Proximity-based location with robustness to Byzantine failures

Guillermo Hernandez, Shuo Tang, Pau Closas *Northeastern University*

BIOGRAPHY

Guillermo Hernandez is a PhD candidate in the Department of Electrical and Computer Engineering at Northeastern University, Boston, MA. His research interests include signal processing, applied cryptography, and GNSS positioning.

Shuo Tang got his BS degree in Vehicle Engineering from China Agricultural University, China and MS degree in Mechanical Engineering at Northeastern University. He is currently a PhD student in Electrical Computer Engineering at Northeastern University, Boston, MA. His research interest focuses on satellite signal processing, positioning and navigation, physics-based learning and computational statistics.

Pau Closas is an Associate Professor at Northeastern University, Boston, MA. He received the MS and PhD degrees in Electrical Engineering from UPC in 2003 and 2009. He also holds a MS in Advanced Mathematics from UPC, 2014. His primary areas of interest include statistical signal processing, robust stochastic filtering, and machine learning, with applications to positioning systems and wireless communications. He is the recipient of the 2014 EURASIP Best PhD Thesis Award, the 9th Duran Farell Award, the 2016 ION Early Achievements Award, and a 2019 NSF CAREER Award.

ABSTRACT

As more short-range communication devices, such as devices equipped with Bluetooth or WiFi are more commonly found within environment that suffer from NLOS or urban canyons, these devices can become part of a collaborative network to develop a sophisticated high-accuracy positioning solution. One such service is the proximity-based service (PBS), which utilizes a proximity-based positioning solution. In an ideal environment, the system can provide highly accurate estimates. However, in practical applications, the presence of outlier users decreases the accuracy of the estimates produced by the proximity-based positioning solution. Therefore, a robust proximity-based positioning solution is proposed to address this concern and improve the positioning accuracy of the PBS.

I. INTRODUCTION

The methodology of collaborative positioning continues to increase the interest of the GNSS community for its numerous benefits and emergence of applications. A collaborative positioning solution requires a device to share relevant position data with a neighboring user, in hopes of enhancing its position accuracy. Deploying this into a network of users has show to reduce the atmospheric effects observed in the GNSS signals (Garello et al., 2012; Minetto et al., 2017, 2022; Hernandez et al., 2023a). By mitigating these effects, the accuracy of the estimated position value is increased. Additionally, it has been proposed as a solution for urban canyon environments where GNSS receivers typically encounter non-line-of-sight (NLOS) conditions, stemming from the limited open-sky view and the inability to observe enough satellites. Its application is wide spread and it includes environments, such as smart cities, because of their common-to find Bluetooth Low-Energy (BLE) devices, WiFi, and other devices capable of short-range communication. In conjunction with employing such collaborative positioning solutions, additional services such as proximity-based services and location-based services can be found in these settings (Faragher and Harle, 2015) to assist in the process of enhancing the estimated position accuracy.

In an ideal situation, a user whose position is unknown (referred to as the p-th user) utilizes a proximity-based service, such as the one in (Hernandez et al., 2023b), to identify neighboring users (or devices) that are readily available to communicate with and are willing to provide their own position information. In this scenario, it is assumed that the users are all honest and fully operating users that can transmit their true and accurate position value, but in the application environment this may not always be the case. Any collaborating user may provide inaccurate position information due to system limitations or internal system malfunctions. Additionally, a malicious user may exist within the network, whose intention is to skew the operation of the service. As a result of this malicious user, the network may encounter a position deception or the accuracy performance of the system may decrease. This can lead to a Byzantine failure, where the malicious user or the unreliable collaborating user goes undetected.

