Vigilante Defender: A Vaccination-based Defense Against Backdoor Attacks on 3D Point Clouds Using Particle Swarm Optimization

Agnideven Palanisamy Sundar[2†], Feng Li[1†], Xukai Zou[2‡], Yucheng Xie[3⋆] and Ryan Hosler[2†]

[1]Department of Computer and Information Technology
[2]Department of Computer Science
[3]Department of Computer Science and Engineering
[†]Purdue University, Indianapolis, IN, USA.
[‡]Indiana University-Indianapolis, Indianapolis, IN, USA.
[⋆]Yeshiva University, New York, NY, USA.
palania@purdue.edu, fengli@purdue.edu, xzou@iu.edu, yucheng.xie@yu.edu, rjhosler@iu.edu

***Abstract*—Backdoor attacks on 3D Point Clouds (PCs) pose a serious threat by embedding hidden triggers into a subset of the training data. These triggers cause targeted misclassifications at inference time while leaving the model's behavior unaffected in the absence of triggers, making them stealthy and difficult to detect. In distributed learning settings, where a central trainer aggregates data from multiple sources and offers only black-box access to the model, a single malicious contributor can compromise the model's integrity if defenses are not in place. We propose a novel client-side defense that empowers individual contributors to act as vigilante defenders. By injecting benign 'vaccination' triggers—identified via Particle Swarm Optimization—into their local training data, defenders can proactively neutralize potential backdoors without prior knowledge of their location or structure. Experiments on standard benchmarks with PointNet and DGCNN show our method significantly reduces attack success while preserving classification accuracy, outperforming existing defenses.**

***Index Terms*—3D Point Clouds, Backdoor Defense, Particle Swarm Optimization, Client-level Defense, Poisoning Attacks**

I. Introduction

Point clouds are increasingly pivotal in autonomous driving, architectural design, and augmented reality due to their detailed 3D representations [1]–[3]. However, their growing adoption brings new security vulnerabilities [4], including the threat of backdoor attacks—stealthy data poisoning strategies where models behave normally except when exposed to hidden triggers.

Compared to adversarial attacks, backdoor attacks are easier to execute and harder to detect, especially in collaborative learning settings. In many 3D point cloud applications, models are trained by aggregating data from multiple external contributors. These contributors upload local datasets to a central trainer, which periodically updates the model and provides only black-box access to end users. A single malicious contributor can compromise the model's integrity by embedding poisoned data if the trainer lacks adequate validation or defense mechanisms. Data poisoning-based backdoor attacks are far easier to implement than adversarial attacks of the same caliber.

To address this risk, we propose a novel client-side defense mechanism called vaccination. Unlike server-based defenses [5]–[8], our method empowers individual contributors to protect the shared model. Specifically, a vigilant defender injects benign 'vaccine' triggers into their training data—synthetic patterns designed to neutralize harmful triggers introduced by an attacker. We leverage Particle Swarm Optimization (PSO) [9] to identify likely backdoor configurations using only black-box access, then use these candidates to retrain the model with correct labels, discouraging reliance on malicious features.

