

Self-supervised federated GNSS spoofing detection with opportunistic data

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

Abstract—Global navigation satellite systems (GNSSs) are vulnerable to spoofing attacks, with adversarial signals manipulating the location or time information of receivers, potentially causing severe disruptions. The task of discerning the spoofing signals from benign ones is naturally relevant for machine learning, thus recent interest in applying it for detection. While deep learning-based methods are promising, they require extensive labeled datasets, consume significant computational resources, and raise privacy concerns due to the sensitive nature of position data. This is why this paper proposes a self-supervised federated learning framework for GNSS spoofing detection. It consists of a cloud server and local mobile platforms. Each mobile platform employs a self-supervised anomaly detector using long short-term memory (LSTM) networks. Labels for training are generated locally through a spoofing-deviation prediction algorithm, ensuring privacy. Local models are trained independently, and only their parameters are uploaded to the cloud server, which aggregates them into a global model using FedAvg. The updated global model is then distributed back to the mobile platforms and trained iteratively. The evaluation shows that our self-supervised federated learning framework outperforms position-based and deep learning-based methods in detecting spoofing attacks while preserving data privacy.

Index Terms—Secure localization, GNSS spoofing detection, federated learning

I. INTRODUCTION

Global navigation satellite systems (GNSSs) provided location and time information is integrated into many aspects of everyday life, with applications ranging from autonomous vehicles to mobile map navigation. However, GNSS is vulnerable to spoofing attacks, with false satellite signals tampering with the location or time information of the GNSS receivers. This can result in misleading navigation [1], making an autonomous vehicle crash [2], or disrupting timing, e.g., in power systems [3]. Furthermore, attack sophistication can range from ones mounted with single relatively simple devices to multiple sophisticated spoofers [4].

A multiplicity of methods has been proposed to defend GNSS receivers, including receiver autonomous integrity mon-

itoring (RAIM), signal processing, and statistical testing [5]–[8]. Such schemes can be effective in detecting attacks utilizing Doppler shift, signal-to-noise ratio (SNR), signals of opportunity (SOP), and inertial measurement unit (IMU), as these opportunistic data and signal properties inherently assist the detection process. These detectors are typically implemented at individual GNSS receivers or their encompassing platforms, with little consideration of privacy preservation or the utilization of distributed data across multiple devices. Operating in diverse volatile settings, with complex radio propagation environments, while facing adversaries, can be challenging, with varying attack detection performance. This calls for data-driven methods to improve accuracy and reliability.

A promising approach to enhance detection is to collect data from mobile platforms with GNSS and leverage machine learning to train detection models. Deep learning-based methods [9], [10] can use signal properties and opportunistic data as input features, e.g., power, phase, and SNR. These methods need a large number of labeled datasets, which are costly and labor-intensive in real-world environments. Moreover, they require significant processing power on the server side. Beyond these challenges, the collection and upload of data to the server for training raises privacy concerns. Position data is highly sensitive, which limits the sharing of data among GNSS receivers and restricts the development of collective, robust detection methods; thus, federated learning [11], [12] is proposed to enhance privacy and detect GNSS attacks.

While recent federated learning approaches [11]–[14] can be used for attack detection, none of the existing works have explored self-supervised federated learning for GNSS attack detection. Since labeling datasets is labor-intensive, self-supervision is important for enabling real-time and adaptive online training, offering greater practicality for real-world applications. Therefore, the objective of this paper is to investigate a self-supervised federated learning framework based on our previous GNSS attack detection scheme [8], which locally generates spoofing likelihood as labels from its detector by using SOP and IMU. A key challenge is how to use these labels with GNSS, opportunistic data, and signals’ properties to train local models across mobile platforms. Another challenge is how to federate and aggregate the local models from different mobile platforms into a stronger global model for detection. In

This work was supported in part by the SSF SURPRISE cybersecurity project, the Security Link strategic research center, the Karl Engver’s Foundation, and the China Scholarship Council. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725. We would also like to acknowledge the work of the organizers of Jammertest 2024.

addition, due to privacy concerns, sharing datasets containing positional, motion, and network-related information with other mobile platforms or server is prohibited by design.

Our proposed framework contains two main parts. The server part is a cloud server responsible for tasks such as housing the global detector model and model aggregation. The edge clients part involves numerous local mobile platforms, each equipped with a self-supervised anomaly detector model. These models utilize long short-term memory (LSTM) networks, which are also well-suited for capturing temporal dependencies in sequential data. Each platform processes a local dataset, which includes time-series information collected from GNSS receiver (GNSS position, automatic gain control (AGC), carrier to noise density ratio (C/N0), Doppler shift, etc.), network infrastructures (network position), and onboard sensors (acceleration, attitude, etc.). The self-supervised model is trained using the spoofing-deviation labels generated from the method in [8]. The local mobile platforms do not upload their local datasets to the cloud server. Instead, they transmit only the parameters of their respective local models. Upon receiving these parameters, the cloud server performs model aggregation by combining them into a single global model using the FedAvg algorithm. The aggregated global model is distributed back to the mobile platforms and then trained again.

In the experimental evaluation, we test GNSS spoofing data that we collected from Jammertest 2024 [15] using multiple vehicle-mounted smartphones. The dataset includes both independent and identically distributed (i.i.d.) and non-i.i.d. GNSS positions and opportunistic data across different mobile devices—the former refers to training and testing data originating from the same trace and device, while the latter involves data from different traces or devices. We found that the proposed self-supervised federated learning and its corresponding centralized learning have performance gains over the baseline. Meanwhile, the proposed federated method preserves position privacy for mobile platforms.

The novelty and contributions of this paper are:

- Utilization of self-supervised federated learning with opportunistic data for GNSS spoofing detection.
- Elimination of the requirement for labeled or annotated datasets in deep learning-based GNSS spoofing detection.
- Performance gain over position-based and deep learning-based methods on real GNSS spoofing detection, in terms of accuracy.

It is also noteworthy that the above are achieved without user privacy deterioration compared with the deep learning method with centralized training.

The remainder of the paper is organized as follows: Sec. II provides background and reviews related work for GNSS attacks and learning-based countermeasures. Sec. III presents our system model and adversary. Sec. IV details the problem and the proposed scheme. Sec. V discusses evaluation and comparison with baseline methods. Finally, Sec. VI concludes the paper.

II. RELATED WORK AND BACKGROUND

This section provides an overview of GNSS attacks and reviews related work of attack detection using machine learning techniques, including deep learning. Subsequently, we summarize recent advancements in federated learning with a focus on security and privacy.