This paper describes a proximity-based solution, that addresses the situation in which a device wishes to utilize a proximity-based service, receiving data from collaborating devices with unintentional or intentional anomalies. The proposed solution utilizes robust statistics methodologies to reduce the impact of these anomalies, and then compute an estimated position value. Robust statistics has been used in the GNSS context in several ways, namely, to improve the use of signals affected by interference (Borio et al., 2018) and the rejection of multipath in positioning computation (Medina et al., 2019). Using a similar approach and applying it to the context of a proximity-based solution, this method establishes an estimator that penalizes the outliers (or the anomalies) received by the p-th user whose location is unknown.

To analyze the efficiency of the proximity-based solution, the accuracy performance of the proposed robust proximity-based estimator was compared a proximity-based estimator (a non-robust estimator). Both of these estimator were evaluated in two simulated environments. The first simulated environment contained two outliers, and the number of valid users (non-outliers) increased up to 25 users. In the second simulated environment, the number of valid users was fixed at 25 users, while the number of outliers increased up to 50 users. In both environments, the proposed estimator provided near-optimal results, outperforming the proximity-based estimator. These results demonstrated the effectiveness of the proposed estimator compared to the proximity-based estimator.

Section II elaborates on the methodology of the proximity-based solution, explaining how to determine collaborating users and the underlying theory for computing an estimated position for the p-th user. In addition, it provides a brief explanation of how the described methodology is sensitive to outliers, thus leading to Byzantine failures. In Section III the robust estimator is introduced as a tool to addressed the existing vulnerabilities and it provides an explanation in how it is integrated with the proximity-based solution to create a robust proximity-based estimator. Section IV details the two different simulated environments to evaluate accuracy performance and its results of the proposed robust proximity-based estimator. Lastly, Section V concludes the paper with proposed future research.

II. PROXIMITY-BASED POSITION

The proximity-based positioning solution is comprised of a user whose location is unknown, denoted as the p-th user and its neighboring users. The neighboring users are indicated as neighboring users if they are within the p-th user's proximity area. The proximity area depends on the p-th user's signal receiving capacity which can be described and imposed in terms such as a signal-to-noise (SNR) ratio. When a neighboring user, denoted as the n-th user, is within the p-th user's proximity area it is assumed that it is able to communicate with the p-th user. When a set of neighboring users, a total of N users are present within the p-th user's proximity area, all the N neighboring users collaborate with each other to establish an estimated position of the p-th user.

The proximity area for the proximity-based positioning solution is described by its radius of Δ_t. By setting this radius parameter, it establishes the criteria that the n-th neighboring user's true distance from the p-th user is less than Δ_t. Each n-th user's observable measurement can be expressed in terms of the p-th user's true position as follows,

$$\mathbf{m}_n = \mathbf{m}_p + \boldsymbol{\delta}_n + \mathbf{w}_n \qquad n = 1, 2, \ldots, N \qquad \|\boldsymbol{\delta}_n\| < \Delta_t \tag{1}$$

where $\mathbf{m}_p$ denotes the true position of the p-th user, $\boldsymbol{\delta}_n$ denotes the distance vector between the n-th user and the p-th user, and $\mathbf{w}_n$ denotes the uncertainty within the n-th user's observable measurement. Each user in the network may experience such uncertainty within their own measurements and this leads to an influence on the outcome of the estimated position of the p-th user, denoted as $\hat{\mathbf{m}}_p$.

Based on the known knowledge of the uncertainty of the model in Eq. (1) different estimators can be established. Consider the observable measurement of the n-th user and assume it does not have an influence or impact on the observable measurement of another user within the network of users. This assumption will indicate that $\mathbf{w}_n$ for all the neighboring users are independent from one another. Furthermore, the distribution of $\mathbf{w}_n$ can range, but it is assumed it has a Gaussian distribution with zero-mean and it has a finite variance of $\sigma^2_{\mathbf{w}_n}\mathbf{I}$, i.e.,

$$\mathbf{w}_n \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{\mathbf{w}_n}\mathbf{I}). \tag{2}$$

