# Exploring Adversarial Attacks on Learning-based Localization

Frost Mitchell*
Phillip Smith*
Kahlert School of Computing
University of Utah
Salt Lake City, Utah, USA

Aditya Bhaskara
Sneha Kumar Kasera
Kahlert School of Computing
University of Utah
Salt Lake City, Utah, USA

## ABSTRACT

We investigate the robustness of a convolutional neural network (CNN) RF transmitter localization model in the face of adversarial actors which may poison or spoof sensor data to disrupt or defeat the algorithm. We train the CNN to estimate transmitter locations based on sensor coordinates and received signal strength (RSS) measurements from a real-world dataset. We consider attacks from adversaries with varying capabilities to include naïve, random attacks and omniscient, worst-case attacks. We apply countermeasures based on statistical outlier approaches and train the CNN against adversarial attacks to improve performance. Adversarial training is shown to completely neutralize some attacks and improve accuracy by up to 65% in other cases. Our evaluation of countermeasures indicates that a combination of statistical techniques and adversarial training can provide more robust defense against adversarial attacks.

## CCS CONCEPTS

• **Networks** → **Location based services**; *Mobile and wireless security*; • **Theory of computation** → **Adversarial learning**; • **Computing methodologies** → *Neural networks.*

## KEYWORDS

transmitter localization, adversarial attacks, neural networks

**ACM Reference Format:**
Frost Mitchell, Phillip Smith, Aditya Bhaskara, and Sneha Kumar Kasera. 2023. Exploring Adversarial Attacks on Learning-based Localization. In *Proceedings of the 2023 ACM Workshop on Wireless Security and Machine Learning (WiseML '23), June 1, 2023, Guildford, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3586209.3591398

## 1 INTRODUCTION

As machine learning techniques continue to find new applications and improve existing techniques across a broad range of mobile computing applications, the risk of exploitation and compromise of these services continues to increase as adversaries adopt these same technologies. We consider the problem of transmitter localization in a wide-area environment using crowdsourced RF sensors, as proposed in [13, 17, 6]. The goal in such a localization setting is to locate malicious transmitters or spectrum offenders. Because the utility of such an application necessarily relies on inputs from crowdsourced user participants, we investigate the robustness of the model in the face of adversarial actors which may poison or spoof sensor data to mislead or otherwise defeat a localization algorithm.

We first train a convolutional neural network (CNN) to estimate transmitter locations based on sensor coordinates and received signal strength (RSS) measurements. The initial model is trained on a real-world dataset from [9] covering a 4 $km^2$ area. The baseline accuracy of this model on the test set is evaluated as a control. We then step through several attack scenarios to assess the robustness of the model and the impact of our proposed countermeasures. We consider attacks from adversaries with varying capabilities.

- `Naïve Attacks`: An adversary inserts measurement(s) consisting of a random location and random (bounded) RSS.
- `Omniscient Attacks`: An adversary, aware of a transmitter location and other device measurements, and with access to a surrogate model trained on identical data, attempts to misdirect or defeat localization.

In order to detect adversarial inputs and produce more robust localization techniques, we propose several countermeasures:

- `Outlier Exclusion`: Removes input sample(s) when the resulting location estimate is more than $\gamma$ meters from the estimate produced without that input.
- `Vector Outlier Exclusion`: Removes input sample(s) whose prediction matrix is more than $\gamma$ (Euclidean) distance from the average prediction matrix for all current input samples.
- `Adversarial Training`: Retrains a CNN model on adversarial samples to produce a model that is more robust to attacks.

These are further explained in sections 4 and 5, respectively.

**Goals and Contributions.** Briefly, the goal of this work is to evaluate naïve attacks on a learning-based localization technique in order to establish a baseline of attack effectiveness. We also evaluate high-complexity attacks from an omniscient adversary to provide an upper limit on the expected impact an attack could have. We also explore the impact of countermeasures, including adversarial training.