A. GNSS Jamming and Spoofing Attacks

GNSS jamming disrupts receivers by transmitting high-power radio frequency signals within or near GNSS frequency bands, effectively overpowering the legitimate satellite signals. GNSS spoofing maliciously manipulates user position and time, especially because civilian GNSS usually lacks authentication, and the protocol, encoding, and modulation are publicly available. Even if GNSS signals were authenticated, the attacker can launch a relaying or replaying attack on GNSS [16]. Prior to spoofing, GNSS jamming is often used to force the GNSS receiver out of the satellite signal lock [17]. The simplest way of generating a spoofing signal is meaconing, which is the retransmission of legitimate GNSS signals with a time delay [16]. However, if the receiver has an accurate timer and already knows its recent location, the delayed time introduced by the meaconer will result in a sudden time shift, which may be detected. A variation of meaconing, selective delay, can rebroadcast individual satellite signals [18]. This can modify the position solution only without changing the time. Other sophisticated spoofing attacks can overcome more technical limitations, such as portable spoofers, which are attached to the victim [19]. Additionally, [20] focuses on the strategy of global positioning system (GPS) spoofing, which combines the road contextual information of the city map and can generate a designed route for spoofing a moving receiver.

B. Artificial Intelligence for GNSS Security

Machine learning, particularly deep learning, against GNSS attacks has shown promising potential and gained momentum, utilizing a range of models such as random forest [21], support vector machine (SVM) [9], [21], multilayer perceptron (MLP) [10], [22], [23], convolutional neural network (CNN) [22], [24], [25], Gaussian mixture model (GMM) [26], LSTM [24], [27], and recurrent neural network (RNN) [27].

For detecting GNSS jamming, [21] focuses on analyzing three common GNSS interference signals by extracting various entropy features (e.g., power spectral entropy), creating a combined entropy dataset, and utilizing SVM and radio frequency methods to classify the signals. Alternatively, [9] treats the classification of jammers in GNSS bands as a black-and-white image classification problem. Time-frequency analysis and image mapping of jammed signals are used to categorize the received signal into six classes. This method achieves notable classification accuracies of up to 94.90% with SVM and up to 91.36% with CNN.

To detect GNSS spoofing, [22] explores a cross-ambiguity function to train data-driven models for probabilistic classification, which focuses on each satellite individually and makes use of complex neural networks, including an MLP and two

types of CNNs. In [26], a GMM-based unsupervised method detects and mitigates GNSS signal spoofing by clustering the positions generated by the benign GNSS signals and isolating spoofed pseudoranges. Furthermore, [24] utilizes both CNN and LSTM to identify a spoofer by classifying the pairwise cross-correlation of different receivers and comparing the cyclic profiles, then observes that CNN achieves the highest accuracy. LSTM for anomaly detection leverages the predictability of the Doppler traits of the received GNSS signals: the data was collected using cost-effective software-defined radio (SDR) receivers and processed on affordable embedded platforms (e.g., Jetson Nano) to predict Doppler shift for spoofing detection [27].

When GNSS signals are mixed up with jamming, spoofing, and other interference signals, a robust deep learning technique combined pre-trained CNN with transfer learning to detect and classify disruptions of GNSS signals based on time-frequency analysis in [25]. An artificial neural network (ANN) trained by particle swarm optimization is proposed to detect various types of interactions affecting GNSS [23]. By using received signal power and distortion in the correlation function as feature vectors, this ANN classifies received signals into categories such as jammed, spoofed, multi-path afflicted, or interference-free. [28] introduces the GNSS-Finland monitoring platform, which employs the FinnRef reference network and deep learning methods to analyze big data from GNSS-Finland to identify trends in signal quality, detect anomalies, assess continuity, and forecast crucial failures in positioning and timing.

However, the limitations of these machine learning methods come when high-quality training data with annotation is unavailable. Additionally, when processing non-i.i.d. datasets, the offline deep learning models face challenges in achieving generalization.

C. Federated Learning for Security and Privacy

Federated learning [29] has emerged as a promising approach for training machine learning models while addressing privacy concerns. Instead of transferring the raw dataset to a central server for training, participating devices collaboratively train the model locally and share only the model updates (gradients or weights) with the central server.

Considering the application of federated learning in security, since the increasing deployment of Internet-of-Things (IoT) devices in daily life has resulted in many vulnerable devices, yet existing intrusion detection techniques are ineffective due to the massive scale of the problem and diverse types of devices and manufacturers involved. Therefore, [13] introduces an autonomous self-learning distributed system that utilizes device-type-specific communication profiles to detect anomalous deviations in communication without human intervention or labeled data, leveraging a federated learning approach for efficient profile aggregation, making it the first system to employ this approach for intrusion detection based on anomaly detection. FedCRI in [14] is a solution for sharing cyber-risk intelligence, wherein mobile cyber-risks were transformed into effective risk detection models based on contributions

from different mobile service providers, and their extensive evaluation on real-world user databases representing 23.8 million users of security-critical mobile apps, enabling effective identification of risks on mobile devices.

Federated learning has also proven its success in privacy-preserving applications beyond security. To improve next-word prediction in smartphone virtual keyboards, Google deploys an RNN language model trained using federated learning [30], a distributed on-device learning framework and the effectiveness of server-based training with stochastic gradient descent is compared to client device training, which showcases the advantage of training language models on client devices without compromising user data privacy and gives users more control over their data usage. Similarly, for object detection in autonomous driving systems, [31] proposes a federated learning-based approach that preserves data privacy while maintaining performance by training the model in a decentralized manner and analyzes the impact of this decentralized approach on object detection performance in a real-world traffic environment. In addition, the challenge of deep learning-based medicine lies in finding sufficiently large and diverse datasets, which are rare in individual institutions, leading to privacy and ownership challenges. Then, in [32], through a paradigm for data-private multi-institutional collaborations, models trained among 10 institutions achieve 99% of the quality achieved with centralized data.

Despite its advantages, federated learning is vulnerable to adversarial attacks. Current defenses either rely on techniques such as differential privacy or analyze model weights using outlier detection methods limited to specific data distributions, so [33] proposes CrowdGuard, a model filtering defense that leverages client data and secure enclaves to analyze individual models without data leaks. It introduces a novel metric to analyze network hidden layer outputs, coupled with a significance-based detection algorithm, enabling effective detection of poisoned models even in non-i.i.d. scenarios. Metric-Cascades in [34] use multiple detection metrics, such as Euclidean magnitude and direction, to filter poisoned model updates. The evaluation demonstrated that it successfully distinguishes backdoors from distortions, making it the first defense resilient to strong adaptive adversaries in real-world scenarios with minimal overhead. On the other hand, a malicious server can also use uploaded models to derive sensitive information. Hence, a decentralized framework is proposed in [35] that utilizes multi-party computation primitives like secret sharing, providing strict privacy guarantees against curious aggregators or colluding data owners.