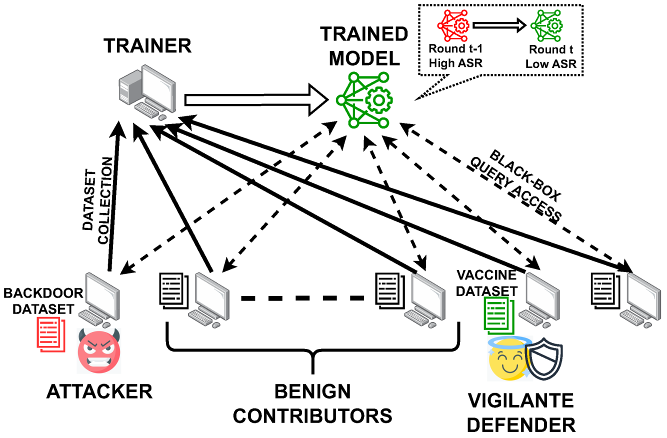
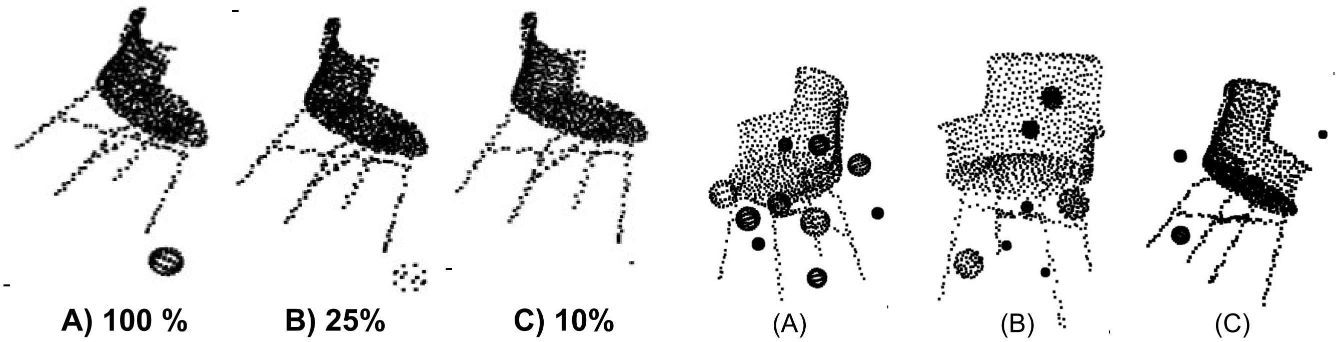


Fig. 1: High-level overview of our vaccination defense executed by a vigilante defender

Our setting assumes that: (1) a benign trainer aggregates data from multiple contributors but does not perform backdoor checks; (2) contributors—including the attacker and defender—have only black-box access to the model; (3) the attacker inserts triggers and mislabels their data to manipulate



(a)

(b)

Fig. 2: Subfigure (a) is an example showing triggers with different number of points. Subfigure (b) demonstrates the test-time ineffectiveness of multiple triggers.

model behavior; and (4) the vigilante defender detects these effects and responds in future training rounds. This threat model follows the assumptions in recent point cloud backdoor literature [10]–[12]. Fig. 1 presents a comprehensive overview of our defense strategy to backdoor attacks in a typical distributed learning scenario

Our results show that this vaccination strategy significantly reduces attack success rates across multiple architectures and datasets while preserving main task accuracy, outperforming prior defenses.

We make the following contributions in the paper:

- We are the first to build a client-level defense against backdoor attacks in Point Cloud environments.
- We use a modified version of Particle Swarm Optimization for effective detection and neutralization of backdoor triggers.
- We demonstrate the efficiency of our approach under two different datasets and three SOTA attack methods.
- We test our approach against existing backdoor defenses under various attacks to show the superiority of our method.

II. Motivation

Existing defenses predominantly rely on the server or model trainer to handle backdoor detection, placing the full burden of defense on a centralized entity. This not only limits transparency and control for data contributors but also introduces systemic risk—if a malicious party uploads poisoned data and the trainer fails to detect it, the model's integrity is compromised for all users.

To address this, we explore a black-box client-side defense paradigm in which contributors can only query the model using their local labeled data. Our proposed client-side approach empowers a vigilant participant to counteract backdoor triggers without server-side intervention. The core idea is to retrain the model using correctly labeled point clouds (PCs) containing triggers, effectively neutralizing their influence. This dual-labeling—where attackers mislabel triggers and defenders correct them—teaches the model to disregard the trigger as a discriminative feature. To further streamline this process, we leverage two Search Space Reduction (SSR) strategies, described next.

A. SSR 1: Trigger Size and Effectiveness Correlation

We first examined how precisely the trigger's properties—size, shape, and position—must be matched to activate the backdoor. To test this, we trained a model with a specific backdoor trigger and evaluated its activation when modifying those properties at test time.

Our results show that even a reduced version of the trigger—containing only 25% or even 10% of the original points—was still effective (Fig. 2a). Minor positional shifts also did not disrupt activation. This suggests that defenders need only identify a small portion of the original trigger, significantly shrinking the search space. In contrast, inference triggers are far stricter in the 2D image domain.

B. SSR 2: Unknown Trigger Ineffectiveness

To further reduce query complexity, we explored the use of multiple vaccine triggers embedded simultaneously in a single 3D PC.