Taking into consideration the uncertainty seen through $\mathbf{w}_n$ and the assumption that the observable measurements have a normal distributed around $\mathbf{m}_p$, then the likelihood function of Eq. (1) can be expressed by,

$$p(\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N | \mathbf{m}_p) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2_{\mathbf{w}_n}}} \exp\left(-\frac{(\mathbf{m}_n - (\mathbf{m}_p + \boldsymbol{\delta}_n))^2}{2\sigma^2_{\mathbf{w}_n}}\right). \tag{3}$$

Maximizing this likelihood function with respect to the estimated parameter, leads to the estimated position for the p-th user. In the situation where everything is known for the model in Eq. (1), $\hat{\mathbf{m}}_p$ can be computed using the optimal estimator, described as,

$$\hat{\mathbf{m}}_{\mathrm{p,OPT}} = \frac{\sum_{n=1}^{N}(\mathbf{m}_p - \boldsymbol{\delta}_n)(\boldsymbol{\sigma}_{\mathsf{w}_n}^{-2}\mathbf{I})}{\sum_{n=1}^{N}(\boldsymbol{\sigma}_{\mathsf{w}_n}^{-2}\mathbf{I})} \tag{4}$$

In the optimal estimator, $\boldsymbol{\delta}_n$ and $\boldsymbol{\sigma}_{\mathsf{w}_n}^2$ of the n-th user are both known. As a result of this knowledge, it leads the optimal estimator to be the most efficient estimator and achieve the lowest possible bound compared to any other estimator. This will indicate the optimal estimator to be the Cramér-Rao bound (CRB) for $\hat{\mathbf{m}}_n$ given the model in Eq. (1).

Although utilizing the optimal estimator will produce optimal results for $\hat{\mathbf{m}}_p$, the knowledge of $\boldsymbol{\delta}_n$ and $\boldsymbol{\sigma}_{\mathsf{w}_n}^2$ is usually unknown within the application environment. With limited knowledge of these parameters, assumptions about the parameters must be taking into consideration. Once an assumption is made a new estimator is created.

Suppose the following assumption is made where the true distance between the n-th user and the p-th user, $\boldsymbol{\delta}_n$, goes to zero, for all users. In addition, let the noise variance for all the n-th users be the same, meaning $\boldsymbol{\sigma}_{\mathbf{w}_n}^2 = \boldsymbol{\sigma}_{\mathbf{w}_m}^2$, where the m-th user is also part of the network of users and is not the same as the n-th user. With these assumptions, a new estimator can be established as,

$$\hat{\mathbf{m}}_{\mathrm{p,PRX}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{m}_n \tag{5}$$

where $\hat{\mathbf{m}}_{\mathrm{p,PRX}}$ denotes the estimated position for the p-th user, produced by the proximity-based estimator. The proximity-based estimator takes the average of all the observed measurements that are given by each user within the network. Although the proximity-based estimator's knowledge is limited, it is able to generate near-optimal estimated results when it has access to a large number of observable measurements and when there are not outliers.

1. Proximity-based Estimator Limitation

The proximity-based estimator is considered to be part of the ℓ_2-norm estimator framework which involves minimizing the sum of the squared differences between the observed measurements and the values predicted by the estimated parameters. This property is seen during the process of obtaining an estimator by means of maximizing the likelihood function, seen in Eq. (3), with respect to the estimated parameter. The step of maximizing the likelihood function can be transformed to maximizing the log-likelihood function. In this approach, with respect to the estimated parameter and ignoring the rest of the unrelated terms, the maximum likelihood estimator of the p-th user's position is given by

$$\hat{\mathbf{m}}_p = \arg\max_{\mathbf{m}_p} \; \log p(\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N|\mathbf{m}_p) = \arg\min_{\mathbf{m}_p} \sum_{n=1}^{N}[\mathbf{m}_n - (\mathbf{m}_\mathbf{p} - \boldsymbol{\delta}_n)]^2. \tag{6}$$