## 2 RELATED WORKS

In [5] Kaligineedi et al. seek to distinguish primary users from malicious users in a cognitive radio system by detecting statistical outliers, and Luo et a. [7] present a learning adversarial model and defense mechanism that can reduce influence in a cooperative spectrum sensing scenario. While many other works focus on spectrum occupancy and reporting, our application applies related techniques to mitigate impacts from malicious users in a mobile RF localization

*Both authors contributed equally to this research.



*WiseML '23, June 1, 2023, Guildford, United Kingdom*

ACM ISBN 979-8-4007-0133-7/23/06.
https://doi.org/10.1145/3586209.3591398

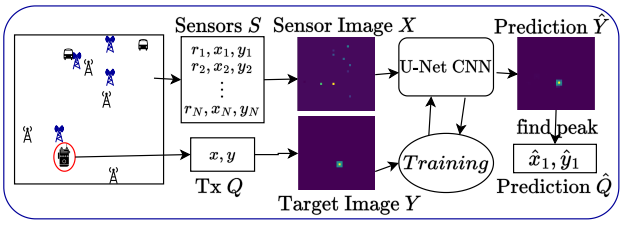



**Figure 1: The image-based localization process**

application that may be found in an open crowdsourced spectrum sensing application. A comprehensive survey of similar techniques across related domains can be found in [1].

Crowdsourced localization of transmitters is proposed in LLOCUS [12]. The authors propose a system with full mobility, where both transmitters and receivers are allowed mobility. Recent advances in learning-based localization include image-to-image solutions using CNNs, where an input image of sensor locations is transformed to a target image of transmitter locations, as used in [15, 10, 16]. To our knowledge, the robustness of these techniques to adversaries has not been investigated.

## 3 MODEL TRAINING

We train a CNN over a dataset consisting of mobile and fixed transmitters and receivers. We use an image-based localization technique from our previous work for RSS-based localization. Sensor measurements are converted into a 2D "map" with each pixel in the map corresponding to the Cartesian coordinates of the sensor, with the pixel intensity corresponding to the measured RSS value, and pixels without a sensor measurement are set to 0. Similarly, the transmitter location is encoded in a sparse 2D image where the pixel location corresponding to the transmitter is set to 1, with all other values at 0. This process is shown in Fig. 1. This measurement map is input to a CNN which learns to approximate the transmitter map. The CNN is based on the UNet architecture, has 19 convolutional layers with a downsampling and upsampling structure that produces an output of the same size as the input.

Formally, $S$, a vector of sensor coordinates and RSS measurements, is encoded into the sensor map $X$. The transmitter coordinates, $Q$, are encoded into the transmitter map $Y$. Let the CNN with model parameters $\theta$ be $h_\theta$. Then the CNN learns to approximate $Y$ as $h_\theta(X) = \hat{Y}$. The maximum value in $\hat{Y}$ is then taken to be the location estimate $\hat{Q}$. In terms of an attack, we consider perturbations to the sensor vector $S$, either by withholding sensors, injecting a perturbation into RSS values or coordinates, or by appending fake sensors as new entries in $S$.

We train our model and evaluate attacks on a real-world dataset taken from from [9]. This data consists of RSS measurements and transmitter and sensor locations covering a 4 km$^2$ area which was taken using POWDER [3], an open-access testbed on the University of Utah campus. The dataset includes the coordinates of a single handheld transmitter at 462.7 MHz moving through campus, along with RSS measurements and coordinates from software-defined radios (SDRs) on rooftops, at ground level, and on mobile shuttles. Since the dataset includes measurements from heterogeneous receivers, with different SRDs, antennas, and device placement, we normalize RSS measurements between 0 and 1 on a per-device basis.

## 4 THREAT SCENARIOS

Our threat scenario is characterized by attacks available to adversaries participating in a crowdsourced spectrum measurement system. Specifically, we envision a system that would be used for transmitter localization in which users submit RF measurements and associated sensor locations to a central server, sometimes described as a fusion center, which then estimates transmitter location. For purposes of our analysis, we only consider sensor measurements and the server location estimation function. For now, we do not test against collusion attacks in which multiple adversaries coordinate to degrade the service. We also limit the scope of attacks to not consider, for example, network, server or infrastructure attacks. We describe three classes of attacks in their respective sections below.