III. SYSTEM MODEL AND ADVERSARY

As shown in Fig. 1, we consider multiple mobile GNSS platforms (e.g., smartphone, car, and drone) equipped with common modules (e.g., Wi-Fi, Bluetooth, cellular, IMU, and speed sensors), and streaming signal-level properties from satellites (AGC, antenna C/N0, baseband C/N0, and Doppler shift). The platforms are connected to a cloud server, which is curious but honest, so sharing datasets that contain location,

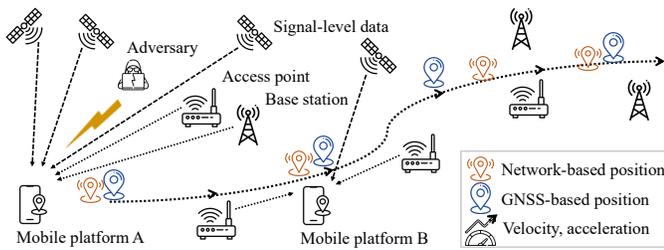


Fig. 1. System and adversary model illustration.

motion, and network information with the cloud server or other mobile platforms is not acceptable.

Provided that a GNSS position attack is available, when a mobile platform moves and navigates within the attack area, the GNSS-provided location will deviate from the actual location. As a result, network-provided positions, motion information, and signal-level properties will be inconsistent with GNSS position information and its benign behavior. However, the training data (encompassing the GNSS receiver, network infrastructures, and onboard sensors) is not manually labeled or annotated to indicate “benign” or “under attack”. Mobile platforms may differ, i.e., they may be navigating in different areas and using different hardware. This results in a so-called non-i.i.d. data situation. Similarly, if the hardware were the same and the navigation traces were similar, the data could be considered i.i.d..

The external adversary uses spoofed or replayed/relayed GPS and/or Galileo signals to force GNSS receivers to incorrectly compute their positions (and/or time) [16], [36]–[38]. We assume that the adversary knows the victim location and can use state-of-the-art attack techniques. We assume the attacker only operates within the GNSS domain and does not attack other network infrastructures, including cellular and Wi-Fi communications. Additionally, we assume the attacker does not have physical control over the server and mobile platforms, and thus cannot manipulate the process of deriving detection models and model parameters.

Notations. We denote the position of the m th mobile platform at time t is located at the coordinates $\mathbf{p}_{\text{true}}(m, t) \in \mathbb{R}^2$, where $m = 1, 2, \dots, M$; M is the total number of mobile platform. The said platform has a GNSS provided position $\mathbf{p}_{\text{gnss}}(m, t)$, a network-provided position $\mathbf{p}_{\text{net}}(m, t)$, as well as speed $\mathbf{v}(m, t)$, acceleration $\mathbf{a}(m, t)$, attitude $\omega(m, t)$ from onboard sensors, and GNSS signal properties $\mathbf{s}(m, t)$, which encompasses mean, median, minimum, and maximum values of AGC, antenna C/N0, baseband C/N0, and Doppler shift.

IV. PROPOSED SCHEME

The proposed scheme uses a federated self-supervised learning-based detection framework, as depicted in Fig. 2. This framework allows for effective collaboration and online knowledge sharing among mobile platforms while safeguarding the privacy of user data.

The mobile platforms collect $\mathbf{p}_{\text{gnss}}(m, t)$, $\mathbf{p}_{\text{net}}(m, t)$, $\mathbf{v}(m, t)$, $\mathbf{a}(m, t)$, $\omega(m, t)$, and $\mathbf{s}(m, t)$ to construct local

datasets (Sec. IV-A). Each platform trains an LSTM regression model in Sec. IV-C using its local dataset as input with the generated labels from Sec. IV-B, and then uploads the model parameters to the cloud server. The cloud server adopts the FedAvg algorithm to aggregate model parameters trained based on the distributed datasets (Sec. IV-D) to improve both the accuracy of spoofing detection and data privacy.

A. Feature Engineering

On each mobile platform, we construct two kinds of features from the local dataset: position-based and signal-based features.

The position-based features stem from our previous scheme [8], which provides a secure fused position $\mu \in \mathbb{R}^2$ with uncertainty $\sigma \in \mathbb{R}^2$ using $\mathbf{p}_{\text{gnss}}(m, t)$, $\mathbf{p}_{\text{net}}(m, t)$, $\mathbf{v}(m, t)$, $\mathbf{a}(m, t)$, and $\omega(m, t)$. Four elements are encompassed: the estimated position residual and uncertainty of latitude and longitude, calculated by $\{\mu(m, t) - \mathbf{p}_{\text{gnss}}(m, t), \sigma(m, t)\} \in \mathbb{R}^4$. The signal-based features are four statistics of satellite signals. We calculated mean, median, minimum, and maximum values of the following physical properties for GPS L1 and Galileo E1 (can be extended to other constellations):

- AGC: Regulates signal amplitude by automatically adjusting the receiver’s gain to compensate for variations in signal power, which can also indicate interference.
- Antenna C/N0: Represents the ratio of the signal power to the noise power density at the antenna, influenced by atmospheric conditions, satellite elevation, and interference.
- Baseband C/N0: Measures the signal quality after down-conversion and filtering, similar to antenna C/N0.
- Doppler shift: Captures frequency changes caused by satellite-receiver relative motion, essential for satellite fingerprint construction and velocity estimation.

In a benign environment, AGC, C/N0, and Doppler shift of different satellites should be different for each satellite. However, spoofing often causes artificially similar values and typically higher signal power. These signal-based features have $4 \times 4 \times 2$ elements, corresponding to four types of statistics, four properties, and two constellations. Hence, the extracted feature set includes 32 signal-based elements and four elements representing the estimated position residual and uncertainty in latitude and longitude, in a total of 36 elements.

Moreover, features are normalized and cleaned within the local dataset. For the position-based features, the estimated position residual and uncertainty may contain extreme values. To mitigate extremes, values exceeding the 95th percentile are capped at the 95th percentile threshold. In the signal-based feature, some signal properties contain invalid values (e.g., missing or faulty measurements). These invalid values are replaced with the minimum value within their respective valid range. Finally, a min-max scaling strategy is applied separately to each feature type. Each feature is rescaled to the range $[0, 1]$ to ensure stable training.