From Eq. (6), the ℓ_2-norm property of minimizing the sum of the squared differences between the observed measurements and the values predicted, or also know as residuals, is seen. Once this is computed, the results lead to the proximity-based estimator seen in Eq. (6). While the residual value is small, the ℓ_2-norm estimator could produce near-optimal results. Additionally, having more observable measurements can further improve the estimator's accuracy leading to near-optimal results. In the case when the residual value increases, then the ℓ_2-norm estimator's accuracy performance declines. The residual value may increase with the presence of an outlier. In this case, the ℓ_2-norm estimator's squared property amplifies the impact an outlier may have on the predicted estimated parameter, establishing the ℓ_2-norm estimator to be sensitive to outliers. Therefore, with the presence of the outlier, the proximity-based estimator can experience a decline in its accuracy performance because it can be vulnerable to any user that is outside of the considered proximity area or to a neighboring user experiencing internal system malfunctions.

A byzantine failure describes a system that fails to identify the presence of an anomaly (Lamport et al., 2019). In the case of the proximity-based positioning scheme, this experience can describe the presence of having an outlier within the observable measurement. As previously described when this failure occurs, the proximity-based positioning solution fails to perform at its highest possible performance level and reducing the integrity of the system.

There can exist numerous reasons why the n-th user can take part of the network of users for the proximity-based positioning solution while being considered an outlier. When using the SNR as a metric to indicate whether the n-th user can qualify as a valid user to be included within the network of users, the n-th user can manipulate and set a valid SNR value to create the illusion that it is closer to the p-th user's true position. Another case that can be taking into account, is when the n-th user is in the proximity area, but its transmitted position is incorrect. Unbeknownst to the p-th user, it would be consider an observable measurement, thus introducing an unforeseen biased within its estimated position calculations. There may arise a situation where the number of users in the proximity area is limited, and the p-th user might be inclined to accept a more scattered set of users in order to increase the overall count and, consequently, enhance its estimated position accuracy. This scenario could resonate closely with an application's intent, where the p-th user might be required to derive its estimated position using observable measurements from users who are considered outliers. In such occasions, there is an inclination to implement a robust estimator that can compute an estimated position for the p-th user that is close to its true position.

III. ROBUST PROXIMITY-BASED ESTIMATOR

Considering the impact that outliers have on non-robust estimators, a robust estimator offers a method to reduce this impact. There are a few different types of robust estimators, including the M-estimators which are related to the maximum likelihood estimation (Huber and Ronchetti, 2011). M-estimators depend on a loss function that is monotonic and can reduce the impact of outliers on the estimator's outcome. The M-estimator with respect to the estimated parameter $\hat{\mathbf{m}}_p$ can be described as follows,

$$\hat{\mathbf{m}}_{p,R} = \arg\max_{\mathbf{m}_p} \sum_{n=1}^{N} \rho(r_n) \tag{7}$$

where $\rho(\cdot)$ denotes the loss function that satisfies the two criteria, r_n denotes the residual between the observed data and the predicted value, and N denotes the total number of observed values within the given data set. Similarly to the two estimators seen in Eq. (4) and (5), the solution to the robust estimator can be found by computing the derivative of the loss function, with respect to the estimated parameter and setting it to zero, as seen as follows,

$$\sum_{n=1}^{N} \psi(r_n) \frac{\partial r_n}{\partial m_{i,p}} = 0 \quad \text{for } i = 1, \ldots, D \tag{8}$$

where $\psi(\cdot)$ denotes the derivative of $\rho(\cdot)$, with respect to the estimated parameter and D denotes the dimension of the estimated parameter. The $\psi(\cdot)$ determines how sensitive the estimator is to the residual and it is also known as the influence function (Huber and Ronchetti, 2011). A common method for solving for an estimated value is to use the Iteratively Re-weighted Least Squares (IRLS), which requires computing a weight function. The purpose of introducing the weight function is to adjust the influence of the residuals. This weight function depends on the loss function, and it is derived from the derivative of the loss function. This is shown as follows,

$$W(x) = \begin{cases} \psi(x)/x & x \neq 0 \\ \psi'(0) & x = 0 \end{cases} \tag{9}$$