### 4.1 Naïve Attacks

These consist of attacks in which the adversary is not aware of transmitter locations, has no information about the localization algorithm and may only possess knowledge of the RF environment obtained from their own participating sensor node. For testing purposes, we assume an adversary randomly selects a location, assigns a random, bounded, RSS measurement and reports this to the server. We abbreviate this attack as A_IRLRR — Attack, Insert Random Location, Random RSS. We bound the RSS measurement between the 10th and 90th percentile of all RSS measurements that are reported and processed by the model for a particular transmitter location estimate. An adversary would not normally be aware of other measurements reported to the server, but this is done to resemble a rational adversary who choose from a reasonable range of values based on what may be observed from their own sensor — in general, an adversary could submit a large RSS value that would likely disrupt the location estimate, but this could also more easily be detected as an anomaly or attack by the server and filtered out. To represent the range of effects from this type of attack, we repeat this test five times across 600 unique test configurations of transmitter and sensor locations for a combined 3000 tests. These results are reported in Fig. 2.

If an adversary has information about a transmitter location, some understanding of the localization algorithm, and access to sensor measurements, and then exploits this knowledge to craft more effective attacks against the system, we call this an informed attack. This attack could be motivated by an adversary operating a transmitter, wishing to avoid detection and thus possessing the precise transmitter location. More generally, an adversary may simply report an RSS value that negatively correlates with what is detected at their own location. While informed attacks represent a more likely scenario, we do not evaluate these explicitly because the effects of any of these attacks would be bounded in effect between the naïve attacks and the worst-case, unrealistic attacks we describe in the next section.

## 4.2 Omniscient Attacks

In this work, the omniscient, otherwise known as *white-box*, attack setting is extremely powerful. Adversaries have full access to all sensor coordinates and RSS values, as well as the transmitter location. An adversary would not normally have access to the CNN used for localization, which we refer to as the *baseline* model. Instead, we assume the adversary has trained a surrogate model to perform the same task. Papernot et al. [11] show that attacks generated using a surrogate model are often effective against a previously unseen model, even if the surrogate model is trained on different data to accomplish the same task.

It may be apparent that this omniscient setting is extremely unrealistic. Any adversary that has access to all sensor measurements with the ability to inject an attack to the localization model has an outsized influence on the model effectiveness. Instead, we consider omniscient attacks to be among the most extreme and effective attacks, providing context in which to view the more naïve attacks.

*4.2.1 Worst-Case Attack.* One way we represent an omniscient attack is using a worst-case attack, abbreviated as *A_WC*. A_WC uses the output prediction image from the CNN, which an adversary would not normally possess, to identify the location for a given RSS value that maximizes the error or distance from the true location of a transmitter. As with the naïve attack, we fix the RSS to either the 10th or 90th percentile of the current input samples to represent a plausible value and resist anomaly detection. These attacks, for 10th and 90th percentile RSS, are depicted in Figs. 3a and 3b, respectively.

*4.2.2 Fast Gradient Sign Method.* We also generate attacks using the *fast gradient sign method* (FGSM) from Goodfellow et al. [4], which uses the same backpropagation algorithm used to train neural networks to instead produce an attack based on the gradient with respect to the input image $X$. Let $\theta$ be the parameters of our model, $X$ be the input image, $Q$ be the localization target, and $J$ be the cost used to train the model using backpropagation. Then FSGM produces a perturbation vector:

$$\eta = \epsilon \cdot \text{sign}(\nabla_X J(\theta, X, Q))$$

In other words, $J$ is an objective which is solved to train the model for localization. When minimizing $J$ using gradient descent, the cost $J$ (the localization error) is decreased iteratively by taking the gradient with respect to the model parameters $\theta$, and updating those parameters in the opposite direction of the gradient $\nabla_\theta J$. FGSM uses the same formulation, but instead of updating model parameters, we take the sign of the gradient with respect to $X$, producing a perturbation vector $\eta$ with values in $\{-\epsilon, \epsilon\}$, which can be added to $X$ to increase the overall error.

