DEFEND: Poisoned Model Detection and Malicious Client Exclusion Mechanism for Secure Federated Learning-based Road Condition Classification

Sheng Liu
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
shengliu@kth.se

Panos Papadimitratos
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

Abstract

Federated Learning (FL) has drawn the attention of the Intelligent Transportation Systems (ITS) community. FL can train various models for ITS tasks, notably camera-based Road Condition Classification (RCC), in a privacy-preserving collaborative way. However, opening up to collaboration also opens FL-based RCC systems to adversaries, i.e., misbehaving participants that can launch Targeted Label-Flipping Attacks (TLFAs) and threaten transportation safety. Adversaries mounting TLFAs poison training data to misguide model predictions, from an actual source class (e.g., wet road) to a wrongly perceived target class (e.g., dry road). Existing countermeasures against poisoning attacks cannot maintain model performance under TLFAs close to the performance level in attack-free scenarios, because they lack specific model misbehavior detection for TLFAs and neglect client exclusion after the detection. To close this research gap, we propose DEFEND, which includes a poisoned model detection strategy that leverages neuron-wise magnitude analysis for attack goal identification and Gaussian Mixture Model (GMM)-based clustering. DEFEND discards poisoned model contributions in each round and adapts accordingly client ratings, eventually excluding malicious clients. Extensive evaluation involving various FL-RCC models and tasks shows that DEFEND can thwart TLFAs and outperform seven baseline countermeasures, with at least 15.78% improvement, with DEFEND remarkably achieving under attack the same performance as in attack-free scenarios.

CCS Concepts

• **Security and privacy** → *Distributed systems security*; *Mobile and wireless security*; • **Computing methodologies** → *Machine learning*; *Neural networks*; • **Applied computing** → **Transportation**.

Keywords

Federated Learning, Label-Flipping Attacks, Road Condition Classification, Defensive Mechanism, Transportation Safety

ACM Reference Format:
Sheng Liu and Panos Papadimitratos. 2026. DEFEND: Poisoned Model Detection and Malicious Client Exclusion Mechanism for Secure Federated Learning-based Road Condition Classification. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), March 23–27, 2026, Thessaloniki, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3748522.3779807

SAC '26, Thessaloniki, Greece

ACM ISBN 979-8-4007-2294-3/2026/03
https://doi.org/10.1145/3748522.3779807

1 Introduction

The automated identification of road surface conditions, e.g., unevenness level, friction magnitude, and material type, is an important tool for Intelligent Transportation Systems (ITS) [5]. For example, when an autonomous vehicle detects in real-time a water-logged road ahead through its cameras and *Road Condition Classification (RCC)* model (notably Deep Neural Network, DNN), it can proactively adjust its speed, activate the traction control system, or reconfigure the suspension. Capitalizing on *image data* is particularly feasible for RCC [24] due to the widespread availability and cost-effectiveness of on-board cameras, capturing rich contextual details essential for classification accuracy.

If each vehicle relied only on its own data for model training, RCC would suffer from an inherently skewed perception; e.g., a vehicle driving mostly in the countryside would lack data and the ability to classify inputs from a city center environment. The *centralized* data collection, with all data uploaded to a server to perform model training is impractical too in the long term: 1) regulations about user privacy[1,2,3] are increasingly strict, 2) computation and energy costs at the data center leave vehicle resources underutilized; and 3) transmitting original data leads to high bandwidth usage and increased response latency for vehicles. *Federated Learning (FL)* [19, 22, 41], notably cross-device horizontal FL, provides a promising solution for the above-mentioned dilemma between regulation considerations and resource utilization, while collecting user contributions across a large-scale deployment with varying environments. Through an iterative client-server collaboration on model parameters, a top-performing RCC model can be learned [37].

However, FL-based RCC systems relying on potentially any participant are vulnerable to compromised, adversarial clients. Even though credential management, access control, and secure communication [26] can be a first line of strong defense, the vulnerability to *Targeted Label-Flipping Attacks (TLFAs)* [30] remains: malicious clients, i.e., internal adversaries, deliberately change their image labels from a source class (true class) to a target class (falsified class), and use such poisoned data for local model training, thus

[1] https://gdpr-info.eu/
[2] https://www.oag.ca.gov/privacy/ccpa
[3] http://en.npc.gov.cn.cdurl.cn/2021-12/29/c_694559.htm

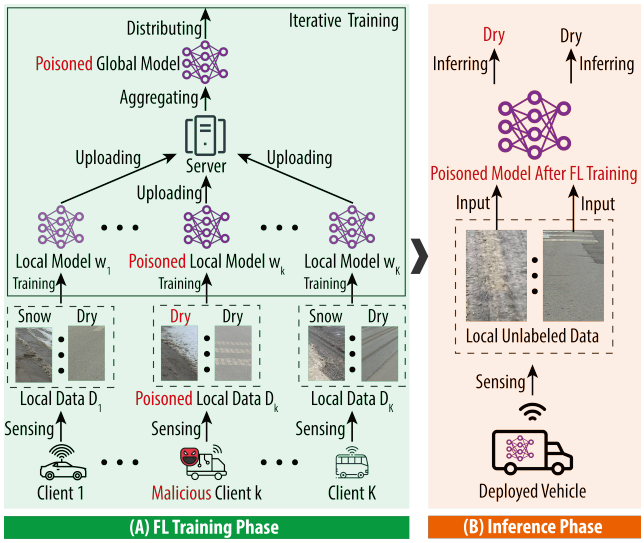

Figure 1: Illustration of TLFAs in FL-RCC. (A) Training Phase: Adversaries deliberately mislabel their data; their local models are poisoned after local training, and the global model is poisoned after global aggregation. (B) Inference Phase: Vehicles equipped with the learned model would predict wrong road conditions that threaten transportation safety.

misleading the prediction results of the aggregated global model. For example, as illustrated in Figure 1, during the FL training phase, if the adversary flips road friction labels from "snow" to "dry", its local model would learn wrongly based on mislabeled data, thus the global model would also be poisoned after each global aggregation round in which such an adversary participates. During the inference phase, the consequent RCC model may identify actual snow conditions as dry, an underestimation of hazardous road conditions jeopardizes transportation safety and may increase accident rates. Results in [18] and in Section 5.2 both indicate huge performance degradation for FL-RCC when TLFAs happen without defense. Moreover, compared to other more complicated attacks, such as backdoor attacks [17], TLFAs only require simpler replacement operations on labels (not replacement or manipulation of original image pixels or features), thus they are practical from an adversarial point of view, in the sense that they are efficient to launch with vehicle resources.

Existing defenses mitigating poisoning attacks face two primary challenges in the context of FL-RCC. First, general *poisoned model detection* methods, such as FoolsGold [7] and FLAME [23], struggle with the Non-Independent and Identically Distributed (Non-IID) nature of vehicular data. Vehicle driving behaviors vary across locations and over time, yielding markedly different client distributions. General detection methods [7, 23] that are not tailored to TLFAs can fail because poisoned models can still easily appear as benign in a heterogeneous environment, as legitimate updates themselves may exhibit large variance. Recent novel poisoning attack mitigation methods primarily focus on backdoor attacks [6] or untargeted attacks [38], thus they are not specifically designed for TLFAs. On the other hand, current countermeasures pay attention to model-level misbehavior detection, while missing an effective joint vehicle-level *malicious client exclusion* strategy based on model-level detection results. By uploading poisoned models, malicious clients can consistently threaten the FL-RCC system if they are not excluded. The state-of-the-art countermeasure against TLFAs for FL-RCC, FLARE [18], still has difficulties in reducing the mis-classification level to match the performance in an attack-free scenario when such attacks happen.