Experiments revealed that the model's Main Task Accuracy (MTA) remained stable as long as these triggers did not touch or closely surround the object. As shown in Fig. 2b, multiple triggers placed near but not on the object do not distort its classification. This behavior is unique to 3D point clouds—unlike 2D images, where overlapping triggers often distort the source input.

Together, these insights allow us to insert multiple diagnostic triggers into the same PC. If a misclassification occurs, we infer that at least one trigger aligns with the original backdoor pattern—greatly narrowing the detection space.

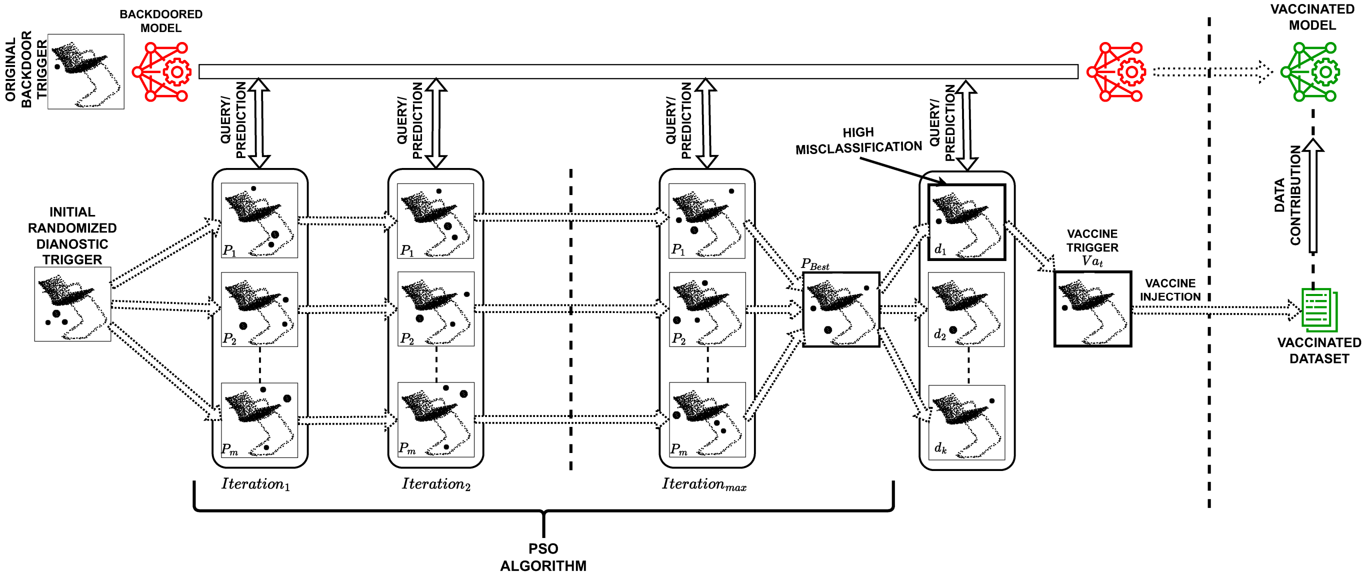


Fig. 3: An overview of the backdoor trigger detection approach using modified Particle Swarm Optimization.

III. Methodology

In this section, we outline the sequential steps of our proposed backdoor defense, integrating the components discussed earlier into a two-phase mechanism.

We use the following terminology: 1) b_t denotes the backdoor trigger used by the attacker. 2) $D_t = \{d_1, d_2, \ldots, d_k\}$ is the set of diagnostic triggers used during test-time by the vigilante defender to probe for backdoor behavior. These are spherical triggers of varying sizes and densities, chosen to limit the parameter space. 3) $Va_t = \{va_1, va_2, \ldots, va_n\}$, with $n \leq k$, is a subset of D_t and represents the vaccine triggers injected into the training data to mitigate backdoor effects.

We now describe the two main phases of our defense process.

A. Backdoor Trigger Diagnosis – PSO

To detect backdoor triggers, we assume the vigilante defender has black-box access to a potentially poisoned model. A naive strategy might involve randomly injecting multiple diagnostic triggers into test data and querying the model until misclassifications occur. However, this brute-force approach is inefficient and computationally costly.