Due to practical concerns on the amount of training data available in the dataset, we use a surrogate model trained using the same training set, rather than different data. Additionally, the surrogate model has a different architecture than the baseline model. The baseline model produces an approximation of the transmitter image, $\hat{Y}$, and the final prediction $\hat{Q}$ is the coordinates of the maximum value in $\hat{Y}$. However, taking the maximum-valued coordinates is not a differentiable operation, meaning that the baseline model cannot be used for FGSM attack if the goal is to maximize localization error. Instead, the surrogate model uses the same architecture as the baseline model, but we append 3 linear layers to the end which directly predict the transmitter coordinates $\hat{Q}$ as real numbers rather than pixel coordinates. This changes the objective from an image approximation problem to a regression problem.

Although FGSM produces a perturbation $\eta$ for every pixel in the image input $X$, an attack $h(\eta + X)$ is not valid in our setting, since we only consider attacks that perturb the sensor vector $S$, not the image $X$. Instead, we select a subset of values from $\eta$ to be used as perturbations to $S$.

*4.2.3 Attacks.* We consider three main types of attacks based on FGSM: sensor perturbation, withholding, and fake sensors. Sensor perturbation attacks are changing existing sensor values by $\pm\epsilon$; withholding attacks are concealing sensor values by removing entries from $S$. fake sensors are spoofed measurements provided to the crowdsourcing mechanism. It is possible to combine these attacks. We specifically consider the following attacks:

(1) `Top-n%`: This attack perturbs the top $n$ sensors with largest magnitude gradient in $\eta$ by a fixed constant $\epsilon$, where $n$ is a percentage of the total sensors ranging from 10-50%, with 100% as a special case.
(2) `Drop-n%`: This attack withholds the top $n$ sensors with largest magnitude gradient, with $n$ ranging from 10-50%.
(3) `Hi-Lo`: This attack produces fake sensors with either high or low values. The $m$ pixels with the largest positive gradients are assigned an RSS value of $\epsilon$, and the and $n$ largest negative gradients are assigned a small negative value of -0.03, which is halfway between the lowest measurement value and the average noise floor which is scaled to 0.

The `Hi-Lo` attack of producing fake sensors can be combined with the other attacks by applying perturbation or withholding. This is the most realistic of the omniscient attacks, where an adversary has control of a small set of real sensors, and then uses fake sensors for the rest of the attack.

The `Top-n%` and `Hi-Lo` attacks are both dependent on the parameter $\epsilon$. For these experiments, we consider $\epsilon \in [0, 0.5]$. The input data is normalized according to the maximum and minimum RSS values from each sensor, but in general a change in RSS of 0.1 represents between 5-10 dB of gain, with 7 dB being typical. In our dataset, we observe a high amount of variation in signal strength within a small locale. Calibration data shows up to 11 dB of variation at a fixed position due to changes in antenna rotation, so a value of $\epsilon = 0.2$ is a realistic amount of variation we might expect over a small area.

# 5 COUNTERMEASURES

## 5.1 Outlier Exclusion

A common defense or countermeasure used broadly in many different domains is the statistical outlier exclusion. Depending on the application and when paired with other defenses, this approach can be effective.

*5.1.1 Excluding Based on Distance.* We first consider a technique to exclude sensors based on location predictions. In our implementation of this countermeasure, we first obtain the model output $\hat{Q} = h(S)$ for the given input $S$. We then iterate through each sensor $s_i \in S$, removing one sensor at a time and comparing the new

predict $\hat{Q}_i = h(S \setminus \{s_i\})$ to the the original output $\hat{Q}$. If the difference $|\hat{Q} - \hat{Q}_i|$ exceeds a defined threshold $\gamma$ for any sensor $s_i$, then the sensor causing maximum difference is removed as an input. Formally, we remove the sensor $s_i$ which maximizes $\max_i |\hat{Q} - \hat{Q}_i|$ from the input set, if the difference is greater than the threshold $\gamma$. We use a distance of $\gamma = 50$ m as the threshold parameter, which was tuned by iterating over the training set and selecting a value that maximized the ratio of exclusions that improved the overall result vs exclusions that negatively affected the overall result. This countermeasure is referred to as Exclude Worst Outlier (C_EWO).