To close this gap, this paper proposes DEFEND: poisoned model Detection and malicious client Exclusion mechanism for FEderated learning-based road coNDition classification. DEFEND first identifies source and target classes (neurons) in TLFAs based on neuron-wise magnitude analysis of the DNN's output layer. With the two recognized neurons, two strategies are implemented in each round: 1) model parameters directly connected to source and target neurons are extracted and clustered by the Gaussian Mixture Model (GMM) to detect poisoned model contributions, which are then filtered out before aggregation; 2) two core metrics evaluating TLFAs, Source Recall (SRec)[4] and Attack Success Rate (ASR)[5], are calculated to monitor the global model performance; if these metric values achieve predefined performance change thresholds, the current global model is still considered poisoned even after the local model filter and is discarded. Finally, leveraging the aforementioned poisoned-model detection, DEFEND introduces an adaptive client rating strategy based on decision theory [27]. A client rating is decremented if its contribution is deemed poisoned in a round and incremented otherwise. Once its rating falls below a threshold, the client is identified as malicious with high confidence and it is promptly excluded from further participation in the FL training. Extensive experiments across three DNNs (ResNet-18 [8], EfficientNet-B1 [32], and Deit-Tiny [34]) and three RCC tasks (friction, material, and unevenness classification) show that the proposed DEFEND outperforms seven baselines (FedAvg [22], Krum [4], Trimmed Mean (TMean) [40], Median [40], FoolsGold [7], FLAME [23], and FLARE [18]) in terms of three evaluation metrics (Global Accuracy (GAcc), SRec, and ASR).

In brief, the main contributions of this paper are:

(1) An improved poisoned model detection strategy specifically designed for FL-RCC systems under TLFAs, based on neuron-wise magnitude analysis and GMM clustering.
(2) An enhanced malicious client exclusion strategy based on adaptive client rating using decision theory leveraging the aforementioned model-level misbehavior detection.
(3) Significantly improved defense against TLFAs compared to the state-of-the-art: DEFEND is the first to maintain model performance equal to that in an attack-free scenario, on average outperforming the best baseline defense by 2.81%, 24.99%, and 15.78% regarding GAcc, SRec, and ASR, respectively.

The rest of this paper is organized as follows. Section 2 reviews camera-based FL-RCC, corresponding TLFAs, and defenses. Section 3 describes the system and adversary models. Our scheme, DEFEND, is introduced and evaluated in Section 4 and Section 5, respectively. Finally, Section 6 concludes this paper and outlines future work.

[4]The fraction of source class samples that are correctly classified.
[5]The ratio of samples with the source label misclassified into the target class.

2 Background and Related Work

2.1 Federated Learning (FL)-based Road Condition Classification (RCC)

Given that road condition changes, for example when the weather changes, can be local and unpredictable, RCC is important for smart vehicles [20] to timely adjust their braking, steering, suspension, and other driving safety and driver assistance systems. RCC can be camera-based [25, 43], as is the focus of this paper, but it can also rely on other sensors [3, 35]. A key distinction is that cameras are and will be increasingly available, allowing car systems to perceive a gamut of conditions with one sensor (the camera), offering data and early detection, e.g., of a water puddle or cracked asphalt before hitting them; at which point, for example, an inertial sensor could have given relevant data but only post-facto.

To leverage privacy-sensitive image data and distributed on-board resources, the feasibility of FL-based RCC systems was recently explored. Specifically, FedRD [42] achieves individual-level privacy protection and high-performing hazardous road damage detection; Then, its follow-up works, FLRSC [36] and FedRSC [37], support multi-label RCC tasks in FL. However, current FL-RCC proposals primarily focus on improving the classification performance of models and the privacy guarantees for vehicles. How to secure FL-RCC systems still remains a largely open problem.

2.2 Poisoning Attacks against FL-RCC

TLFAs [16, 33] are essentially Data Poisoning Attacks (DPAs) [10, 15, 33]. Model poisoning attacks [31] demand higher levels of adversarial expertise and specialized knowledge (e.g., understanding model architectures and training configurations) as well as greater adversarial computational resources (e.g., for manipulating millions of model parameters) [9, 30]; thus they are not as practical as DPAs if launched by resource-restricted on-board vehicle platforms. Furthermore, as local models are sent to the server for aggregation, without the server accessing client local data for privacy considerations, DPAs are more difficult to detect compared to model poisoning attacks.

DPAs can be divided into untargeted DPAs and targeted DPAs [30]. The former degrades the overall performance of the global model and thus can be more easily detected and countered by the system before model deployment; the latter targets specific inputs, thus can be easier to disguise, especially in a heterogeneous environment. As some misclassifications in RCC are more dangerous than others, targeted DPAs, notably TLFAs, are more practical and significant than untargeted DPAs in our investigation here. To see why this is the case, let us revisit the above-mentioned example of road friction level: mislabeling the actual "snow" road surface as "dry" may have vehicle safety implications, while the opposite misclassification, "dry"→"snow", reduces traffic efficiency.

2.3 Defensive Mechanisms for FL-RCC

Various countermeasures are proposed to defend against general DPAs in FL, without considering neither RCC nor TLFAs. FoolsGold [7] assumes that poisoned models are more similar than benign models; thus, it first calculates the cosine similarity between local model output layers and then penalizes the updates with larger similarity to mitigate potential poisoning attacks. CONTRA [2] also measures the cosine similarity of local updates to record credibility for promotion or penalty of updates. FLAME [23] utilizes jointly differential privacy, model clustering, and weight clipping technologies, trading off global model performance for adversary detection efficiency. However, the data in FL-RCC is inherently Non-IID, as distinct spatial and temporal vehicle driving patterns lead to diverse data distributions. The intrinsic heterogeneity complicates the task of distinguishing malicious updates from benign ones [6], as innocent updates also have high variance, leading to significant countermeasure performance degradation on RCC. More refined features specific to TLFAs should be extracted, and more effective detection mechanisms should be designed, taking the mentioned heterogeneity into consideration.

Recent novel countermeasures against poisoning attacks pay attention to mitigating backdoor attacks. FreqFed [6] capitalizes on the discrete cosine transform to distinguish good and bad updates in the frequency domain. CrowdGuard [29] analyzes hidden layer outputs of local models and executes iterative pruning to detect backdoors. However, such attacks/adversaries elaborately falsify both the original image data and the corresponding labels. These attacks and their corresponding defenses are fundamentally different from TLFAs, adding triggers, and DEFEND, utilizing the whole output layer.

Only few defensive mechanisms are specifically designed for vehicles. LFGurad [30] first proposes a hierarchical FL framework for vehicular networks, then feeds the activations of the output layer in each local model into a multi-class support vector machine for malicious model classification; finally, it is evaluated on the structured traffic sign classification dataset. RoHFL [44] is another robust hierarchical FL framework that develops a logarithm-based normalization method to address maliciously scaled model parameters. OQFL [39] uses a quantum-behaved particle swarm optimization method that can automatically update hyper-parameters in FL against adversarial attacks targeted for autonomous driving. Both RoHFL and OQFL are evaluated on general classification datasets such as MNIST and Fashion-MNIST. However, they all focus on passive misbehavior detection at the model level, while ignoring active malicious client exclusion at the vehicle level.

FLARE [18] is the state-of-the-art solution designed for FL-RCC against TLFAs; however, both its HDBSCAN-based model clustering method and its count-based client filtering method are moderately effective, with a very significant RCC performance gap when comparing FLARE under attack and a TLFA-free situation. Specifically, the SRec value for a no-attack scenario is 80%; and it drops to 60% when under TLFAs, with a similar gap for ASR values, 5% vs. 30%. In contrast, DEFEND thwarts TLFAs, achieving SRec and ASR of about 80% and 5%, respectively, even under attack. All in all, under TLFAs, there is no existing countermeasure that can achieve the same FL-RCC model performance as the TLFA-free scenario.

3 System Model and Adversary Model

3.1 System Model

Vehicular Protocols for Secure and Private Communication: We consider an FL-RCC system with one trusted server and a massive set of available clients to contribute, each registered via a

cloud-based Vehicular Public Key Infrastructure (VPKI) [1, 12]. Vehicles obtain short-lived pseudonym certificates [11], syntactically unlinkable and anonymized credentials issued for minutes or hours, to guarantee authenticity, integrity, non-repudiation, and privacy (conditional anonymity and long-term unlinkability). This design choice ensures compatibility with standardized security and privacy for cooperative ITS systems, notably V2X (Vehicle to Vehicle/Infrastructure) [1, 13, 26]. Misbehavior attributed to one or more pseudonyms can lead to rapid revocation [14], thus timely efficient eviction of the wrongdoer from the system. Clients establish end-to-end confidential and authenticated channels with the server over TLS [28], using their current pseudonym, ensuring secure and privacy-preserving delivery of model updates.