With the weight function, the IRLS optimizes the estimated parameters by iteratively updating the weights of the residuals, resulting in the minimization of a weighted least squares problem. The dispersion of the residuals must be taken into consideration, as it can vary due to the presence of outliers. To address this dispersion, a normalization factor, denoted as $\hat{\sigma}$ is introduced to normalize this dispersion and ensure more reliable estimated values. With this normalization factor and the weight function, the IRLS procedure solves the following formulation, in an iterative manner,

$$\sum_{n=1}^{N} W(r_n/\hat{\sigma}) \frac{r_n}{\hat{\sigma}} \frac{(\partial r_n/\hat{\sigma})}{\partial m_{i,p}} = 0 \quad \text{for } i = 1, \ldots, D. \tag{10}$$

Using this alternative approach to compute an estimated parameter requires first initializing the estimate value. Then, the iteration portion of the IRLS involves computing the residuals, determining the normalization factor, calculating the weighted value for each residual (using the weight function), and finally solving for the weighted least squares. This iterative process continues until the difference between the current estimated value and the previous estimated value converges within a specified tolerance level.

Considering the methodology of robustness, it provides a method to address the present concerns of the proximity-based solution that were discussed in Section II. Therefore, to apply a robust estimator to the proximity-based solution, a loss function that satisfies the two criteria and a normalization factor are taken into consideration.

One of the loss functions commonly utilized for a M-estimator is the Huber's loss function (Zoubir et al., 2018). It satisfies the two criteria of being a monotonic function and it is equipped with the properties of penalizing outliers, or reducing their impact (Zoubir et al., 2018). The Huber's loss function depends on the degree of robustness parameter, denoted as γ, which sets the desired properties of allowing the loss function to have properties of an ℓ_1-norm and an ℓ_2-norm estimator, as described by,

$$\rho(\mathbf{x}) = \begin{cases} \frac{1}{2}\mathbf{x}^2 & \text{for } |\mathbf{x}| \leq \gamma, \\ \gamma(|\mathbf{x}| - \frac{1}{2}\gamma) & \text{for } |\mathbf{x}| > \gamma \end{cases} \quad \text{let } \mathbf{x} = \mathbf{m}_n - \hat{\mathbf{m}}_p \tag{11}$$

By Eq.(11), if the condition of $|\mathbf{x}| \leq \gamma$ is true, than the Huber's loss function will hold the properties of an ℓ_2-norm estimator. This will occur when the residual between the observed measurement and the predicted parameter is small, indicating the observed measurement is a non-outlier measurement value (or a valid user). In the case where the condition $|\mathbf{x}| > \gamma$ is satisfied, the observed measurement will be considered an outlier measurement and the loss function has ℓ_1-norm estimator characteristics.

The derivative of the Huber's loss function, with respect to the estimated parameter $\hat{\mathbf{m}}_p$, is given by,

$$\psi(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{for } |\mathbf{x}| \leq \gamma, \\ \gamma \cdot \text{sign}(\mathbf{x}) & \text{for } |\mathbf{x}| > \gamma \end{cases} \tag{12}$$

With its derivative, the weight function of the Huber's loss function is described as,

$$W(\mathbf{x}) = \begin{cases} 1 & \text{for } |\mathbf{x}| \leq \gamma, \\ \frac{\gamma}{|\mathbf{x}|} & \text{for } |\mathbf{x}| > \gamma \end{cases}. \tag{13}$$