*5.1.2 Vector Outlier Exclusion.* This countermeasure still uses a statistical approach to remove outliers, but uses the entire output prediction image from the CNN instead of a single output prediction value for each input. The model output $\hat{Y} = h(S)$ is compared to the output prediction image $\hat{Y}_i = h(S \setminus \{s_i\})$, where we remove one sensor $s_i$ at a time. We calculate the relative change across each pixel, $E_i = \left| \frac{\hat{Y} - \hat{Y}_i}{\hat{Y}} \right|$. If any sensors cause a change in output relative to the original image that exceeds a threshold of $\gamma = 0.15$, then the sensor producing the maximum change $E_i$ is again excluded. This countermeasure was similarly tuned on unadulterated input data using the baseline CNN model. This countermeasure is referred to as Exclude Predicted Worst Vector Outlier (C_EPWVO).

## 5.2 Adversarial Training

As is common in the computer vision community [4, 11], we employ adversarial training as a defense against the threats described in Section 4. For this process, we take the trained CNN model and resume the training process, but with adversarial inputs. We generate these adversarial inputs using the omniscient attacks described in Section 4.2.3.

We have noted that the omniscient attacks are not realistic in a practical adversarial scenario. In this case, the aim of our adversarial training is not to defend against gradient-based attacks, but to evaluate if a model will become more robust to the more realistic attacks from Section 4.1 due to exposure to the FGSM-based inputs.

# 6 EVALUATION

## 6.1 Naïve Attacks and Countermeasures

We applied the *Insert Random Location, Random RSS attack* (A_IRLRR) to assess the general performance of our models and countermeasures. As depicted in Fig. 2, the attack success ratio was approximately 10% with little variation from the baseline to the AdvTr model. Notably, for this scenario we define a successful attack to be *any* result in which the location error increases relative to the estimated location for the base case. We applied, separately, the Exclude Worst Outlier (C_EWO) countermeasure and the Exclude Predicted Worst Vector Outlier (C_EPWVO) countermeasures to the outputs of the A_IRLRR which are also depicted in Fig. 2. In this case, neither of these countermeasures alone appeared effective in defending against the naïve A_IRLRR attack, and generally tended to appear to increase the attack success rate. This seems to occur because the outlier exclusion techniques will often remove inputs which would normally contribute to the accuracy of localization, though they appear as outliers relative to the other inputs. Indeed,

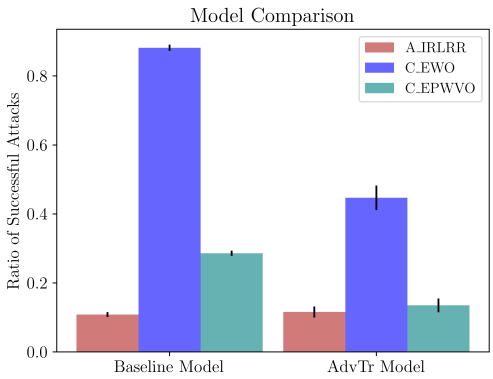



**Figure 2: Effectiveness of attacks and countermeasures for the baseline and adversarial training models over 3000 random test scenarios. A successful attack for this test is any attack that increases localization error.**

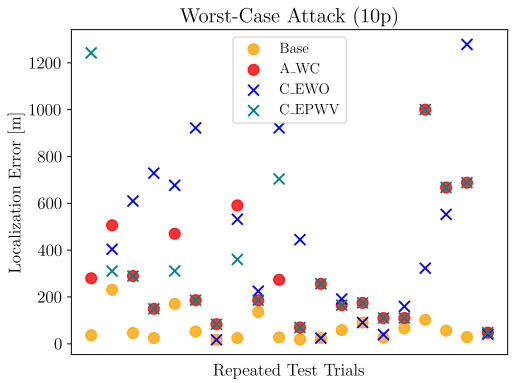
in other results not included here, we found that in many scenarios only a small percentage of the inputs materially contributed to the accuracy of the localization estimate. We observe a relative improvement in performance when either countermeasure is combined with the AdvTr model countermeasure. The C_EPWVO countermeasure may be combined with others to further improve performance against the A_IRLRR attack.