FL Procedure: K clients $\mathbb{C} = \{c_1, c_2, ..., c_K\}$ form a cluster to undertake the current RCC model training task. The task involves E classes/labels and a sequence $S = [1, 2, ..., E]$ to encode them, where larger numbers represent more dangerous road condition classes. Consequently, output layer L consists of E neurons: $L = [l_1, l_2, ..., l_E]$. Each client, c_k, owns a private dataset, $\mathbb{I}_k$, with n_k image-label pairs, where images are captured by on-board cameras and could be labeled by driver feedback or annotation tools [45]. In each FL round t, a subset of clients $\mathbb{C}^t \subseteq \mathbb{C}$ are randomly selected for local training, where $|\mathbb{C}^t| = M$, $M \leq K$. After receiving the newest global model ω^t from the server, each participant $c_k \in \mathbb{C}^t$ updates the model to ω_k^t using $\mathbb{I}_k$ according to Equation (1), where η represents the learning rate, and $\mathcal{L}_k$ denotes the loss function of client c_k, e.g., cross-entropy. Finally it sends ω_k^t back to the server for aggregation.

$$\omega_k^t = \omega^t - \eta \nabla_{\omega^t} \mathcal{L}_k(\omega^t, I_k) \tag{1}$$

A new global model ω^{t+1} is formed as Equation (2), where q_k is the aggregation weight assigned to c_k. Generally speaking, q_k is the normalized data size [22]; however, to avoid malicious clients providing falsified information that could increase the attack impact, here we set q_k as $\frac{1}{M}$. It is also important to note that this paper focuses on mitigating TLFAs, rather than improving the no-attack model performance. Moreover, countermeasures against TLFAs, including ours, do not rely on specific aggregation strategies. Thus, we choose Equation (2) for security and simplicity.

$$\sum_{c_k \in \mathbb{C}^t} q_k \times \omega_k^t \tag{2}$$

These local training and global aggregation processes are executed iteratively until the global model converges. The final learned model is deployed to automated vehicles, over secure server-client communication, to support improved real-time RCC with unseen road surface images.

3.2 Adversary Model

With P malicious clients in the system, where $P \leq \frac{K}{2}$, each adversary flips its local labels from a source class f to a less hazardous target class g (without altering input features), then trains its local model on the poisoned data. After that, each adversary submits corrupted local updates to the server to poison the global model during the aggregation process. The goal of malicious clients is to selectively degrade the model performance on the source class, thereby causing vehicles to underestimate hazard levels (e.g., misclassifying snow roads as dry, as shown in Figure 1); which poses a greater safety risk than uniform performance degradation (e.g., untargeted poisoning attacks randomly flip labels without a specific misclassification goal).

To do so, f should be smaller than g (if $f > g$, the attack goal is disrupting traffic efficiency rather than safety, e.g., flipping from actual dry roads to snow). Strategies of the adversary can also be adaptive, i.e., choosing different source and/or target classes during specific rounds, aiming to bypass potential defensive mechanisms.

We assume that adversaries cannot compromise the trusted server, and they cannot control the random client selection process (choosing $\mathbb{C}^t$) at the server side. We do not dwell on the introduction of malicious clients in the system - they can be gradually registered with the system, provisioned with credentials, and modified functionality that deviates from the system specification (or similarly, they can be compromised clients, e.g., with the FL functionality adversarially modified).

4 Our Scheme: DEFEND

Scheme Overview: The poisoned model Detection and malicious client Exclusion mechanism for FEderated learning-based road coNDition classification (DEFEND) workflow in each round is illustrated in Figure 2. Algorithm 1, *Line 1* initializes a client blacklist $\mathbb{B}$, an SRec value, an ASR value, and for each client c_k a rating value $r_k(0)$. *Lines 2-8* randomly select clients not in the blacklist to execute local model training and upload updates in each round, protected by security and privacy protocols, notably pseudonymously authenticated TLS. *Lines 9-15* analyze neuron-wise magnitudes regarding output layer L to identify source and target classes as f' and g'. *Lines 16-17* detect poisoned models via a Gaussian Mixture Model (GMM) based on U^t: value changes of parameters connected to f' and g'. *Lines 18-24* validate the new global model, ω^{t+1}, based on SRec and ASR values to decide accept or discard it. *Lines 25-34* update rating values and the blacklist to exclude malicious clients. Each of the detected outliers in $\mathbb{C}_{out}^t$ sends model parameters over each of the secure channels in a non-repudiable manner. Given the use of a valid pseudonym (contributions of a given client are anonymized yet they can be linked to each other across FL rounds), client rating can be reduced, so that a deemed malicious client can be excluded from current and future FL processes, while at the same time rendering a client participation in different FL executions unlinkable.

Poisoned Model Detection: Poisoned models trained under TLFAs and honest models trained normally have contradictory objectives [9], resulting in more significant differences for those parameters directly connected to source and target neurons. Thus, in each round, after getting the value changes regarding output layer as $\Delta_{k,L}^t = \{\omega_{k,l}^t - \omega_l^t | l \in L\}$ for each c_k, we calculate the magnitude (ℓ_2-norm) for each output neuron l as $\|\Delta_{k,l}^t\|_2$ according to Equation (3), where d_l is the number of parameters associated with output neuron l.

$$\|\Delta_{k,l}^t\|_2 = \sqrt{\sum_{i=1}^{d_l} \left(\Delta_{k,l,i}^t\right)^2}, \qquad k \in C_t,\ l \in L. \tag{3}$$

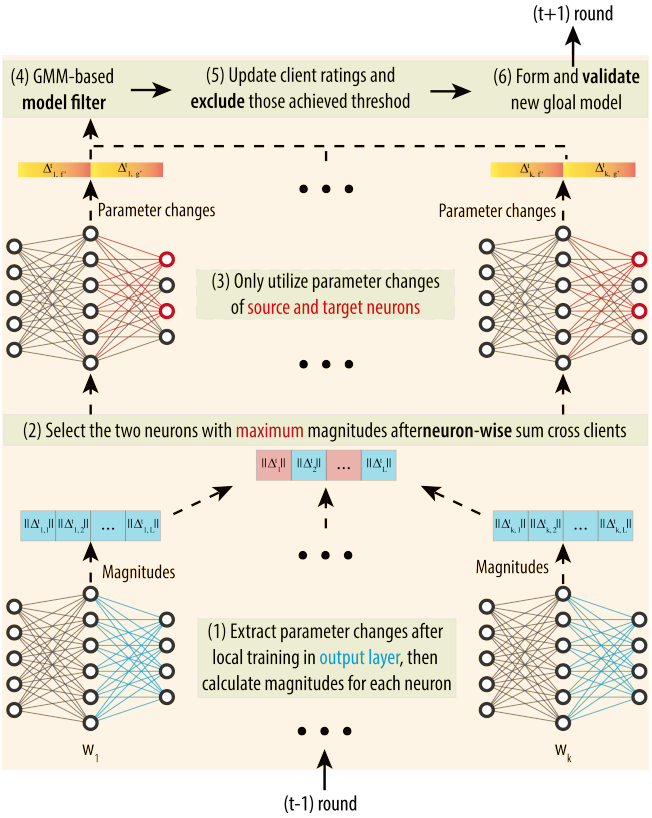

Figure 2: Workflow of DEFEND in round t. Steps (1) and (3) extract features and are executed in parallel for each local model. Step (2) identifies source and target neurons in TL-FAs. Steps (4)-(6) execute the local model filtering, malicious client exclusion, and global model validation, respectively. Note that output layer parameters are marked in blue, while source and target neuron parameters are marked in red.