Particle Swarm Optimization (PSO) is an optimization technique inspired by the social behavior of birds flocking or fish schooling. It uses a swarm of particles, where each particle represents a potential solution in the search space. Each particle updates its position based on its own best-known position and the global best position found by the swarm. The optimization process involves iteratively adjusting particle positions and velocities to explore the solution space and converge on an optimal solution.

We employ a modified PSO algorithm tailored to our constraints. Unlike standard PSO, where each particle represents a candidate solution, our method treats a set of diagnostic trigger positions as one particle. Each particle includes a fixed number of spherical triggers with randomized size, density, and 3D position. We generate multiple such particles and iteratively refine them using PSO.

We introduce a shift factor $\mathbf{s}_{ij}$ for each trigger j in particle i, initialized randomly and fixed during optimization. This allows each trigger in a particle to move independently, enhancing spatial exploration.

Let $\mathbf{P}_i = \{\mathbf{p}_{i1}, \ldots, \mathbf{p}_{im}\}$ represent the position vectors for triggers in particle i, and $\mathbf{S}_i = \{\mathbf{s}_{i1}, \ldots, \mathbf{s}_{ik}\}$ be their corresponding shift factors. We adapt PSO's update rules as follows:

$$\mathbf{v}_i^{(t+1)} = \omega \mathbf{v}_i^{(t)} + c_1 r_1 (\mathbf{p}_{\text{best},i} - \mathbf{P}_i^{(t)}) + c_2 r_2 (\mathbf{p}_{\text{g\_best}} - \mathbf{P}_i^{(t)}) \quad (1)$$

$$\mathbf{P}_i^{(t+1)} = \mathbf{P}_i^{(t)} + \mathbf{v}_i^{(t+1)} \quad (2)$$

$$\mathbf{p}_{ij}^{(t+1)} = \mathbf{p}_{ij}^{(t)} + \mathbf{s}_{ij} \cdot \mathbf{v}_i^{(t+1)} \quad (3)$$

Where:

- $\mathbf{v}_i^{(t)}$: velocity of particle i at iteration t.
- $\mathbf{p}_{ij}^{(t)}$: position of trigger j in particle i at iteration t.
- $\mathbf{p}_{\text{best},i}$: best local position found by particle i.
- $\mathbf{p}_{\text{g\_best}}$: global best position across all particles.

To prevent triggers from overlapping or colliding with the primary object in the point cloud, we apply a collision-avoidance function. The shift factor $\mathbf{s}_{ij}$ is defined in relation to the trigger's radius r and is bounded within $[0.1r, 0.5r]$, ensuring smooth but significant positional adjustments during optimization.

After several iterations, we select the global best particle—i.e., the set of diagnostic trigger positions that causes the highest misclassification rate—and test each diagnostic trigger individually. The most effective trigger—the one leading to the

TABLE I: The experiment results on different backdoor attacks, models, and datasets. We test the performance of our Backdoor Cure (B-Cure) method in the presence of different attacks. [PN++ is PointNet++]

Dataset	Model	Attack(→)	PointBA		PCBA		EfficientBA	
		Method(↓)	ASR	MTA	ASR	MTA	ASR	MTA
ModelNet40	PointNet	No Defense	100	87.31	93.27	87.11	90.13	87.71
		B-Cure	01.34	87.10	01.42	87.00	03.24	87.13
	PN++	No Defense	99.72	89.83	94.18	88.20	91.45	87.91
		B-Cure	02.12	88.03	01.47	86.92	03.68	87.04
	DGCNN	No Defense	98.65	91.93	95.96	91.10	92.17	91.12
		B-Cure	01.40	90.86	01.89	90.89	04.91	90.82
ShapeNetPart	PointNet	No Defense	100	98.01	96.26	97.66	90.11	97.84
		B-Cure	02.81	97.48	02.15	97.12	02.98	97.16
	PN++	No Defense	99.45	97.89	95.14	97.91	91.73	97.82
		B-Cure	02.52	97.11	02.95	97.01	05.95	97.17
	DGCNN	No Defense	100	98.11	94.92	97.97	90.96	97.15
		B-Cure	03.16	98.02	03.19	97.66	04.11	96.80

highest misclassification—is selected as the vaccination trigger and injected into the defender's training data for mitigation.

Fig. 3 illustrates this process: the poisoned model enables PSO to explore trigger placements, and over time, the best-performing combination is identified and isolated for defense.