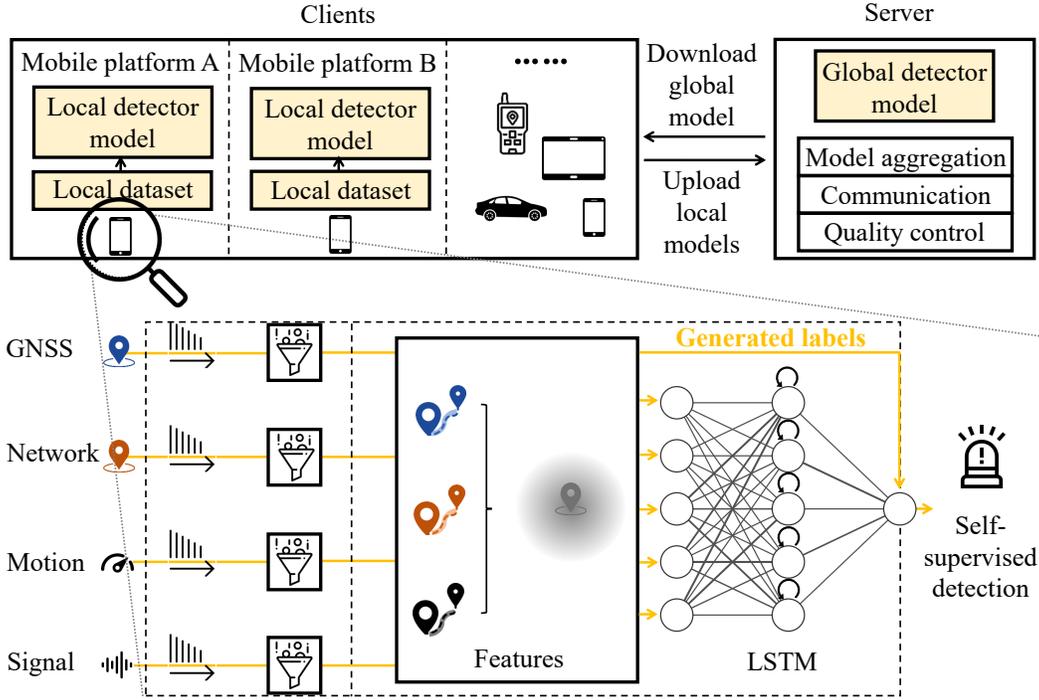


Fig. 2. Overview of self-supervised federated GNSS spoofing detection.

B. Label Generation

Self-supervised labels are generated based on the method proposed in [8]. The generation uses $\mathbf{p}_{\text{gnss}}(m, t)$, $\mathbf{p}_{\text{net}}(m, t)$, $\mathbf{v}(m, t)$, $\mathbf{a}(m, t)$, and $\boldsymbol{\omega}(m, t)$ to compute a scalar value of the estimated GNSS position deviation, normalized to the range $[0, 1]$, for each (m, t) in a fully automated manner.

We first compute the Euclidean norm of the estimated position residual, $\mu(m, t) - \mathbf{p}_{\text{gnss}}(m, t)$ (the difference between the secure fused position in [8] and GNSS position), to obtain a scalar representation of the estimated GNSS deviation. To remove extreme values, deviations exceeding the 95th percentile are capped at this threshold. Finally, we apply min-max scaling to normalize the values.

These normalized values serve as self-supervised labels, representing the estimated GNSS position deviation at each (m, t) . As they are generated entirely locally without any external manual annotation, this approach enables self-supervised learning. Note that the algorithm used for the generation of these labels is not a contribution of this work; instead, we use these labels to build our self-supervised federated learning for GNSS spoofing detection.

C. Model Structure

We implement the detection model based on an LSTM neural network on each mobile platform, as LSTM is well-suited for processing time series data. The network input is the features from Sec. IV-A, and the output is a scale value in $[0, 1]$. The objective is to capture the temporal dependencies of each signal feature and identify anomalies.

Layer structure. The model contains two LSTM layers and a fully connected output layer. The first LSTM layer contains 100 units, which takes time series feature input and outputs a sequence of hidden states at each moment. The second LSTM layer also contains 100 units but processes the sequence of hidden states from the first layer and outputs the hidden state at the last time step only.

Activation function. In the last fully connected layer, we use the Sigmoid activation function to output a probability value to indicate whether the current position is a result of GNSS spoofing.

Loss function. The loss function of the model uses the mean squared error (MSE) to reduce the prediction error of deviation.

Batch size and learning rate. The batch size of the model is set to 72; the learning rate is adjusted proportionally to adapt to the different data volumes of each device.

Early stopping strategy. The EarlyStopping strategy is used in training. When the verification loss does not improve after 20 epochs, the training is stopped to avoid overfitting.

D. Model Aggregation

Mobile platforms do not upload local datasets but upload trained local model parameters. The cloud server aggregates these local model parameters and uses the FedAvg algorithm to average them to form a global model. The server sends global model parameters to all devices after each iteration to achieve continuous learning. It performs multiple iterations of aggregation on the local dataset of different devices to improve the generalization ability of the model.

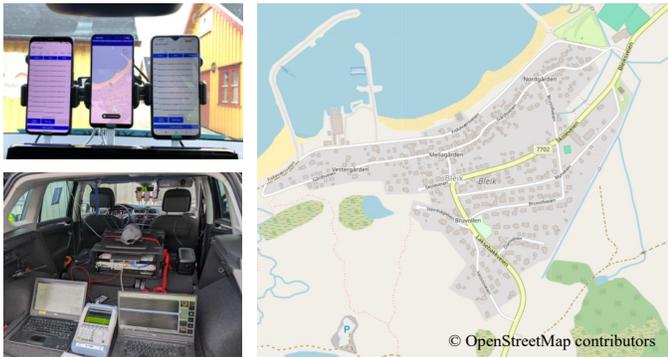


Fig. 3. Jammertest main test area (right) and mounted smartphones in a vehicle (left).

E. Quality Control

After the global model undergoes sufficient rounds of federated training, its detection accuracy should stabilize. At this point, the system initiates a quality control process to assess parameter updates from each local model. In each learning iteration, upon receiving a local model update, the cloud server distributes this update to other participating mobile platforms for independent evaluation on their respective local datasets. Then, the mobile platforms upload the predicted labels, and the server computes a performance metric—specifically, the area under the curve (AUC) score—based on these uploaded labels and the predictions from the previous global model. The FedAvg algorithm will only accept this parameter update (from a local model) if the metric is better than a criterion threshold. Given that we only consider all participants who contribute model parameters that are benign, the quality control here aims to filter out low-quality data samples.