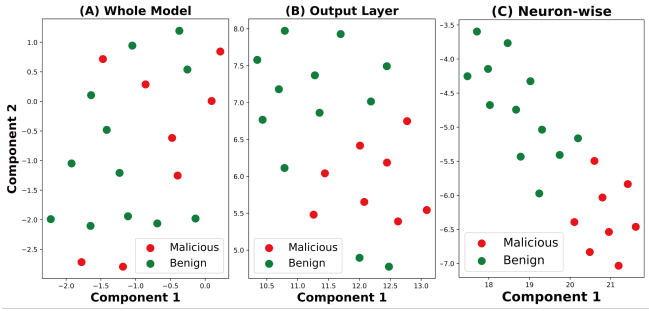

Figure 3: Comparison between malicious and benign updates based on three kinds of features: (A) whole model parameters, (B) output layer parameters, and (C) neuron-wise parameters (with two more distinctive clusters).

Then, we sum up all local model magnitudes for l as $||\Delta_l^t||_2$ according to Equation (4). The two neurons with the highest accumulated magnitudes, denoted as f' and g' ($f' < g'$), are recognized as source and target neurons.

Algorithm 1 Protocol of DEFEND

Initialize black list $\mathbb{B} = \emptyset$, $SRec^{old} = 0$, $ASR^{old} = 1$, and rating value $r_k(0) = \delta(r^{max} - r^{min})$ for each client v_k
for each round $t \in [1, T]$ **do**
  $\mathbb{C}^t \leftarrow$ randomly select M clients from $\mathbb{C} - \mathbb{B}$
  The server sends ω^t to all clients in $\mathbb{C}^t$
  for each client $c_k \in \mathbb{C}^t$ in parallel **do**
    Update local model ω_k^t
    Send ω_k^t back to the server
  end for
  The server receives ω_k^t from $\mathbb{C}^t$
  for each ω_k^t the server **do**
    $\Delta_{k,L}^t = \{\omega_{k,l}^t - \omega_l^t | l \in L\}$ ▷ Output layer changes
    Calculate magnitudes $||\Delta_{k,l}^t||_2$ for $l \in L$ ▷ ℓ_2-norm
  end for
  $\{||\Delta_{l_1}^t||_2, ..., ||\Delta_{l_E}^t||_2\} \leftarrow$ neuron-wise magnitudes
  $f', g' \leftarrow$ Top-2($\{||\Delta_{l_1}^t||_2, ..., ||\Delta_{l_E}^t||_2\}$) ▷ $f' < g'$
  $U^t \leftarrow \{\Delta_{k,l}^t | c_k \in \mathbb{C}^t, l \in \{g', f'\}\}$
  $\mathbb{C}_{out}^t = \text{GMM}(U^t)$
  $\omega^{t+1} = Aggeregate\{\omega_k^t | c_k \notin \mathbb{C}_{out}^t\}$
  $SRec^{new}, ASR^{new} \leftarrow$ Validate (ω^{t+1})
  $\Delta SRec = SRec^{new} - SRec^{old}$, $\Delta ASR = ASR^{new} - ASR^{old}$
  if $\Delta SRec < SRec^{thr}$ or $\Delta ASR > ASR^{thr}$ **then**
    $\omega^{t+1} = \omega^t$
  end if
  $SRec^{old} = SRec^{new}$, $ASR_k^{old} = ASR_k^{new}$
  for $c_k \in \mathbb{C}^t$ **do**
    if $c_k \in \mathbb{C}_{out}^t$ **then**
      $r_k(t) = max\{r_k(t-1) - \gamma, r_{min}\}$
      if $r_k(t) \le r^{min}$ *And* $c_k \notin \mathbb{B}$ **then**
        Add c_k in $\mathbb{B}$
      end if
    else
      $r_k(t) = min\{r_k(t-1) + \beta, r^{max}\}$
    end if
  end for
end for
return ω^{T+1}

$$||\Delta_l^t||_2 = \sum_{k \in C_t} ||\Delta_{k,l}^t||_2, \qquad l \in L. \tag{4}$$

Changes of parameters directly connected to the two neurons of each local model, $U^t = \{\Delta_{k,l}^t | c_k \in \mathbb{C}^t, l \in \{g', f'\}\}$, are extracted as critical features and fed into GMM to form two clusters: one good local model cluster and one bad local model cluster. Using the Uniform Manifold Approximation and Projection (UMAP) [21] dimension reduction method, Figure 3 visualizes the comparison between malicious and benign updates in FL-RCC based on three different feature types. It shows that, compared to utilizing the parameters for the entire model or the output layer parameters, leveraging neuron-wise parameters is more effective in distinguishing poisoned models under TLFAs from benign ones, as two more obvious clusters are observed.

Compared to hard clustering methods, such as KMeans [9] and HDBSCAN [18], GMM applies soft probabilistic clustering, avoiding rigid decision boundaries, thus making it suitable for heterogeneous environments (we consider this issue both in feature extraction to ensure detection accuracy). When the distinction between poisoned and benign models is blurry, GMM captures this uncertainty by modeling the underlying distribution of model parameters, enabling more flexible and calibrated detection. The denser cluster is identified as the bad one (as in [9] and as shown in Figure 3), so its local models will be filtered out before global aggregation.

Moreover, once source and target classes (the attack goal) are known via the described neuron-wise magnitude analysis, we can calculate SRec and ASR during the training to consistently monitor global model performance regarding the two classes. If the SRec value drops or the ASR value increases significantly compared to the values in the last round, i.e., value changes achieve the predefined thresholds $SRec^{thr}$ and ASR^{thr}, we discard this round's global model for robustness, as it may still be poisoned even after discarding the deemed adversarial local model contributions.

Malicious Client Exclusion: To reduce the damage to the global model caused by data poisoning by malicious clients across FL rounds, we maintain a rating score $r_k \in [r^{min}, r^{max}]$ for each client c_k, which is updated in each round t according to Equation (5), where $\beta, \gamma \in (0, r^{max}]$ are both constants that control the corresponding rating reward and penalty steps, respectively.

$$r_k(t) = \begin{cases} \min\{\, r_k(t-1) + \beta,\ r^{\max} \,\}, & \text{if } \omega_k^t \text{ benign,} \\ \max\{\, r_k(t-1) - \gamma,\ r^{\min} \,\}, & \text{if } \omega_k^t \text{ poisoned.} \end{cases} \tag{5}$$

The r_k value decreases if the model parameters uploaded by c_k are deemed adversarial in the current round, and increases if the detection result is deemed benign. The update rule accumulates per-client behavior over time so that transient or noisy detection errors do not immediately lead to exclusion. Once $r_k \leq r^{min}$, c_k is identified as malicious with high confidence, thus excluded from the future client selection process, avoiding malicious clients consistently poisoning the global model training process. Compared to the count-based strategy [18], the cumulative score reduces false positives caused by temporary conditions (e.g., small local data, stochastic training effects) while remaining sensitive to persistent, adversarial behavior.

Complexity Analysis: Assume the dimensionalities of the whole DNN model, the output layer of DNN, and one neuron in the output layer are d_w, d_o, and d_e, respectively. Note that $d_w \gg d_o \gg d_e$. The computation overhead of DEFEND in each round includes the following parts: 1) $O(Md_o)$ to calculate the output layer changes of M clients; 2) $O(MEd_e)$ to compute the neuron-wise magnitudes of E output neurons and M clients; 3) $O(E \log E)$ to identify the source and target neurons from E neurons; 4) $O(Md_e)$ to cluster neuron-wise parameters of M clients via GMM; and 5) $O(M)$ to maintain blacklist and rating values for M clients. Such that, the overall complexity of DEFEND is $O(Md_o)$. Compared to other countermeasures based on the entire model, the output layer, or K-Means, e.g., Median ($O(M \log Md_w)$), TMean ($O(M \log Pd_w)$), Krum ($O(M^2 d_w)$), and FoolsGold ($O(M^2 d_o)$), in brief, DEFEND is computation-efficient.

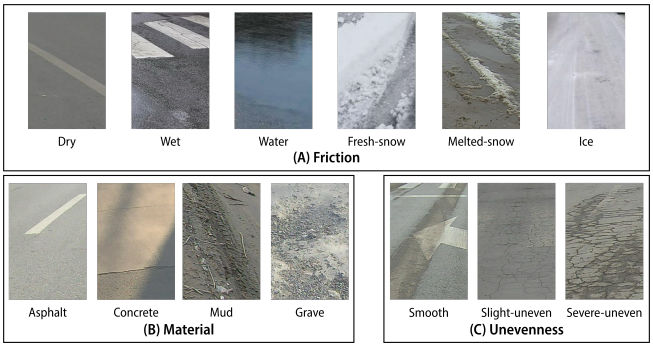

Figure 4: Image examples of the RSCD dataset.