By using the Huber loss function, along with its associated weight function and its properties of penalizing outliers, a robust estimator for $\hat{\mathbf{m}}_{\text{p,PRX}}$ can be established to obtain near-optimal results. Additionally, incorporating a normalization factor, such as the normalized median absolute deviation (MAD), $\hat{\sigma}_{\text{MAD}}$, helps to normalize the residual errors and improve robustness. The normalized MAD is defined as,

$$\hat{\sigma}_{\text{MAD}}(\mathbf{x}) = c_m \text{Med}(|\mathbf{x} - \text{Med}(\mathbf{x})|), \tag{14}$$

where $\text{Med}(\mathbf{x})$ is the median of $\mathbf{x}$ and c_m is the normalization constant (Rousseeuw and Croux, 1993). With this substitution, the observable measurement given by the n-th user is considered an outlier when its residual absolute value is greater than γ, otherwise it will be considered a valid user. For notation purposes, let N_i denote the total number of valid users and let N_o denote the total number of outliers. The summation of N_i and N_o is the total number of observable measurements or the total number of neighboring users, N, considered for the proximity-based solution. As for the weight for each residual, this is given by Eq. (13). Let the W_i denote the weight for the n-th valid user and let W_o denote the weight for the n-th user considered to be an outlier.

Using the IRLS method to estimate the position of the p-th user, an initial value for $\hat{\mathbf{m}}_p$ is set. The residual for the n-th user is then computed, followed by the normalization factor $\hat{\sigma}_{\text{MAD}}$, and the weight for each residual is obtained. This leads to the robust proximity-based estimator, defined by the following,

$$\hat{\mathbf{m}}^{(k)}_{\text{p,RPRX}} = \frac{\sum_i^{N_I} W_i^{(k)} \mathbf{m}_n + \sum_o^{N_O} W_o^{(k)} \mathbf{m}_n}{\sum^N W^{(k)}} \tag{15}$$

where $W^{(k)}$ includes the weight function for the valid and the outlier users for the k-th iteration. Therefore, Eq. (15) can be used as a substitute for Eq. (5) for the proximity-based solution.

IV. SIMULATED TEST AND RESULTS

The accuracy performance of the proximity-based estimator, seen in Eq. (5), and the robust proximity-based estimator, seen in Eq. (15), were analyzed and compared to one another within the two different simulated environments. In the first simulated environment, the objective was to evaluate the accuracy performance of each estimator with the constant presence of outliers and an increasing set of valid users. In the second simulated environment, the approach was similar to the first, but with a fixed set of valid users and an increasing number of outliers.

In the first simulated environment, the true position of the p-th user, $\mathbf{m}_p$ was set to a deterministic location and its proximity area was set to have a radius, i.e., $\Delta_t = 15$m. The environment was populated with a constant presence of two outliers. These outliers were set to be outside of the proximity area in order to introduce the offset of the outlier bias. The true distance between the p-th user and the outlier was set to be $\Delta_t < \|\boldsymbol{\delta}_n\| \leq 250$m. The observable measurement uncertainty was generated from a Gaussian distribution that had a variance of a meter. As for the valid users, the distance between the n-th user and p-th user was set to be $\|\boldsymbol{\delta}_n\| \leq \Delta_t$ and these users also had a Gaussian distribution that had a variance of a meter for its observable measurement uncertainty. As for the robust parameters, the degree of robustness parameter was set to the value given in (Zoubir et al., 2018) $\gamma = 1.345$ and the normalization factor for the normalized median absolute deviation (MAD) was also set to the value given in (Rousseeuw and Croux, 1993), $c_m = 1.4815$.

The first simulated environment was set to replicate an environment where there were two outliers present and the number of valid users increased. Initially, there were two n-th users within the p-th user's proximity area and two outliers ($N_o = 2$) outside the p-th user's proximity area. The number of valid users continued to increase until there were 25 n-th users present in the p-th user's proximity area ($N_i = 25$). At every valid user increment, the Monte Carlo model was used to obtain up to 1000 observation values from every user. At every observation, the estimated position of the p-th user was computed using the proximity-based estimator ($\hat{\mathbf{m}}_{\mathrm{p,PRX}}$) and the robust proximity-based estimator ($\hat{\mathbf{m}}_{\mathrm{p,RPRX}}$). In addition to these estimators, the optimal estimator, seen in Eq. (4), was used to compute the estimated position of the p-th user. Since it achieved the Cramér-Rao bound, it served as the benchmark for evaluating the performance of the other estimators. The Root Mean-Squared Error (RMSE) was used as a metric to evaluate the performance for each estimator. Fig. 1 plots the RMSE results from all the estimators in this simulation case.