V. EXPERIMENTAL EVALUATION

As shown in Fig. 3, our GNSS spoofing dataset is sourced from Jammertest 2024 [15], and includes data collection from six Android smartphones. Phone 1 and Phone 4 are Google Pixel 8. Phone 2 and Phone 6 are Google Pixel 4 XL. Phone 3 is Xiaomi Redmi 9, and Phone 5 is Samsung Galaxy S9. All smartphones support multiple constellations, while only Google Pixel 4 XL and Pixel 8 support double frequencies. The dataset consists of 85 drive-testing traces in total that were recorded throughout the day, from morning to evening, around Bleik town. The smartphones record timestamps, GNSS positions, network positions, IMU data, and GNSS signal properties (AGC, antenna C/N0, baseband C/N0, and Doppler shift) via Android APIs. Ground truth positions are obtained from two u-blox ZED-F9P receivers using benign constellations with the help of a nearby reference station. Labels for self-supervised learning are generated locally through the opportunistic data fusion algorithm [8], as Sec. IV-B.

The deep learning framework is Keras with a TensorFlow backend, and we choose LSTM module to process time series data, so it can capture temporal dependencies in features. Regarding the federated learning part, we implement the

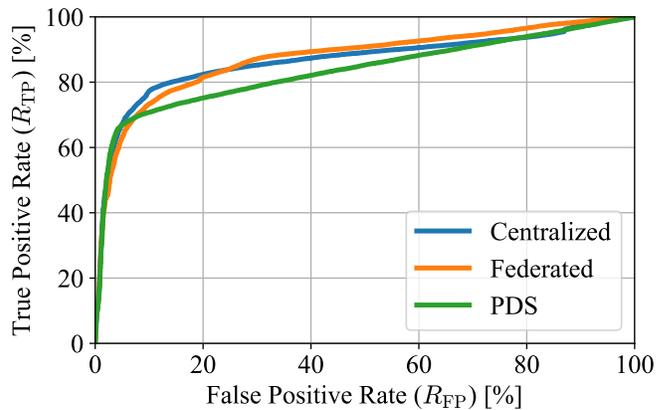


Fig. 4. ROC curves of the centralized and federated self-supervised detection, and the position-based detection.

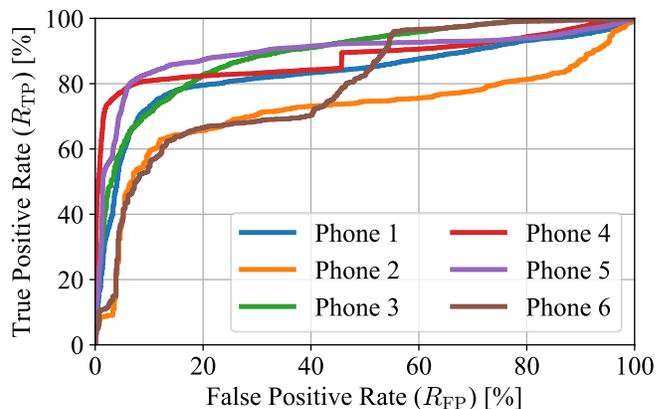


Fig. 5. ROC curves of the proposed self-supervised detection for each smartphone based on their local i.i.d. data.

FedAvg algorithm on the cloud server to average all local model parameters. Additionally, we do not simulate processing overhead, transmission delays, or packet loss in wireless communications between the server and clients.

We have different train/test data-splitting methods. One is to split the data based on different smartphones without considering different traces. Another is to randomly choose 10% of traces for testing, while the rest of the traces are for training. In addition, as our detection is self-supervised, we use the entire dataset for training and testing for the comparison of centralized or federated detection.

A. Evaluation Metrics

To evaluate the performance of the proposed scheme, we use true positive rate (R_{TP}) versus false positive rate (R_{FP}) and plot receiver-operating characteristic (ROC) curves. Additionally, we calculate AUC values, which are the area under the ROC curves. To systematically evaluate different scenarios, our experiments analyze the metrics across same-device training and testing, same-model training and testing, and cross-model generalization using same or different traces.

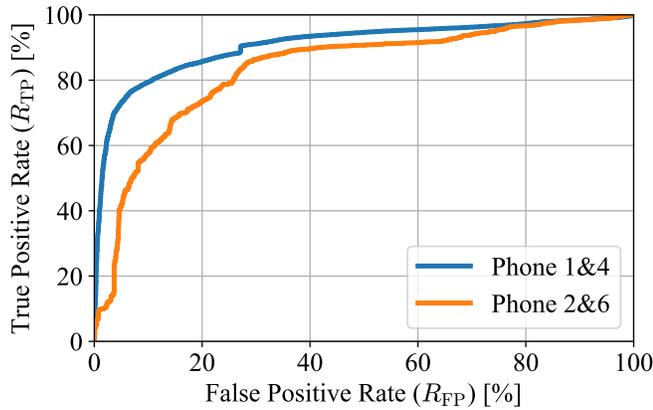


Fig. 6. ROC curves of the proposed self-supervised federated detection for each smartphone model. Training and testing data are collected from the same smartphone model and driving traces.

B. Evaluation Results

1) Comparison Between Centralized and Federated:

We compare the detection accuracy of the proposed self-supervised detection in a centralized or federated manner with the position-based detection scheme (PDS) [8]. ROC curves for the methods of position fusion, centralized training, and the proposed federated learning are shown in Fig. 4. We find ROC curve and training loss value of the proposed become stable after 600–1000 epochs. The centralized training curve has an AUC value of 86.6%. The federated learning curve’s AUC is 87.4%, while the curve of PDS is 83.5%.

This comparison shows that the self-supervised learning-based approach outperforms the baseline method, and validates that federated learning can achieve competitive performance compared to centralized training. Furthermore, the proposed scheme has at most 10% true positive rate gain over PDS [8] in Fig. 4, and preserves location data privacy.

In our scheme, centralized training achieves a higher accuracy than federated one, when $R_{FP} < 10\%$. This phenomenon is common in federated learning due to data samples being sequential and shuffled across batches in centralized training. In contrast, federated learning involves parallel training across clients, where each client trains on relatively unrefined models and locally available data. As a result, the model struggles to generalize effectively by capturing shared patterns across the entire dataset.