## 6.2 Omniscient Attacks

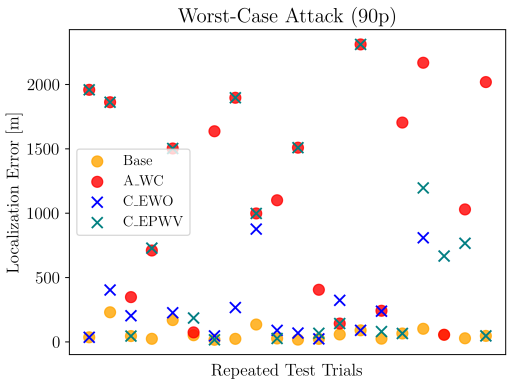
Here we evaluate attacks in the omniscient setting, where the adversary is aware of the transmitter location and has access to a localization model to test and produce attacks.

*6.2.1 Worst-Case Attack and Countermeasures.* In addition to the A_IRLRR attack, we evaluated the effectiveness of the C_EWO and C_EPWVO countermeasures against the worst case A_WC attack on the Baseline CNN model. For 20 test scenarios we first produced the base case estimate with unadulterated input values. We then applied the A_WC attack to the same inputs. We separately applied the countermeasures to the outputs of the A_WC attack. These results are presented in Fig. 3. For the 90th percentile A_IRLRR attack depicted in Fig. 3b, both countermeasures were effective in defending against the attack and reducing localization error relative to the base case in which no attack was present. The effectiveness of the countermeasures in defending against the 10th percentile A_IRLRR attack yielded mixed results.

*6.2.2 FGSM Attacks.* FGSM-based attacks were evaluated against both the baseline model and the adversarial trained model, denoted as *AdvTr* in figures. We evaluated attacks on each of the 828 samples in the test set. Since any gradient-based attack is extremely likely to cause significant increase in the model error, instead of reporting the number of successful attacks we instead report the median localization error. We note that both the baseline model and the adversarial trained model had a median error of 38 m and 36 m, respectively.




(a) Results for attack with 10th percentile RSS input.




(b) Results for attack with a 90th percentile RSS input.

**Figure 3: Attacks and countermeasures for worst-case scenario with bounded RSS values and optimized coordinate placement across twenty test trials. Base represents base case localization without any effects. A_WC is the worst-case attack scenario. C_EWO and C_EPWV are the *exclude worst outlier* and *exclude predicted worst vector outlier* countermeasures, respectively.**

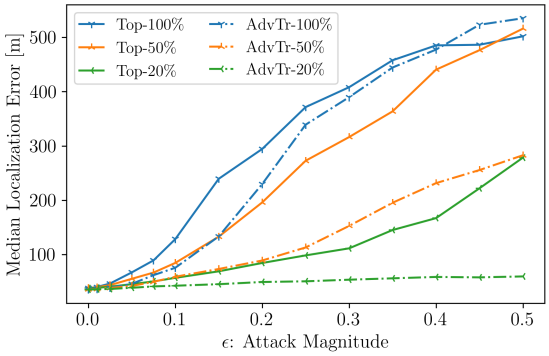



**Figure 4: The median error caused by `Top-n%` attacks, with and without adversarial training.**

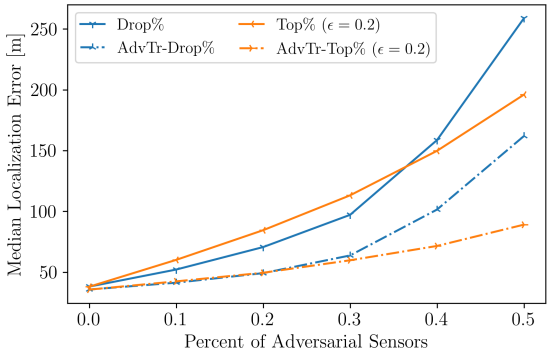



**Figure 5: The median error caused by `Drop-n%` and `Top-n%` attacks, with and without adversarial training.**

In Fig. 4 we show the success rate of the `Top-n%` attack. Here we add $\epsilon$ to the top 20%, 50%, and 100% of sensors, ordered by the magnitude of their pixel gradient. As mentioned before, we consider $\epsilon$ between 0 and 0.5. For clarity, Fig. 4 only shows a subset of the attack percentages, which ranged from 10-50%.