Table 1: Default model training configurations in this paper.

Term	Value
Local epoch	3
Global round	60
Learning rate (lr)	0.03
Momentum for lr	0.5
Optimizer	SGD
Batch size	64
Loss function	Cross-Entropy

Table 2: Practical model information of the Friction task.

Model Version	Model Size (MB)	Inference Time (ms)	Memory Usage (MB)
ResNet-18	21.32	2.18	150.52
EfficientNet-B1	12.44	9.16	237.30
DeiT-Tiny	10.54	4.38	97.55

Practical Considerations: As mentioned in the system model, we leverage standardized V2X security and privacy protocols for DEFEND, notably VPKI and pseudonyms, to ensure unlinkability, authenticity, and non-repudiation. Current countermeasures primarily focus on model-level misbehavior detection, without misbehaving client exclusion. With the mentioned V2X protocols, DEFEND can smoothly implement the client rating and exclusion strategy after the poisoned model detection strategy. Moreover, the neuron-wise magnitude analysis can identify the attack goal of TLFAs, even if adaptive adversaries change their goals during the training. DEFEND takes full advantage of such information for critical feature extraction and local model performance validation in each round, thus maximizing its mitigation effect.

5 Evaluation

5.1 Experimental Setup

We simulate 100 clients and one server in the PyTorch environment, using an NVIDIA A100 GPU (with 40GB memory) and an Icelake CPU (with 128GB memory) for the entire system. By default, 30 clients are malicious, and 20 clients are randomly selected as participants in a round, out of the total of 100 simulated clients. We evaluate each method's performance on three RCC tasks (Friction, Material, and Unevenness) based on the Road Surface Classification Dataset (RSCD)[6] that comprises one million real-world road

[6] https://thu-rsxd.com/rscd/

Table 3: The overall evaluation results within default configurations. All values are ratios in %.

Model	Method	RCC @ Friction			RCC @ Material			RCC @ Unevenness		
		GAcc ↑	SRec ↑	ASR ↓	GAcc ↑	SRec ↑	ASR ↓	GAcc ↑	SRec ↑	ASR ↓
ResNet-18 [8]	FedAvg-NA‡[22]	85.26	72.28	5.84	80.36	82.24	3.07	74.65	67.90	9.93
	FedAvg [22]	72.83	44.88	30.85	70.17	35.49	44.13	69.14	46.74	50.34
	Krum [4]	81.44	48.89	22.92	58.87	23.09	21.84	48.54	45.70	53.38
	TMean [40]	80.78	45.84	18.48	55.75	34.27	40.43	52.65	44.37	25.23†
	Median [40]	81.56	48.44	22.52	73.87	46.85	25.07	65.45	47.33	34.55
	FoolsGold [7]	81.64	50.64	15.56	72.53	42.53	35.20	70.71†	49.35	26.33
	FLAME [23]	63.32	51.20	21.28	52.51	40.66	45.20	41.36	16.29	69.52
	FLARE [18]	82.80†	61.72†	14.64†	77.09†	64.08†	14.69†	67.98	55.13†	28.13
	DEFEND (Ours)	**84.93***	**74.20**	**2.84**	**78.61**	**80.11**	**2.88**	**73.71**	**79.85**	**7.32**
EfficientNet-B1 [32]	FedAvg-NA	86.08	75.68	3.40	80.15	79.25	4.75	74.48	80.53	8.08
	FedAvg	80.48	38.12	32.96	74.63	51.17	25.20	65.48	39.42	29.42
	Krum	64.49	4.92	69.32	52.64	42.88	20.13	66.98	43.67	26.65
	TMean	81.46	46.52	27.04	69.30	26.27	47.39	66.86	43.55	26.78
	Median	82.54	53.88	22.16	75.49	54.59	20.19	66.63	44.93	28.02
	FoolsGold	83.64†	59.80	16.76	77.84†	61.15	17.84	66.48	38.32	26.32
	FLAME	82.56	52.36	20.96	59.19	31.97	14.32	70.47	50.25	20.15
	FLARE	83.46	62.04†	16.08†	77.56	65.73†	12.88†	70.60†	59.27†	17.65†
	DEFEND (Ours)	**84.43**	**80.40**	**5.40**	**77.87**	**82.99**	**3.31**	**72.87**	**79.13**	**7.98**
Deit-Tiny [34]	FedAvg-NA	86.54	77.32	3.44	80.31	80.61	3.79	74.08	78.80	7.02
	FedAvg	77.89	25.52	48.40	72.83†	44.88	30.85	60.93	30.03	45.47
	Krum	67.15	17.48	38.04	72.82	50.48	31.81	66.94	40.73	27.60
	TMean	64.03	34.88	40.84	47.37	41.76	45.55	64.93	41.73	37.50
	Median	78.91	33.64	45.72	51.81	47.33	41.28	**77.01**	55.72†	14.65†
	FoolsGold	82.61	53.60	25.08	**76.85**	65.23†	14.75†	68.59	45.23	17.33
	FLAME	57.54	45.60	25.84	42.36	43.57	19.92	60.34	49.47	33.63
	FLARE	83.02†	58.84†	17.80†	72.19	52.27	29.09	68.77	53.70	18.88
	DEFEND (Ours)	**84.58**	**80.76**	**4.84**	65.55	**82.11**	**4.75**	69.14†	**78.52**	**7.63**

* **Bold numbers are the best performances in a group.**

† Numbers with underline are the second-best values in a group.

‡ NA denotes No Attack. Others without this symbol are all under TLFAs.

images acquired from on-board cameras. We create non-IID data partitioning for the 100 clients following the state of the art [18] from a Dirichlet(α) distribution with $\alpha = 1.0$ by default.

For computational efficiency, all images are resized to 224×224×3. Three task-specific subsets include: 1) Friction: 58,800 training and 14,550 test samples across six classes (dry, wet, water, fresh-snow, melted-snow, and ice); 2) Material: 57,000 training and 15,000 test images spanning four surface types (asphalt, concrete, mud, and gravel); and 3) Unevenness: 57,542 training and 18,000 test images labeled by three degrees of roughness (smooth, slight-uneven, and severe-uneven). During training, each adversarial client applies TLFAs by relabeling its local samples as follows: water→dry (Friction), gravel→asphalt (Material), and severe-uneven→smooth (Unevenness). Image examples of each subset are provided in Figure 4. The global test sets remain untouched and serve solely for inference. Model training details, such as learning rate and batch size, are summarized in Table 1 as in [18].

We adopt three lightweight (suitable for vehicular deployment) DNN models: ResNet-18 [8], EfficientNet-B1 [32], and DeiT-Tiny [34]. The model size, inference time per image, and memory usage of each model completing the Friction task are summarized in Table 2. We evaluate seven schemes in the experiment: FedAvg [22] (not countering adversarial behavior but the basis for other baseline schemes), Krum [4], TMean and Median [40], FoolsGold [7], FLAME [23], FLARE [18] and our proposed DEFEND ($SRec^{thr} = 0.1$, $ASR^{thr} = 0.1$, $r^{max} = 1.00$, $r^{min} = 0.00$, $r_k(0) = 0.80$, $\beta = 0.05$, and $\gamma = 0.20$; these parameters are chosen empirically). All schemes share the same training configurations in Table 2 for fair comparison. Performance is quantified by three metrics: GAcc, SRec, and ASR.