B. Vaccine Injection – Backdoor Cure

If a diagnostic trigger yields a misclassification rate significantly above a threshold α, it likely mimics the attacker's backdoor trigger, signaling a potential backdoor presence. In such cases, it becomes critical to suppress the model's reliance on this trigger during inference.

To do so, the vigilante defender injects this top diagnostic trigger—now referred to as the vaccine trigger Va_t—into a subset of its 3D point cloud (PC) data while preserving correct labels. This retraining step helps the model learn that the trigger is not a reliable indicator of any class. Only one vaccine trigger is used ($n = 1$), and it is embedded in $\beta\%$ of samples across all classes to dilute the association between the trigger and any specific label. The parameters α (misclassification threshold) and β (vaccination rate) can be tuned based on how aggressively the defender aims to counteract the backdoor.

IV. Experimental Evaluation

This section outlines the experimental setup and demonstrates the effectiveness of our defense under various conditions.

*1) **Datasets**:* We use two benchmark datasets: ModelNet40 [13] and ShapeNetPart [14]. a) ModelNet40: 12,311 CAD models from 40 categories, split into 9,843 training and 2,468 testing samples. b) ShapeNetPart: 16,002 models across 16 categories, with 12,128 for training and 2,874 for testing.

*2) **Models**:* We evaluate the performance of three neural network models: PointNet [15], PointNet++ [16], and DGCNN [17]. PointNet is a simple deep learning model for processing point clouds, known for its ability to handle unordered point sets. PointNet++ is an advanced version of PointNet, incorporating hierarchical networks for capturing local structures. DGCNN (Dynamic Graph CNN) utilizes dynamic graph-based convolutional networks to capture complex patterns in data.

*3) **Attack Methods**:* This subsection outlines three prominent backdoor attacks that we will test our defense method against.

1. PointBA [11]: Introduces backdoors by perturbing point clouds via spatial transformations and feature disentanglement.

2. PCBA [10]: Injects small clusters of points as optimized triggers based on spatial and geometric features.

3. EfficientBA [18]: Uses a confidence-based scoring mechanism and greedy search to insert object-shaped triggers efficiently, following both 2D and 3D attack paradigms. We will use airplane-shaped trigger, following [18].

*4) **Attack, Defense, and Training Settings**:* Let's look at the hyperparameters we'll be using unless mentioned otherwise.

Attack. Following [11], we target the 'Toilet' class in ModelNet40, and 'Lamp' in ShapeNetPart. Each attack poisons 5% of the attacker's dataset by relabeling samples with embedded backdoor triggers.

Defense. We deploy 8 diagnostic triggers (k), select the top diagnostic trigger as the vaccine trigger ($n = 1$), and use 20 PSO particles (m), capped at 10,000 total queries (500 per particle). Detection uses only 500 unique records. To implement the mitigation, we inject the selected vaccine trigger into 10% (β) of the vigilante defender's local dataset while retaining the original class labels of those samples.

TABLE II: Comparison of our work with state-of-the-art backdoor detection methods for both ModelNet40 and ShapeNetPart using the PointNet++ architecture.

Dataset	Attack(→)	PointBA		PCBA		EfficientBA	
	Method(↓)	ASR	MTA	ASR	MTA	ASR	MTA
ModelNet40	No Defense	99.72	89.83	94.18	88.20	91.45	87.91
	PointCRT	73.86	86.55	66.71	87.11	74.35	86.41
	DBAPC	69.34	86.24	68.23	85.98	69.93	86.72
	CloudFort	78.67	82.61	62.53	80.74	72.77	81.93
	B-Cure	**02.12**	**88.03**	**01.47**	**86.92**	**03.68**	**87.04**
ShapeNetPart	No Defense	99.45	97.89	95.14	97.91	91.73	97.82
	PointCRT	77.78	96.78	74.64	97.61	78.48	96.87
	DBAPC	69.86	**97.63**	70.61	96.82	72.87	96.46
	CloudFort	71.39	95.99	61.57	92.57	73.95	93.30
	B-Cure	**02.52**	97.11	**02.95**	**97.01**	**05.95**	**97.17**

Model Training Models are trained for 200 epochs using the Adam optimizer (learning rate = 0.001) with standard augmentations: sampling, normalization, scaling, and translation. Each round uses a quarter of each contributor's data across four contributors (one attacker, one defender, two benign), split over four rounds.