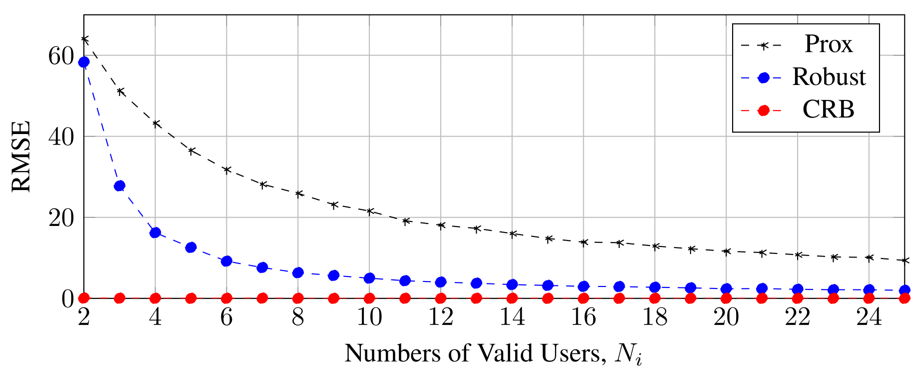

Figure 1: The RMSE results for the optimal, robust, and the proximity-based estimators, with at most $N_i = 25$ users within the proximity range. Through each iteration, the number of users set as outliers was set to $N_o = 2$ users.

The accuracy performance of each estimator, as seen in Fig. 1, improved as the number of valid users increased. While both estimators were affected by the outliers, their influence was greater on the proximity-based estimator. As a result, it required more valid users to achieve the same level of accuracy that the robust proximity-based estimator reached with fewer valid users. As seen in Section III, the robust estimator incorporated its ℓ_1-norm characteristics to the two outliers, reducing the influence of their residuals. This reduction became more effective as the number of valid users increased. For the valid users, the robust proximity-based estimator utilized its ℓ_2-norm properties to increase the weight of the residual given by these users. This behavior is reflected in Fig. 1 when more than 50 percent of total number of users, N are valid users. Therefore, based on these results, when there are significantly fewer outliers in the observable measurements, the robust proximity-based estimator produces near-optimal results with fewer valid users compared to the proximity-based estimator.

In the second simulated environment, the accuracy performance of the estimators $\hat{\mathbf{m}}_{\mathrm{p,PRX}}$ and $\hat{\mathbf{m}}_{\mathrm{p,RPRX}}$ were evaluated in a similar environment as in the first environment. In this case, the number of users within the p-th user's proximity area remained at a fixed set of $N_i = 25$ users and the number of users outside of the proximity area increased. The p-th user's proximity area had the same radius value of 15m and the area outside of the proximity area for the outliers was the same as before. The observable measurements had the same Gaussian distribution properties as in the first simulated environment. The degree of robustness parameter γ and the normalization factor constant c_m were also set to their respective values.

Similar to the first simulated environment, the Monte Carlo model was used for the second environment to generate the observable measurements. At each outlier increment, the model generated up to 1000 observations from each user, the outliers and the valid users. The RMSE was also considered as a metric to evaluate the accuracy performance of each estimator. For the proximity-based estimator, $\hat{\mathbf{m}}_{\mathrm{p,PRX}}$, the RMSE was computed using the observable measurements from all the users. Then the RMSE was computed using only the observable measurements of the the valid users, using the same estimator. This was done to eliminate the influence the outliers had on the estimators, thus serving as the estimator's benchmark. A similar process was performed for the robust proximity-based estimator, $\hat{\mathbf{m}}_{\mathrm{p,RPRX}}$.