5.2 Results Analysis

Baseline Results: As per Table 3, compared to FedAvg-NA (No Attack), FedAvg performance under TLFAs drops significantly, a 37.60% average reduction in SRec and a 32.00% average increase in ASR, indicating that FL-RCC is vulnerable to TLFAs. Although the six baseline countermeasures (Krum, TMean, Median, FoolsGold, FLAME, and FLARE) improve performance compared to FedAvg, there is still a huge performance gap to the FedAvg-NA performance. Take ResNet-18 with Unevenness as an example, the SRec values of the baseline countermeasures range from 16.29% to 55.13%, while the ASR values range from 25.32% to 69.52%. Even with the best of the baseline defenses, there is still significant risk: more than half of severe-uneven road conditions are recognized as smooth conditions, threatening safety. Specifically, the best baseline performance in each group of Table 3 (per model type and task type for fair comparison) is still worse than FedAvg-NA on average by 16.32% and 11.01% in terms of SRec and ASR, respectively. Such results indicate that the TLFAs mitigation of FL-RCC by existing countermeasures is limited, with stronger defense required before FL-RCC deployment.

DEFEND Results: In contrast, our method, DEFEND, remarkably boosts SRec and reduces ASR at the same time. On average, 2.81%, 24.99%, and 15.78% improvement against the best baseline in each group are observed for DEFEND regarding GAcc, SRec,

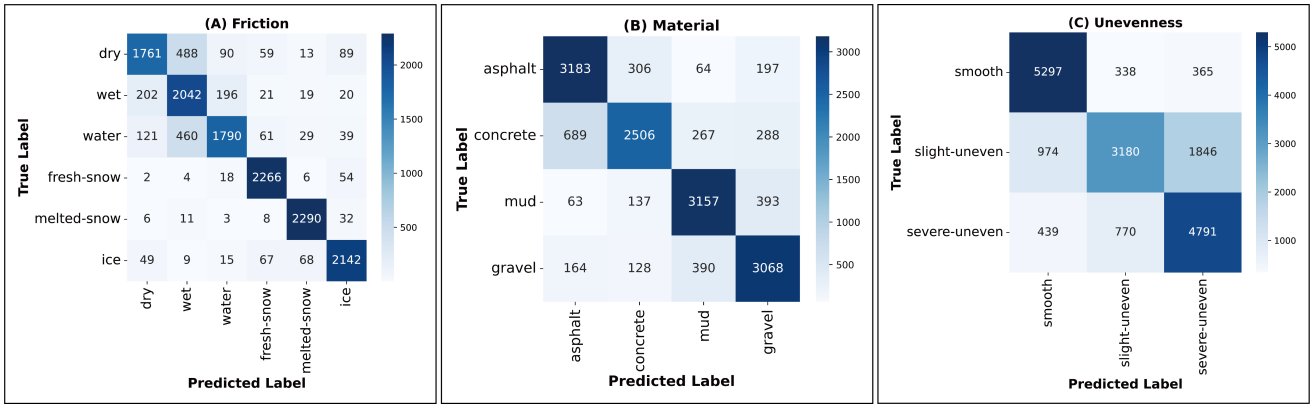

Figure 5: Confusion matrices of DEFEND with ResNet-18 in three RCC tasks.

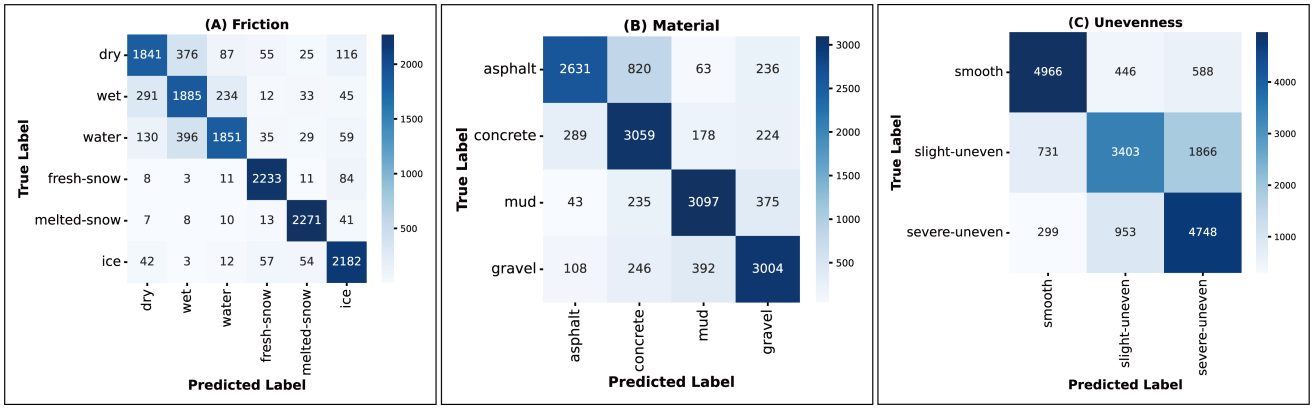

Figure 6: Confusion matrices of DEFEND with EfficientNet-B1 in three RCC tasks.

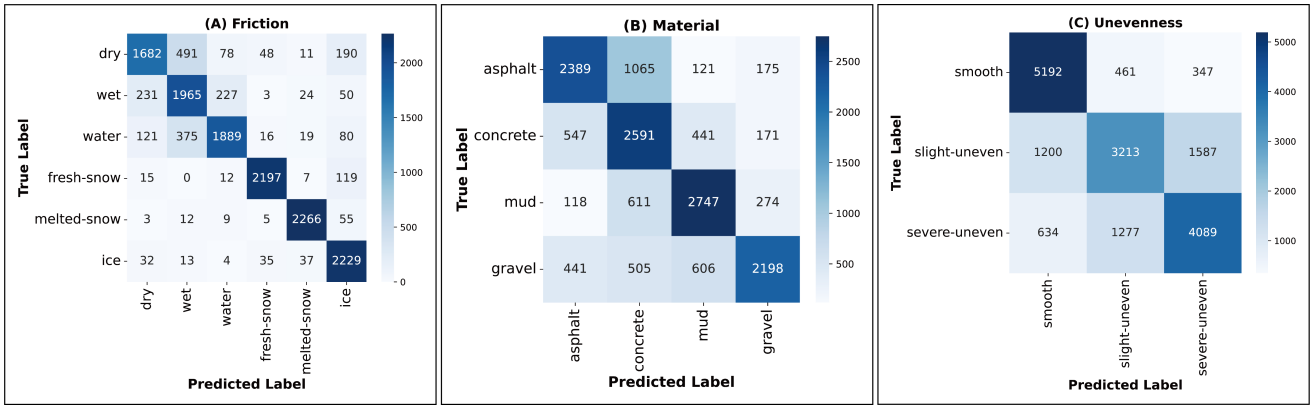

Figure 7: Confusion matrices of DEFEND with Deit-Tiny in three RCC tasks.

and ASR, respectively. The results show that DEFEND can accurately detect poisoned models and timely exclude malicious clients. Surprisingly, the average SRec and ASR values of DEFEND are 79.79% and 5.22%, respectively, even slightly better than FedAvg-NA (77.18% and 5.48%, respectively), further indicating the effectiveness of DEFEND. The confusion matrices of DEFEND in three tasks are provided in Figure 5 (with ResNet-18), Figure 6 (with EfficientNet-B1), and Figure 7 (Deit-Tiny). These confusion matrices clearly show for all classes, including source and target classes per RCC task being the attack goals (water→dry, gravel→asphalt, and severe-uneven→smooth), have high prediction performance (diagonal values). Such results further indicate that DEFEND can effectively prevent attackers from achieving goals that threaten transportation safety. Moreover, the nine confusion matrices show that DEFEND can work well for all three RCC tasks and three DNN models, indicating the stability and compatibility of DEFEND.