*5) **Evaluation Metrics:*** We assess our models using two metrics: the Attack Success Rate (ASR) and Main Task Accuracy (MTA). ASR measures the percentage of point clouds misclassified into the target class with the backdoor trigger present, indicating the trigger's impact. MTA measures the correct classification rate of point clouds without any triggers, reflecting the negative impact of vaccination on training. The defense aims to minimize ASR while maximizing MTA.

V. Performance Evaluation

A. Trigger influence removal with Backdoor Cure

Table I presents the effectiveness of our vigilante vaccination defense across three attacks, evaluated on two datasets and three model architectures. In this setup, the attacker poisons the model during the first round of training, while all other contributors, including the vigilante defender, submit benign datasets. After this round, the defender uses the trained model to generate vaccine triggers, which are then injected into the training data for the second round. The reported results reflect model performance after this second training round, with α set to 0.4.

Our defense significantly reduces ASR for all attack types. The reduction is particularly strong against PointBA and PCBA, likely due to the similarity between their triggers and our spherical diagnostic triggers. For EfficientBA, which uses a non-spherical airplane-shaped trigger, some points remain effective post-vaccination; however, ASR still drops substantially. In all cases, the highest ASR observed after applying Backdoor Cure is only 5.95%.

B. Comparison with SOTA Detection Methods

We compare our work with three backdoor detection/robustness approaches. At the time of writing this paper, there are no direct client-level defenses.

1. PointCRT [7]: This method detects backdoors by adding varying levels of corruption to each test data record and checking for misclassification. In our work, we apply density, rotation, and distortion-based corruptions. To convert this detection method into a defense method, we use the corruption that triggers misclassification as a vaccine trigger and retrain the model.

2. DBAPC [8]: This method introduces minimal noise to the point cloud and monitors the extent of misclassification. Similar to PointCRT, we transform this into a defense method by using the noise that induces high misclassification rates as the vaccine to retrain the model.

3. CloudFort [19]: CloudFort is a backdoor defense method which employs a two-stage approach: *spatial partitioning* and *ensemble prediction*. The method partitions the input point cloud into multiple sub-point clouds by systematically removing points from different spatial regions, reducing the likelihood of retaining backdoor triggers.

As shown in Table. II, other methods aren't particularly effective in defending against backdoor attacks. The results demonstrate the performance under the PointNet++ model, indicating that DBAPC performs slightly better than PointCRT and CloudFort. This is due to DBAPC's ability to detect some points related to the original backdoor trigger, which aids in defense. Conversely, PointCRT approach fails to significantly reduce the ASR of the backdoor. CloudFort is particularly good against PCBA due to its low point count. It is important to note that these were originally detection and robust algorithms, not defense methods. Our detection process outperforms these two methods by requiring far fewer trials to detect the trigger closest to the original.

VI. Related Works

Backdoor attacks in point cloud data have recently garnered attention due to the unique challenges and potential impact they present [19]. Several studies have demonstrated the feasibility and increasing sophistication of such attacks: Li et al. [11] showed that 3D classifiers are susceptible to backdoors, Xiang et al. [10] highlighted the severity of misclassifications from subtle manipulations, and Zheng et al. [12] proposed imperceptible, robust attack techniques. More recent works by Ning et al. [20] and Bian et al. [21] introduced stealthier attacks using point features and auto-encoders, respectively, underscoring the need for specialized defenses.

While backdoor defenses in point cloud settings are limited, several approaches have emerged. Some defenses initially developed for image data, such as fine-pruning and spectral analysis, have been adapted to point clouds [5], [6]. Adversarial training has also been applied to improve robustness [22], [23], and Bian et al. [24] introduced iBA using self-reconstruction to embed imperceptible triggers. PointCRT [7] and DBAPC [8] detect backdoors via misclassification patterns under perturbations or random placements, while PointAPA [25] uses spatial partitioning and ensemble prediction. However, these are server-side defenses, limiting applicability for client-level control.

