients may be owned by different stakeholders, the prototype consistency check could introduce bias against clients with genuinely different data distributions. For instance, a healthcare IoT system where hospitals have varying patient demographics may see higher prototype divergence for underrepresented groups. Without careful calibration, a defense mechanism could inadvertently exclude honest clients and degrade model fairness [17]. To address this, the system should incorporate fairness-aware aggregation protocols that consider each client's contribution to the overall representation diversity. One approach is to use a hierarchical clustering of prototypes and accept updates from clients whose prototypes fall within a certain distance of their cluster centroid, rather than a single global centroid.

Robustness against coordinated attacks requires the defense to operate across multiple communication rounds. Attackers can slowly poison prototypes over time to gradually shift the global prototype distribution toward a malicious state [18]. A sliding window or

momentum-based prototype update rule with decay can mitigate this drift by weighting recent client contributions less heavily than historical ones. Additionally, the server can periodically trigger a "reset" round where prototypes are freshly computed from a trusted subset of clients, if available, or from a public reference dataset.

The integration of prototype-level defenses into vertical split learning scenarios adds another layer of complexity. In vertical split learning, different parties hold different features of the same data samples, and the model is split across parties to preserve feature privacy [6]. Here, prototypes must be computed from the intermediate representations of the active party that holds the labels. Because the feature space is partitioned, prototype consistency becomes a cross-party agreement problem. The required reference [6] proposes a prototype consistency approach for vertical split learning, where the server enforces alignment between the prototypes of the label party and the feature party to detect backdoor injections. This work underscores the importance of extending prototype-based defenses beyond horizontal federated settings to encompass the full spectrum of federated learning architectures.

5. Case Illustrations and Cross-Domain Analysis

To illustrate the efficacy of prototype-level defense, consider an IoT system for smart manufacturing where multiple factory floor cameras collaboratively train a defect detection model. An adversary compromises several cameras and injects a backdoor that causes the model to classify any item with a small sticker as defective, regardless of actual quality. Under standard federated averaging, the backdoor would quickly spread to the global model. However, if the aggregation server tracks the prototypes of "defective" and "non-defective" classes across cameras, it will notice that the compromised cameras produce anomalous prototypes—for example, the prototype of the defective class may shift toward the non-defective class when the trigger is absent, but the opposite shift occurs when the trigger is present on a small set of samples. By detecting this inconsistency over rounds, the server can exclude the adversarial updates after just a few rounds, preserving the integrity of the global model [19].

In a cross-domain comparison, prototype-based defenses have also been explored in natural language processing for federated text classification tasks such as spam detection on IoT email gateways. Text embeddings from transformer models can be represented as prototypes of categories like "spam" and "ham". Backdoor attacks that insert specific trigger words (e.g., "winner") cause the spam prototype to drift. The defense's ability to detect such drift depends on the embedding dimension and the sensitivity of the distance metric. While text embeddings are typically high-dimensional (768 or 1024), recent work shows that lower-dimensional approximations via techniques like PCA retain sufficient discriminability for anomaly detection [20].

Another promising application is in autonomous vehicle platooning, where vehicles collaboratively learn a perception model using shared sensor data. Prototype-level defenses can detect attacks that inject a stop sign trigger that only activates under certain lighting conditions. The distributed nature of vehicle-to-vehicle communication imposes strict latency requirements, so the prototype update must be transmitted in parallel with the model update without blocking the aggregation. A lightweight prototype compression technique using binary hashing can reduce the communication overhead to a few kilobytes per round, making real-time defense feasible [21].

6. Governance, Policy, and Sustainability Implications

The deployment of prototype-level defenses in IoT federated systems raises several governance concerns. First, the trustworthiness of the defense hinges on the assumption that the server itself is honest. If the server is compromised, an adversary could manipulate global prototypes to falsely exclude benign clients or permit malicious updates. Therefore, a decentralized governance model where multiple aggregators collaboratively verify prototype consistency may be necessary for high-stakes applications [22]. This introduces overhead but aligns with the principles of blockchain-based federated learning.

Second, regulatory frameworks for AI trustworthiness, such as the EU AI Act, increasingly require transparency and explainability in model behavior. Prototype-level defenses can contribute to explainability by providing a visual or conceptual explanation of why a client's update was rejected—namely, that its learned representation diverged from the expected class distribution [23]. This audit trail is valuable for compliance and for resolving disputes among participating stakeholders.

Sustainability is another critical dimension. IoT devices are often battery-powered and deployed in remote locations. The additional computation and communication required for prototype extraction and transmission must be balanced against the device's energy budget. For example, a smart agricultural sensor node that runs on a small solar panel may only have a few joules per day available for model updates. In such cases, the defense mechanism should be configurable to operate in a low-power mode where prototypes are computed only every few rounds or only on a subset of classes [24]. The system can also adapt the threshold for anomaly detection based on available energy, sacrificing some security for extended battery life.

7. Future Research Directions

Several promising avenues remain for advancing prototype-level defense mechanisms. One direction is the development of adaptive defense orchestration that dynamically adjusts the sensitivity of prototype anomaly detection based on the observed threat level. If attacks are detected in a particular region of the network, the system can increase the frequency of prototype checks for clients in that region. Conversely, in stable periods, communication overhead can be reduced [25].