Impact of Malicious Client Rate: DEFEND is robust with ASR values always at a very low level as malicious client rate increases, for all three DNN models and three RCC tasks. As shown in Figure 8 (with ResNet-18), Figure 9 (with EfficientNet-B1), and Figure 10

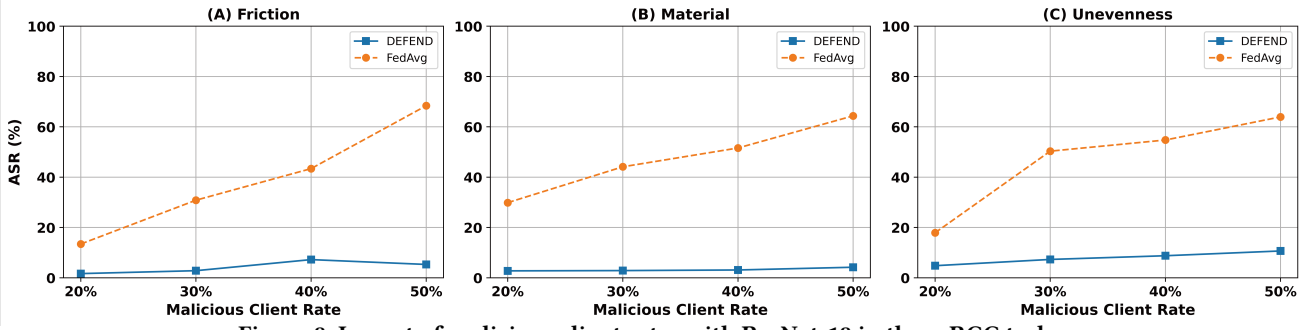

Figure 8: Impact of malicious client rates with ResNet-18 in three RCC tasks.

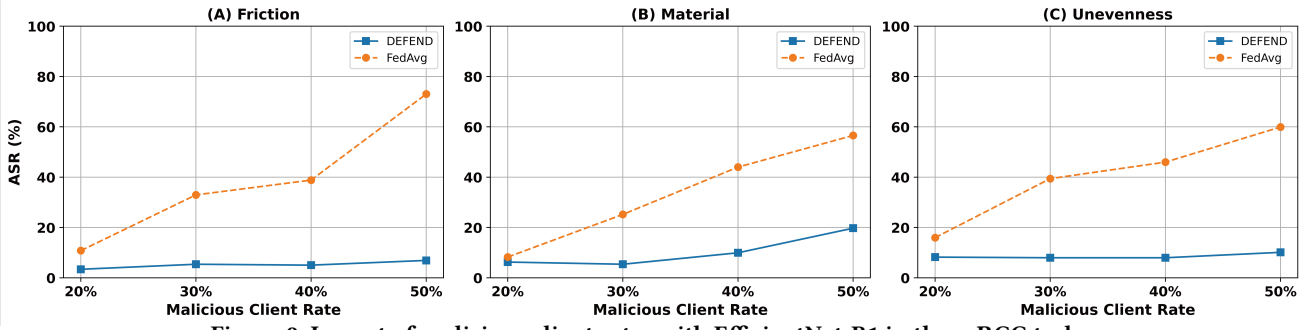

Figure 9: Impact of malicious client rates with EfficientNet-B1 in three RCC tasks.

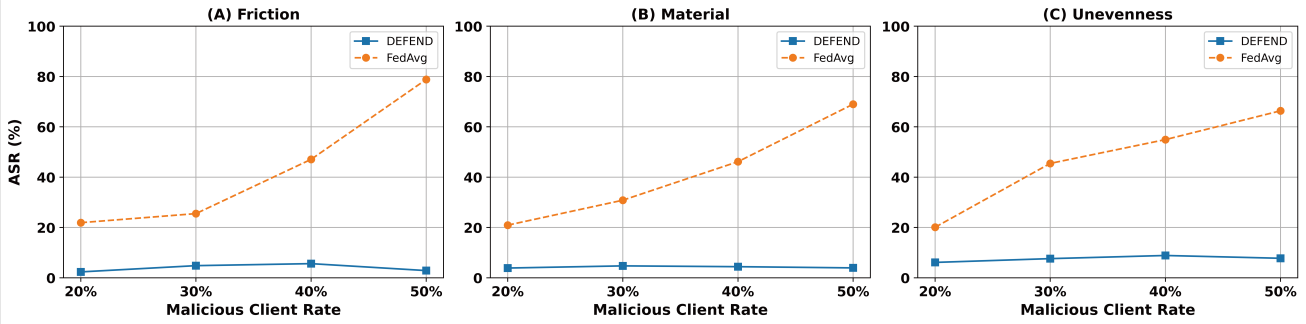

Figure 10: Impact of malicious client rates with Deit-Tiny in three RCC tasks.

(with Deit-Tiny), when malicious client rates gradually increase from 20% to 50% with a step of 10%, the ASR for DEFEND does not significantly grow accordingly. Take EfficientNet-B1 for example: ASR ranges are [3.40%, 6.92%] for the Friction, [5.40%, 19.73%] for the Material, and [7.98%, 10.13%] for the Unevenness tasks. Even when half of clients are malicious, the model validation process in DEFEND, along with the preceding neuron-wise magnitude analysis, can still detect severely poisoned global models and discard them. However, the FedAvg ASR performance deteriorates seriously as the malicious client rate rises: 73.04% (from 10.80%) for the friction, 56.59% (from 8.24%) for the Material, and 59.93% (from 15.97%) for the Unevenness tasks. Increased malicious client rates significantly increase TLFA impact, while DEFEND can still work well even when the malicious client rate is very high, regardless of tasks and models.

Observations on Model Type: ResNet-18, as a lightweight Convolutional Neuron Network (CNN), suffers considerable degradation without defense mechanisms in place, while it benefits the most from DEFEND in terms of ASR. On average, EfficientNet-B1 delivers the strongest baseline performance in no-attack settings, but also experiences the steepest drops under attack. In contrast, the lightweight Transformer model, DeiT-Tiny, is highly vulnerable, with ASR frequently exceeding 40% under FedAvg; while DEFEND markedly reduces ASR to match attack-free baselines, its performance remains less stable than CNN counterparts. These results show that CNNs offer increased robustness for FL-RCC, whereas Transformers require stronger defenses for comparable robustness.

6 Conclusion

This paper proposes a defensive mechanism, DEFEND, to secure FL-RCC systems against TLFAs. In each round, DEFEND detects poisoned models through neuron-wise magnitude analysis and GMM, and it validates global model performance after recognizing source and target classes. Moreover, based on model-level detection results, DEFEND adaptively rates clients and excludes those distrusted clients. Extensive evaluations involving various models, tasks, baselines, and metrics indicate the superiority of DEFEND

over the state of the art: DEFEND under attack maintains the same model performance as in an attack-free scenario. Our future work shall extend DEFEND to autonomous driving tasks beyond RCC, e.g., behavioral intention prediction, explore machine unlearning and knowledge distillation to correct the already poisoned global model, and implement DEFEND in real vehicle environments, e.g., using NVIDIA Jetson-based edge devices.

References

[1] IEEE Std 1609.2. 2023. IEEE Standard for Wireless Access in Vehicular Environments–Security Services for Application and Management Messages. *IEEE Std 1609.2-2022 (Revision of IEEE Std 1609.2-2016)* (2023), 1–349.

[2] Sana Awan, Bo Luo, and Fengjun Li. 2021. CONTRA: Defending Against Poisoning Attacks in Federated Learning. In *ESORICS* (Virtual).

[3] Akanksh Basavaraju, Jing Du, Fujie Zhou, and Jim Ji. 2020. A Machine Learning Approach to Road Surface Anomaly Assessment Using Smartphone Sensors. *IEEE Sensors Journal* 20, 5 (2020), 2635–2647.

[4] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with Adversaries: Byzantine Tolerant Gradient Descent. In *NeurIPS* (Long Beach, CA, USA).

[5] Yuyi Chen, Shichun Yang, Rui Wang, Zhuoyang Li, Qiuyue Li, Zexiang Tong, Yaoguang Cao, and Fan Zhou. 2025. Enhancing Road Surface Recognition via Optimal Transport and Metric Learning in Task-Agnostic Intelligent Driving Environments. *Expert Systems with Applications* 266 (2025), 125978.

[6] Hossein Fereidooni, Alessandro Pegoraro, Phillip Rieger, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. 2024. FreqFed: A Frequency Analysis-Based Approach for Mitigating Poisoning Attacks in Federated Learning. In *NDSS* (San Diego, CA, USA).

[7] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. 2020. The Limitations of Federated Learning in Sybil Settings. In *RAID* (San Sebastian, Spain).

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR* (Las Vegas, NV, USA).