VII. Conclusion

This work introduces the first contributor-led defense against backdoor attacks in the domain of 3D Point Clouds, leveraging a novel vaccination-based strategy. By employing Particle Swarm Optimization (PSO), we effectively identify and neutralize backdoor triggers, ensuring robust model performance in a black-box setting. Our approach significantly reduces attack success rates while maintaining high classification accuracy, demonstrating superiority over existing defenses. This contribution paves the way for empowering data contributors in distributed learning systems and sets a foundation for exploring advanced defense mechanisms in complex data environments.

Future work includes automating the selection of vaccine parameters to reduce reliance on domain expertise and adapting our method for 2D imaging and Federated Learning (FL), enabling local model interactions for enhanced security.

Acknowledgment

This work is partially supported by the National Science Foundation under Grant Numbers DGE-2146359, DGE-2011117, CNS-2431725, and DGE-2436733, as well as by INDOT JTRP SPR-4934.

References

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[2] H. Yue, Q. Wang, H. Zhao, N. Zeng, and Y. Tan, "Deep learning applications for point clouds in the construction industry," *Automation in Construction*, vol. 168, p. 105769, 2024.

[3] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui, "Mobile augmented reality survey: From where we are to where we go," in *IEEE Access*, vol. 5. IEEE, 2017, pp. 6917–6950.

[4] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[5] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *RAID*, 2018, pp. 273–294.

[6] B. Tran, J. Li, A. Madry, A. Makkuva, L. Schmidt, D. Tsipras, and A. Vladu, "Spectral signatures in backdoor attacks," in *Advances in Neural Information Processing Systems*, 2018, pp. 8000–8010.

[7] S. Hu, W. Liu, M. Li, Y. Zhang, X. Liu, X. Wang, L. Y. Zhang, and J. Hou, "Pointcrt: Detecting backdoor in 3d point cloud via corruption robustness," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 666–675.

[8] Z. Xiang, D. J. Miller, S. Chen, X. Li, and G. Kesidis, "Detecting backdoor attacks against point cloud classifiers," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3159–3163.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[10] C. Xiang, X. Li, C. Liu, N. He, J. Zhao, and M. Liu, "A backdoor attack against 3d point cloud classifiers," *arXiv preprint arXiv:2105.12378*, 2021.

[11] X. Li, C. Zhao, A. Zheng, C. Zhang, and Y. Tian, "Pointba: Towards backdoor attacks in 3d point cloud," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4294–4303.

[12] A. Zheng, C. Zhao, C. Zhang, and Y. Tian, "Imperceptible and robust backdoor attack in 3d point cloud," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," *Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[14] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q.-X. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[15] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[16] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[17] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–12, 2019.

[18] Y. Wu, X. Han, H. Qiu, and T. Zhang, "Computation and data efficient backdoor attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4805–4814.

[19] W. Lan, Y. Yang, H. Shen, and S. Li, "Cloudfort: Enhancing robustness of 3d point cloud classification against backdoor attacks via spatial partitioning and ensemble prediction," *arXiv preprint arXiv:2404.14042*, 2024.

[20] X. Ning, Q. Xie, J. Xu, W. Jiang, J. Li, and Y. Ma, "Stealthy and robust backdoor attack against 3d point clouds through additional point features," *arXiv preprint arXiv:2412.07511*, 2024.

[21] Y. Bian, S. Tian, and X. Liu, "Mirrorattack: Backdoor attack on 3d point cloud with a distorting mirror," *arXiv preprint arXiv:2403.05847*, 2024.

[22] C. Xiang, C. Qi, N. Liu, T. Zhou, J. Chen, S. Bai, L. Wang, J. Yu, Y. Yang, and M. Liu, "Generating 3d adversarial point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9136–9144.

[23] H. Zhou, J. Zhang, X. Peng, J. Zhang, H. Li, S. Liu, and M. Wang, "Pointguard: Provably robust 3d point cloud classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 013–14 022.

[24] Y. Bian, S. Tian, and X. Liu, "iba: Backdoor attack on 3d point cloud via reconstructing itself," *IEEE Transactions on Information Forensics and Security*, 2024.

[25] X. Wang, M. Li, P. Xu, W. Liu, L. Y. Zhang, S. Hu, and Y. Zhang, "Pointapa: Towards availability poisoning attacks in 3d point clouds," in *European Symposium on Research in Computer Security*. Springer, 2024, pp. 125–145.