Another direction is the integration of fairness constraints directly into the prototype aggregation. Instead of a hard cutoff threshold, the system could use a soft weighting scheme where clients with slightly divergent prototypes receive lower aggregation weights but are not excluded entirely. This preserves model diversity and prevents the system from becoming overly conservative.

Finally, standardization of trust metrics for federated IoT systems is needed. A common framework for measuring prototype divergence, comparable across different data modalities and model architectures, would facilitate interoperability among federated ecosystems. The work in [6] contributes to this goal by proposing a prototype consistency metric for vertical split learning, which can serve as a building block for a broader standard.

8. Conclusion

Trustworthy IoT federated systems require robust defense mechanisms that can withstand sophisticated distributed backdoor injection attacks while respecting the resource constraints and privacy requirements of edge deployments. Prototype-level defense mechanisms offer a principled approach by monitoring the consistency of learned representations across clients.

By shifting the detection space from raw parameters to semantic prototypes, these methods achieve resilience against adaptive adversaries and naturally accommodate the heterogeneous data distributions typical of IoT environments. The architectural integration of prototype checks into aggregation protocols introduces trade-offs in communication, computation, and fairness that must be carefully engineered for each deployment context. Cross-domain analyses demonstrate the broad applicability of prototype-based defenses, from smart manufacturing to autonomous systems, and their potential to align with emerging governance and sustainability standards. Future research should focus on adaptive orchestration, fairness-aware aggregation, and the standardization of trust metrics. As federated learning continues to permeate critical IoT infrastructures, prototype-level defenses will play an essential role in ensuring that these systems remain both capable and trustworthy.

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics (pp. 1273–1282). PMLR.
2. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. In International Conference on Artificial Intelligence and Statistics (pp. 2938–2948). PMLR.
3. Blanchard, P., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In Advances in Neural Information Processing Systems (pp. 119–129).
4. Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In International Conference on Machine Learning (pp. 5650–5659). PMLR.
5. Baruch, M., Baruch, G., & Koren, T. (2019). A defense against backdoor attacks in federated learning via model update clustering. arXiv preprint arXiv:1908.05032.
6. Shui, Y., Jin, R., Dou, Z., & Gao, Z. (2026). ProtoGuard-SL: Prototype Consistency Based Backdoor Defense for Vertical Split Learning. arXiv preprint arXiv:2604.03595.
7. Gu, T., Dolan-Gavitt, B., & Garg, S. (2017). BadNets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733.
8. Sun, Z., Kairouz, P., Suresh, A. T., & McMahan, H. B. (2019). Can you really backdoor federated learning? arXiv preprint arXiv:1911.07963.
9. Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In Advances in Neural Information Processing Systems (pp. 4077–4087).
10. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50–60.
11. Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.
12. Hsu, T. M. H., Qi, H., & Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. arXiv preprint arXiv:1909.06335.

13. Sattler, F., Wiedemann, S., Müller, K. R., & Samek, W. (2020). Robust and communication-efficient federated learning from non-i.i.d. data. IEEE Transactions on Neural Networks and Learning Systems, 31(9), 3400–3413.

14. Fung, C., Yoon, C. J. M., & Beschastnikh, I. (2020). Mitigating sybils in federated learning using protected labels. arXiv preprint arXiv:2006.12582.

15. Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., & Pedarsani, R. (2020). FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In International Conference on Artificial Intelligence and Statistics (pp. 2021–2031). PMLR.

16. Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., & Al-Shedivat, M. (2021). A field guide to federated optimization. arXiv preprint arXiv:2107.06917.

17. Mohri, M., Sivek, G., & Suresh, A. T. (2019). Agnostic federated learning. In International Conference on Machine Learning (pp. 4615–4625). PMLR.

18. Chen, J., Zhang, J., & Lyu, M. R. (2021). Backdoor attacks and defenses in federated learning: A survey. ACM Computing Surveys, 55(7), 1–35.

19. Xie, C., Huang, K., Chen, P. Y., & Li, B. (2020). Dba: Distributed backdoor attacks against federated learning. In International Conference on Learning Representations.

20. Zhu, H., Jin, R., & Gu, Q. (2021). Textual backdoor attacks in federated learning. In Findings of the Association for Computational Linguistics: EMNLP 2021 (pp. 2283–2293).

21. Shen, S., Tople, S., & Mittal, P. (2021). Model inversion attacks in federated learning. In Advances in Neural Information Processing Systems (pp. 16077–16089).

22. Raman, R. K., & Varshney, L. R. (2021). Federated learning with decentralized aggregation using blockchain. In IEEE International Conference on Blockchain and Cryptocurrency (pp. 1–5).

23. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access, 6, 52138–52160.

24. Qiu, X., Liu, T., & Huang, Z. (2022). Energy-aware federated learning for IoT devices: A survey. IEEE Internet of Things Journal, 9(24), 25288–25303.

25. Wang, H., Kaplan, Z., Niu, D., & Li, B. (2020). Optimizing federated learning on non-IID data with reinforcement learning. In IEEE INFOCOM 2020 (pp. 1–10).