[9] Najeeb Moharram Jebreel, Josep Domingo-Ferrer, David Sánchez, and Alberto Blanco-Justicia. 2024. LFighter: Defending against the Label-flipping Attack in Federated Learning. *Neural Networks* 170 (2024), 111–126.

[10] Yifeng Jiang, Weiwen Zhang, and Yanxi Chen. 2023. Data Quality Detection Mechanism Against Label Flipping Attacks in Federated Learning. *IEEE Transactions on Information Forensics and Security* 18 (2023), 1625–1637.

[11] Mohammad Khodaei, Hongyu Jin, and Panagiotis Papadimitratos. 2018. SECMACE: Scalable and Robust Identity and Credential Management Infrastructure in Vehicular Communication Systems. *IEEE Transactions on Intelligent Transportation Systems* 19, 5 (2018), 1430–1444.

[12] Mohammad Khodaei, Hamid Noroozi, and Panos Papadimitratos. 2023. SECMACE+: Upscaling Pseudonymous Authentication for Large Mobile Systems. *IEEE Transactions on Cloud Computing* 11, 3 (2023), 3009–3026.

[13] Mohammad Khodaei and Panos Papadimitratos. 2015. The Key to Intelligent Transportation: Identity and Credential Management in Vehicular Communication Systems. *IEEE Vehicular Technology Magazine* 10, 4 (2015), 63–69.

[14] Mohammad Khodaei and Panos Papadimitratos. 2021. Scalable & Resilient Vehicle-Centric Certificate Revocation List Distribution in Vehicular Communication Systems. *IEEE Transactions on Mobile Computing* 20, 7 (2021), 2473–2489.

[15] Denise-Phi Khuu, Michael Sober, Dominik Kaaser, Mathias Fischer, and Stefan Schulte. 2024. Data Poisoning Detection in Federated Learning. In *ACM SAC* (Avila, Spain).

[16] Léo Lavaur, Yann Busnel, and Fabien Autrel. 2024. Systematic Analysis of Label-flipping Attacks against Federated Learning in Collaborative Intrusion Detection Systems. In *ARES* (Vienna, Austria).

[17] Songze Li and Yanbo Dai. 2024. BackdoorIndicator: Leveraging OOD Data for Proactive Backdoor Detection in Federated Learning. In *USENIX Security* (Philadelphia, PA, USA).

[18] Sheng Liu and Panos Papadimitratos. 2025. Safeguarding Federated Learning-based Road Condition Classification. In *IEEE CNS* (Avignon, France).

[19] Sheng Liu, Linlin You, Rui Zhu, Bing Liu, Rui Liu, Han Yu, and Chau Yuen. 2024. AFM3D: An Asynchronous Federated Meta-Learning Framework for Driver Distraction Detection. *IEEE Transactions on Intelligent Transportation Systems* 25, 8 (2024), 9659–9674.

[20] Fanny Malin, Ilkka Norros, and Satu Innamaa. 2019. Accident Risk of Road and Weather Conditions on Different Road Types. *Accident Analysis & Prevention* 122 (2019), 181–188.

[21] Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426* (2020).

[22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient Learning of Deep Networks from Decentralized Data. In *AISTATS* (Fort Lauderdale, FL, USA).

[23] Thien Duc Nguyen, Phillip Rieger, Roberta De Viti, Huili Chen, Björn B Brandenburg, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, et al. 2022. FLAME: Taming Backdoors in Federated Learning. In *USENIX Security* (Boston, MA, USA).

[24] Marcus Nolte, Nikita Kister, and Markus Maurer. 2018. Assessment of Deep Convolutional Neural Networks for Road Surface Classification. In *ITSC* (Maui, HI, USA).

[25] Mohammad Otoofi, Leo Laine, Leon Henderson, William J. B. Midgley, Laura Justham, and James Fleming. 2024. FrictionSegNet: Simultaneous Semantic Segmentation and Friction Estimation Using Hierarchical Latent Variable Models. *IEEE Transactions on Intelligent Transportation Systems* 25, 12 (2024), 19785–19795.

[26] Panagiotis Papadimitratos, Levente Buttyan, Tamas Holczer, Elmar Schoch, Julien Freudiger, Maxim Raya, Zhendong Ma, Frank Kargl, Antonio Kung, and Jean-Pierre Hubaux. 2008. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine* 46, 11 (2008), 100–109.

[27] Panos Papadimitratos and Z.J. Haas. 2006. Secure Data Communication in Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications* 24, 2 (2006), 343–356.

[28] Eric Rescorla. 2018. *The transport layer security (TLS) protocol version 1.3.* Technical Report.

[29] Phillip Rieger, Torsten Krauß, Markus Miettinen, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. 2024. CrowdGuard: Federated Backdoor Detection in Federated Learning. In *NDSS* (San Diego, CA, USA).

[30] KM Sameera, P Vinod, Rafidha Rehiman KA, and Mauro Conti. 2024. LFGurad: A Defense against Label Flipping Attack in Federated Learning for Vehicular Network. *Computer Networks* 254 (2024), 110768.

[31] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. In *NDSS* (Virtual).

[32] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *ICML* (Long Beach, CA, USA).

[33] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data Poisoning Attacks against Federated Learning Systems. In *ESORICS* (Guildford, UK).

[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training Data-efficient Image Transformers & Distillation through Attention. In *ICML* (Virtual).

[35] Braian Varona, Ariel Monteserin, and Alfredo Teyseyre. 2020. A Deep Learning Approach to Automatic Road Surface Monitoring and Pothole Detection. *Personal and Ubiquitous Computing* 24, 4 (2020), 519–534.

[36] Ioannis V. Vondikakis, Ilias E. Panagiotopoulos, and George J. Dimitrakopoulos. 2023. An Adaptive Federated Learning Framework for Intelligent Road Surface Classification. In *ITSC* (Bilbao, Spain).

[37] Ioannis V. Vondikakis, Ilias E. Panagiotopoulos, and George J. Dimitrakopoulos. 2024. FedRSC: A Federated Learning Analysis for Multi-Label Road Surface Classifications. *IEEE Open Journal of Intelligent Transportation Systems* 5 (2024), 433–444.

[38] Ning Wang, Yang Xiao, Yimin Chen, Yang Hu, Wenjing Lou, and Y. Thomas Hou. 2022. FLARE: Defending Federated Learning against Model Poisoning Attacks via Latent Space Representations. In *ASIA CCS* (Nagasaki, Japan).

[39] Waleed Yamany, Nour Moustafa, and Benjamin Turnbull. 2023. OQFL: An Optimized Quantum-Based Federated Learning Framework for Defending Against Adversarial Attacks in Intelligent Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems* 24, 1 (2023), 893–903.

[40] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust Distributed Learning: Towards Optimal Statistical Rates. In *ICML* (Stockholm, Sweden).

[41] Linlin You, Sheng Liu, Bingran Zuo, Chau Yuen, Dusit Niyato, and H Vincent Poor. 2023. Federated and Asynchronized Learning for Autonomous and Intelligent Things. *IEEE Network* 38, 2 (2023), 286–293.

[42] Yachao Yuan, Yali Yuan, Thar Baker, Lutz Maria Kolbe, and Dieter Hogrefe. 2021. FedRD: Privacy-preserving Adaptive Federated Learning Framework for Intelligent Hazardous Road Damage Detection and Warning. *Future Generation Computer Systems* 125 (2021), 385–398.

[43] Tong Zhao, Junxiang He, Jingcheng Lv, Delei Min, and Yintao Wei. 2023. A Comprehensive Implementation of Road Surface Classification for Vehicle Driving Assistance: Dataset, Models, and Deployment. *IEEE Transactions on Intelligent Transportation Systems* 24, 8 (2023), 8361–8370.

[44] Hongliang Zhou, Yifeng Zheng, Hejiao Huang, Jiangang Shu, and Xiaohua Jia. 2023. Toward Robust Hierarchical Federated Learning in Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems* 24, 5 (2023), 5600–5614.

[45] Yijie Zhou, Likun Cai, Xianhui Cheng, Qiming Zhang, Xiangyang Xue, Wenchao Ding, and Jian Pu. 2024. OpenAnnotate2: Multi-Modal Auto-Annotating for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles* (2024), 1–13.