2) *Generalization Between Different Devices:* This evaluation aims to assess the generalization performance of the proposed method across different devices, both within individual devices and among devices of the same smartphone model.

First, we compare the detection accuracy of the proposed self-supervised detection within every individual phone, termed one device training and same device testing. The result is shown in Fig. 5. The phones will not upload data; instead, they conduct training and testing of detectors separately and locally. Phone 1 and Phone 4 are the exact smartphone model, but the data of collected driving traces are different (non-i.i.d.).

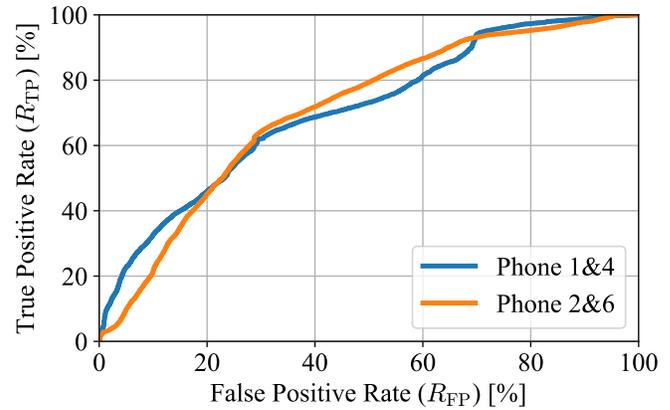


Fig. 7. ROC curves of the proposed self-supervised federated detection for different smartphone models. Train/test data-splitting is based on smartphone model: Training data is collected from one smartphone model and testing data from another model, both obtained along the same driving traces.

Phone 2 and Phone 6 are also the same smartphone model, while Phone 3 and Phone 5 refer to different smartphone models. We can observe that the true positive rate of our algorithm for the Google Pixel 4 XL phones is not as good as other phones. These varying levels of performance across different phone models are potentially due to hardware differences or non-i.i.d. data distributions.

Then, we conduct self-supervised federated detection within devices of the same smartphone model name, termed one model training and same model testing. The result is shown in Fig. 6. Compared with the results of individual device training and testing in Fig. 5, federated detection within a smartphone model has higher accuracy, highlighting the benefits of collaborative training.

Similarly, we attempt to train the detection using the devices of one smartphone model and test the detection using other smartphone models, termed one model training and different model testing. The result is shown in Fig. 7. The curve of Phone 1&4 uses Google Pixel 8 phones for training and all other phones for testing; the curve of Phone 2&6 uses Google Pixel 4 XL phones for training and all other phones for testing. We observe that the detector cannot perform well on a given hardware model if it has not been trained on data from the same model.

3) *Generalization Between Different Traces:* Different from the previous comparisons that use the same trace for both training and testing, this experiment divides the traces into training traces and testing traces.

In Fig. 8, we compare the performance of the proposed self-supervised detection in a centralized or federated manner. The centralized training curve has an AUC value of 86.5%. The federated learning curve is 86.7%, while the PDS curve is 83.4%. These AUC values and their corresponding curves closely match those depicted in Fig. 4, i.e., 0.7% difference at most. This indicates that the proposed method is good at generalizing between different traces.

Likewise, we compare the detection accuracy of the pro-

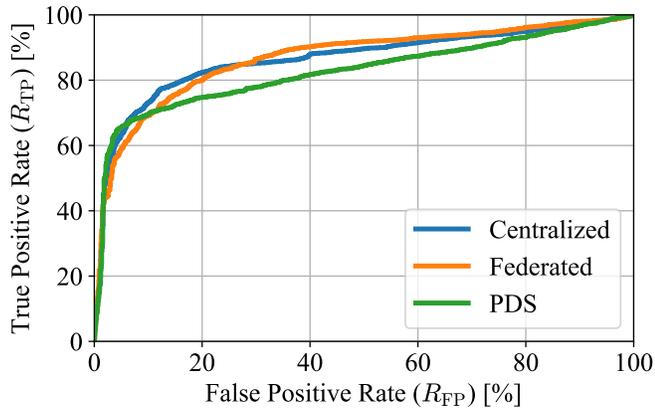


Fig. 8. ROC curves of the centralized and federated self-supervised detection, and the position-based detection. Train/test data-splitting is based on driving trace: 10% of traces are randomly chosen for testing, while the rest of the traces are for training.

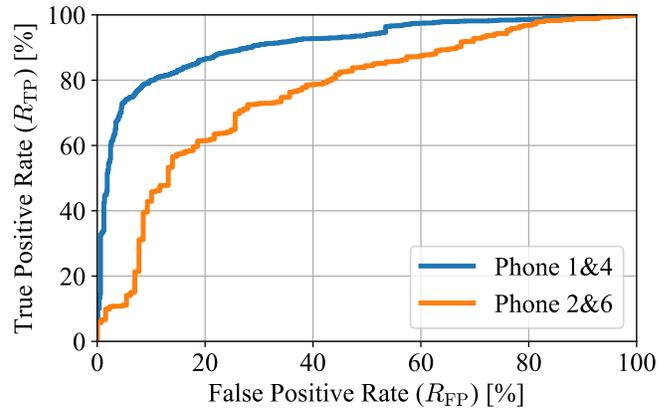


Fig. 10. ROC curves of the proposed self-supervised detection for each smartphone model. Training and testing data are collected from the same smartphone model but with different driving traces, i.e., train/test data-splitting is based on driving trace.

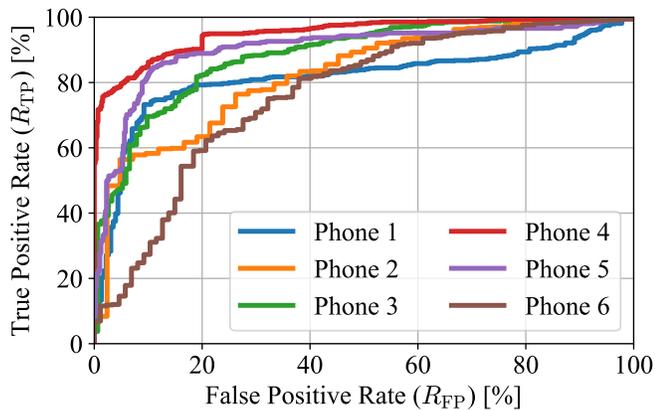


Fig. 9. ROC curves of the proposed self-supervised detection for each smartphone. Train/test data-splitting is based on driving trace. Specifically, for each curve, training and testing data are from the same smartphone but with different traces.