For the `Top-100%` attack adversarial training does not have a large impact on median error, but the improvement is drastic for the attacks with fewer adversarial sensors. The average improvement for the restricted case of 10-50% adversarial control increased with $\epsilon$, with an average improvement of 65% for $\epsilon = 0.5$.

*6.2.3 Drop-n% Attacks.* The `Drop-n%` attacks withhold a percentage of the highest gradient sensors, so it is independent of any constant $\epsilon$. In general, this attack was approximately as effective as a `Top-n%` attack with $\epsilon = 0.2$. These two attacks are shown in Fig. 5. They are similar in their effectiveness and the impact of adversarial training.

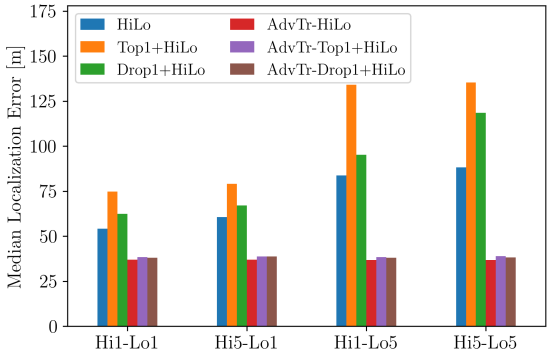



**Figure 6: The median error caused by `Hi-Lo` attacks ($\epsilon = 0.5$), with and without adversarial training.**

*6.2.4 Hi-Lo Attacks.* The `Hi-Lo` attacks, unlike the previous attacks, do not require the adversary to control a large percentage

of sensors. Instead, we inject low-RSS and high-RSS fake sensors into the sensor vector $S$, with values of -0.03 and $\epsilon$, respectively. In Fig. 6 we show results from adding 1 and 5 of each type of sensor. For the baseline model, the low-RSS sensors are particularly effective, since the `Lo5` attacks have a significantly higher median error than the `Lo1` attacks. The effectiveness of low-RSS sensors compared to high-RSS sensors may seem counter-intuitive. These fake sensors are almost identical to the 0-valued pixels that make up the majority of the image. Although it is difficult to determine exactly why these small negative values are impactful in this model, in neural networks, we assume that negative-valued pixels decrease the likelihood that the transmitter is near that location.

The `Hi-Lo` attack can also include sensors controlled by an adversary, as shown by the `Top1` and `Drop1` variants shown in Fig. 6, where a single sensor was either perturbed by $\epsilon$ or withheld by the adversary. The `Top1` attack was more effective than the `Drop1` variant. However, all these attacks were entirely neutralized by adversarial training.

## 7 DISCUSSION

The attacks executed in this work were effective at producing high error in an otherwise reliable localization system. The statistical outlier techniques alone were unable to effectively prevent attacks, but the effect of the statistical outlier approaches shown in Fig. 3b indicate some improved robustness in the case of high RSS values. We expect that more sophisticated countermeasures could provide additional robustness, especially when used in tandem with the C_EPWVO countermeasure.

One of our expectations at the outset of this work was that adversarial training would improve localization accuracy. We assumed that providing robustness to noise injection attacks would help improve robustness to existing noise in the training and test set, but this was not the case, with only a small accuracy improvement of 1.4 m on average. Research from computer vision [2, 8, 14] suggests that robustness to underlying noise cannot be provided by augmentations such as adversarial training.

However, the success of adversarial training as a countermeasure for both gradient and naïve attacks is effectively shown by our results. This also indicates that such training will also be an effective countermeasure against other "informed" attacks lying between the naïve and omniscient spaces. We suspect that such robustness may be provided even without training on such attacks specifically.

## 8 CONCLUSION

We evaluated the effectiveness of several attacks and countermeasures applied to a CNN-based RF localization application. We selected a set of representative attacks ranging from naïve to worst-case. We also evaluated two classes of countermeasures: statistical outlier approaches and an adaptive CNN trained with additional adversarial inputs. Both techniques show promise in providing a robust learning-based localization system.