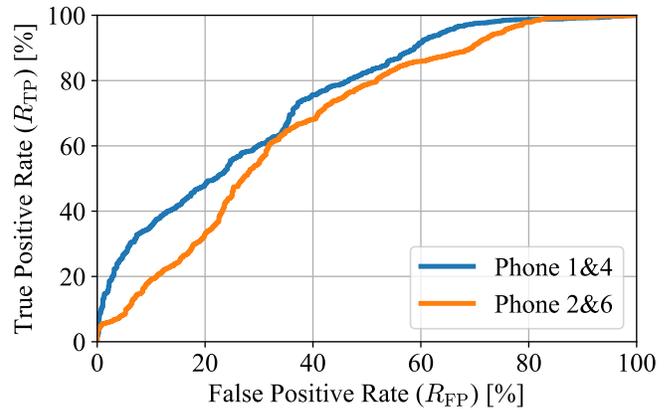


Fig. 11. ROC curves of the proposed self-supervised detection for different smartphone models. Training and testing data are collected from different smartphone models and driving traces. Specifically, the curve for Phone 1&4 is trained on Google Pixel 8 and tested on other phones. Within each model, the train/test split is further based on driving traces.

posed self-supervised detection within every individual phone, but with different traces for training and testing. The results in Fig. 9 show similar true positive rate curves and relative performance gains to those in Fig. 5.

Next, we conduct self-supervised federated detection within devices of the same smartphone model name but using different traces for training and testing. Compared to Fig. 6, which is based on i.i.d. data where the same traces are used for both training and testing, Fig. 10 shows a similar performance. However, we observe that their true positive rate is a bit lower than Fig. 6. Additionally, detection performance varies across different smartphone models: Phone 1&4 achieves much higher accuracy than Phone 2&6. This may be attributed to the larger training set available for Phone 1&4 (9899+10626 samples) compared to Phone 2&6 (6238+6734 samples). A larger dataset generally improves accuracy, and the transition from an i.i.d. setting (Fig. 6) to a non-i.i.d. setting (Fig. 10) also introduces additional generalization challenges.

Furthermore, similar to Fig. 7, we extend our evaluation by training the detector on devices of one smartphone model using traces of the training set, then testing it on a different smartphone model with different traces. The results are presented in Fig. 11. Fig. 7 and Fig. 11 show a similar performance, as both involve non-i.i.d. training and testing settings. The key difference is that Fig. 11 needs a more comprehensive generalization, since it introduces the train-test split for both the smartphone model and the driving traces.

C. Discussion

Our detection method is self-supervised and federatively trained by multiple devices that provide heterogeneous traces and opportunistic data. Unlike existing deep learning-based detections, the proposed method does not require manual annotation for training data, meaning that its training process is self-supervised by the labels generated from a traditional

detection, PDS [8]. Interestingly, while PDS provides training labels for the proposed federated detection, the proposed method outperforms PDS in accuracy. Additionally, without any loss of privacy, each device transmits only model parameters instead of position-related data.

The proposed federated detection can be generalized to different situations, and we show the generalization of the detection model through a comprehensive evaluation. It evaluates on different devices, smartphone models, and driving traces. The results demonstrate that the method performs robustly on i.i.d. and non-i.i.d. data.

D. Limitations and Roadmap

We observe that the detection performance on Pixel 4 XL (Phone 2 and Phone 6) is lower than that on other smartphones. Based on feature engineering, this may be because Pixel 4 XL has a significantly different distribution of signal power from satellites than other smartphones. Furthermore, the dataset size, including network position data, for the Pixel 4 XL is much smaller than that of the Pixel 8.

Our next step is to implement this self-supervised federated detection in a real-world wireless communication environment. Although the current evaluation is based on a real-world dataset, the federated learning algorithm has not yet been deployed on actual mobile platforms with live communication. Communication will introduce network delay and loss, which may affect our detection. Furthermore, we currently consider all mobile platforms to be benign and honestly contribute model parameters. If attackers participate in model aggregation, we need to introduce more practical defense mechanisms, supported by theoretical analysis, to defend against attacks.

VI. CONCLUSION

In this paper, we present self-supervised federated GNSS spoofing detection leveraging opportunistic data that enables mobile platforms to share knowledge about GNSS spoofing without leaking position privacy. Furthermore, it achieves better privacy preservation and detection performance than existing position-based and deep learning-based methods due to the expected benefits of self-supervised and federated GNSS spoofing detection. Labels are generated by PDS [8], which are used to train local LSTM models immediately. The LSTM input features include estimated position deviations and features derived from GNSS signals. Local model parameters are uploaded to a server, not the actual measurements, and then the server performs model aggregation and quality control. Our evaluation using a real-world dataset from Jammertest 2024 shows performance improvements and exhibits a good generalization across different devices, smartphone models, and driving traces.

REFERENCES

- [1] D. Goodin, "GPS interference caused FAA reroute Texas air traffic. experts stumped," *Ars Technica*, 2022. [Online]. Available: <https://arstechnica.com/information-technology/2022/10/cause-is-unknown...>
- [2] J. Shen, J. Y. Won, Z. Chen, and Q. A. Chen, "Drift devil: Security multi-sensor fusion based localization high-level autonomous driving GPS spoofing," in *Proc. 29th USENIX Security*, virtual event, Aug. 2020.
- [3] Q. Bin, C. Ziwen, X. Yong, H. Liang, and S. Sheng, "GPS spoofing-based time synchronisation attack advanced metering infrastructure its protection," *J. Eng.*, vol. 2020, no. 9, pp. 809–815, 2020.
- [4] S. Liu, X. Cheng, H. Yang, Y. Shu, X. Weng, P. Guo, K. C. Zeng, G. Wang, and Y. Yang, "Stars can tell: Robust method defend GPS spoofing attacks using off-the-shelf chipset," in *Proc. 30th USENIX Security*, virtual event, Aug. 2021.
- [5] S. Khanafseh, N. Roshan, S. Langel, F.-C. Chan, M. Joerger, and B. Pervan, "GPS spoofing detection using RAIM INS coupling," in *Proc. IEEE/ION PLANS*, Monterey, CA, USA, May 2014.
- [6] M. Maaref and Z. M. Kassas, "Autonomous integrity monitoring vehicular navigation cellular signals opportunity IMU," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5586–5601, 2021.
- [7] F. Rothmaier, Y.-H. Chen, S. Lo, and T. Walter, "A framework GNSS spoofing detection combinations metrics," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 6, pp. 3633–3647, 2021.
- [8] W. Liu and P. Papadimitratos, "Probabilistic detection GNSS spoofing using opportunistic information," in *Proc. IEEE/ION PLANS*, Monterey, CA, USA, Apr. 2023.
- [9] R. Morales Ferre, A. de la Fuente, and E. S. Lohan, "Jammer classification GNSS bands machine learning algorithms," *Sensors*, vol. 19, no. 22, p. 4841, 2019.
- [10] P. Borhani-Darian, H. Li, P. Wu, and P. Closas, "Detecting GNSS spoofing using deep learning," *EURASIP J. Adv. Signal Process.*, vol. 2024, no. 1, p. 14, 2024.
- [11] P. Wu, H. Calatrava, T. Imbiriba, and P. Closas, "Jammer classification federated learning," in *Proc. IEEE/ION PLANS*, Monterey, CA, USA, Apr. 2023.
- [12] M. Deng, R. Luo, and Z. Yao, "GNSS interference signal classification based federated learning," in *Proc. 100th IEEE VTC*, Washington, DC, USA, Oct. 2024.
- [13] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DfIoT: Federated self-learning anomaly detection system IoT," in *Proc. 39th IEEE ICDCS*, Dallas, TX, USA, Jul. 2019.
- [14] H. Fereidooni, A. Dmitrienko, P. Rieger, M. Miettinen, A.-R. Sadeghi, and F. Madlener, "FedCRI: Federated mobile cyber-risk intelligence," in *Proc. NDSS*, San Diego, CA, USA, Apr. 2022.
- [15] Jammertest, "The world's largest open jamming spoofing test," *Jammertest*, 2024. [Online]. Available: <https://jammertest.no/about-2/>
- [16] M. Lenhart, M. Spanghero, and P. Papadimitratos, "Relay/replay attacks GNSS signals," in *Proc. 14th ACM WiSec*, virtual event, Jun. 2021.
- [17] Z. M. Kassas, J. Khalife, A. A. Abdallah, and C. Lee, "I am not afraid GPS jammer: Resilient navigation signals opportunity GPS-denied environments," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 37, no. 7, pp. 4–19, 2022.
- [18] P. Papadimitratos and A. Jovanovic, "GNSS-based positioning: Attacks countermeasures," in *Proc. IEEE Mil. Commun. Conf.*, San Diego, CA, USA, Nov. 2008.
- [19] D. Schmidt, K. Radke, S. Camtepe, E. Foo, and M. Ren, "A survey analysis GNSS spoofing threat countermeasures," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 1–31, 2016.
- [20] S. Narain, A. Ranganathan, and G. Noubir, "Security GPS/INS based on-road location tracking systems," in *Proc. IEEE S&P*, San Francisco, CA, USA, May 2019.
- [21] J. Xu, S. Ying, and H. Li, "GPS interference signal recognition based machine learning," *Mob. Netw. Appl.*, vol. 25, no. 6, pp. 2336–2350, 2020.
- [22] P. Borhani-Darian, H. Li, P. Wu, and P. Closas, "Deep neural network approach detect GNSS spoofing attacks," in *Proc. 33rd ION GNSS+*, virtual event, Sep. 2020.
- [23] S. Tohidi and M. R. Mosavi, "Effective detection GNSS spoofing attack using multi-layer perceptron neural network classifier trained PSO," in *Proc. 25th CSICC*, Tehran, Iran, Jan. 2020.
- [24] D. R. Kartchner, R. Palmer, and S. K. Jayaweera, "Satellite navigation anti-spoofing using deep learning receiver network," in *Proc. IEEE CCAAW*, Cleveland, OH, USA, Jun. 2021.
- [25] A. Elango, S. Ujan, and L. Ruotsalainen, "Disruptive GNSS signal detection classification different power levels using advanced deep-learning approach," in *Proc. 12th ICL-GNSS*, Tampere, Finland, Jun. 2022.
- [26] Z. Feng, C. K. Seow, and Q. Cao, "GNSS anti-spoofing detection based gaussian mixture model machine learning," in *Proc. 25th IEEE ITSC*, Macau, China, Oct. 2022.

- [27] R. Calvo-Palomino, A. Bhattacharya, G. Bovet, and D. Giustiniano, "Short: LSTM-based GNSS spoofing detection using low-cost spectrum sensors," in *Proc. 21st IEEE WoWMoM*, Cork, Ireland, Aug. 2020.
- [28] S. Kaasalainen, M. Mäkelä, L. Ruotsalainen, T. Malmivirta, T. Fordell, K. Hanhijärvi, A. Wallin, T. Lindvall, and S. Nikolskiy, "Reason-resilience security geospatial data critical infrastructures," in *Proc. 11th ICL-GNSS*, Tampere, Finland, Jun. 2021.
- [29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning deep networks decentralized data," in *Proc. 20th AISTATS*, Ft. Lauderdale, FL, USA, Apr. 2017.
- [30] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.
- [31] D. Jallepalli, N. C. Ravikumar, P. V. Badarinath, S. Uchil, and M. A. Suresh, "Federated learning object detection autonomous vehicles," in *Proc. 7th IEEE BigDataService*, Oxford, United Kingdom, Aug. 2021.
- [32] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen *et al.*, "Federated learning medicine: Facilitating multi-institutional collaborations without sharing patient data," *Sci. Rep.*, vol. 10, no. 1, p. 12598, 2020.
- [33] P. Rieger, T. Krauß, M. Miettinen, A. Dmitrienko, and A.-R. Sadeghi, "Close gate: Detecting backdoored models federated learning based client-side deep layer output analysis," *arXiv preprint arXiv:2210.07714*, 2022.
- [34] T. Krauß and A. Dmitrienko, "Avoid adversarial adaption federated learning multi-metric investigations," *arXiv preprint arXiv:2306.03600*, 2023.
- [35] Y. More, P. Ramachandran, P. Panda, A. Mondal, H. Virk, and D. Gupta, "SCOTCH: Efficient secure computation framework secure aggregation," *arXiv preprint arXiv:2201.07730*, 2022.
- [36] K. Zhang, E. G. Larsson, and P. Papadimitratos, "Protecting GNSS open service navigation message authentication distance-decreasing attacks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 2, pp. 1224–1240, 2022.
- [37] M. Lenhart, M. Spanghero, and P. Papadimitratos, "Distributed mobile message level relaying/replaying GNSS signals," in *Proc. ION ITM*, Long Beach, CA, USA, Jan. 2022.
- [38] M. Spanghero and P. Papadimitratos, "Time-based GNSS attack detection," *IEEE Trans. Aerosp. Electron. Syst.*, pp. 1–18, 2024.