A further area of study is the use of countermeasures such as an anomaly detection system in conjunction with other techniques to improve results further. We also plan to train the CNN on additional and specialized classes of adversary attacks, and in alternative contexts such as high-precision indoor localization, or with attacks from an informed adversary.

## ACKNOWLEDGMENTS

This work was supported by the National Science Foundation as part of awards CNS-1827940 and CNS-1564287, as well as the PAWR Project Office grant 10046930.

## REFERENCES

[1] Damilola Adesina, Chung-Chu Hsieh, Yalin E. Sagduyu, and Lijun Qian. 2023. Adversarial machine learning in wireless communications using rf data: a review. *IEEE Communications Surveys and Tutorials*, 25, 1, 77–100. DOI: 10.1109/COMST.2022.3205184.

[2] Aharon Azulay and Yair Weiss. 2019. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20, 1–25.

[3] Joe Breen et al. 2021. Powder: platform for open wireless data-driven experimental research. *Computer Networks*, 197, 108281.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

[5] Praveen Kaligineedi, Majid Khabbazian, and Vijay K. Bhargava. 2010. Malicious user detection in a cognitive radio cooperative sensing system. *IEEE Transactions on Wireless Communications*, 9, 8, 2488–2497. DOI: 10.1109/TWC.2010.061510.090395.

[6] Mojgan Khaledi, Mehrdad Khaledi, Shamik Sarkar, Sneha Kasera, Neal Patwari, Kurt Derr, and Samuel Ramirez. 2017. Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 235–247.

[7] Z. Luo, S. Zhao, Z. Lu, J. Xu, and Y. E. Sagduyu. 2022. When attackers meet ai: learning-empowered attacks in cooperative spectrum sensing. *IEEE Transactions on Mobile Computing*, 21, 05, (May 2022), 1892–1908. DOI: 10.1109/TMC.2020.3030061.

[8] Eric Mintun, Alexander Kirillov, and Saining Xie. 2021. On interaction between augmentations and corruptions in natural corruption robustness. *Advances in Neural Information Processing Systems*, 34, 3571–3583.

[9] Frost Mitchell, Aniqua Baset, Sneha Kumar Kasera, and Aditya Bhaskara. A Dataset of Outdoor RSS Measurements for Localization. Version 1.0. Zenodo, (Oct. 2022). DOI: 10.5281/zenodo.7259895.

[10] Frost Mitchell, Aniqua Baset, Neal Patwari, Sneha Kumar Kasera, and Aditya Bhaskara. 2022. Deep learning-based localization in limited data regimes. In *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, 15–20.

[11] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 506–519.

[12] Shamik Sarkar, Aniqua Baset, Harsimran Singh, Phillip Smith, Neal Patwari, Sneha Kasera, Kurt Derr, and Samuel Ramirez. 2020. Llocus: learning-based localization using crowdsourcing. In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 201–210.

[13] Harsimran Singh, Shamik Sarkar, Anuj Dimri, Aditya Bhaskara, Neal Patwari, Sneha Kasera, Samuel Ramirez, and Kurt Derr. 2018. Privacy enabled crowdsourced transmitter localization using adjusted measurements. In *2018 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, 95–106.

[14] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. 2020. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33, 18583–18599.

[15] Çağkan Yapar, Ron Levie, Gitta Kutyniok, and Giuseppe Caire. 2022. Locunet: fast urban positioning using radio maps and deep learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4063–4067.

[16] Caitao Zhan, Mohammad Ghaderibaneh, Pranjal Sahu, and Himanshu Gupta. 2022. Deepmtl pro: deep learning based multiple transmitter localization and power estimation. *Pervasive and Mobile Computing*. DOI: 10.1016/j.pmcj.2022.101582.

[17] Anatolij Zubow, Suzan Bayhan, Piotr Gawłowicz, and Falko Dressler. 2020. Deeptxfinder: multiple transmitter localization by deep learning in crowdsourced spectrum sensing. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–8.