From all the RMSE results, seen in Fig. 2, the proximity-based estimator had the worst performance, while the robust proximity-based estimator produced near-optimal results when the number of outliers was less than the number of valid users. In this case, the robust proximity-based estimator produced estimated values that closely resembled an estimator that did not experience the presence of outliers. In the case when the number of outliers exceeded the number of valid users, the robust estimator lost its efficiency and it reduced its ℓ_1-norm characteristics. As for the proximity-based estimator, its performance deteriorated as the number of outliers increased, eventually saturating at a level of error primarily influenced by the outliers.

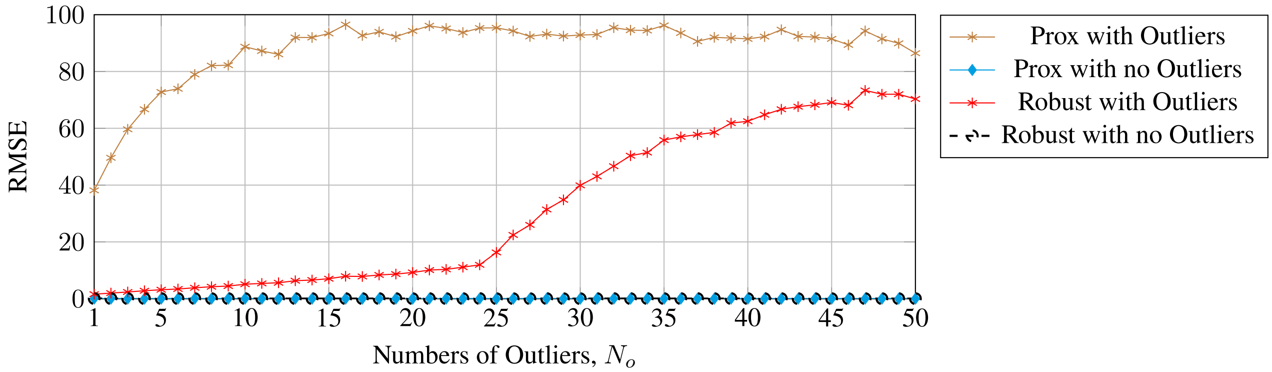

Figure 2: The RMSE was computed for both the proximity-based estimator and the robust estimator under two scenarios: with the $N_i = 25$ users within the proximity range and the presence of outlier users, and with users within the proximity range but without outliers. The total number of users within the proximity range was set to 25, and the number of present outliers ranged from 1 to 25.

Based on these results, the robust proximity-based estimator has a limit in its robustness properties, and when this occurs it loses its robustness behavior. This occurred when more than 50 percent of the observable measurements were outliers. In this case, the estimator was unable to distinguish between the outliers and the valid users. As a result, the weight of the residual provided by the outliers was not reduced causing the estimator to become bias towards the outliers. Consequently, instead of computing estimated values close to the p-th user's true position, it computed unreliable estimated values that lie between the p-th user's true position and the bias introduced by the outliers. As the number of outliers grew and the number of valid users remain fixed, the level of error went towards the same saturated level that was dictated by the influenced of the outliers, similar to the proximity-based estimator. In other words, the robust proximity-based estimator resemble the behavior of an ℓ_2-norm estimator after its robustness threshold.

Compared to the first simulation, the results of the second environment supported the same conclusion. When outliers account for less than 50 percent of the observable measurements, the robust proximity-based estimator is able to reduce the residual weights of the outliers, thereby minimizing their influence. In this case, the estimator provided estimated values that were close to an estimator that did not observe any outliers. It remained true while the robust proximity-based estimator operated below its robustness threshold. Below this threshold, the robust proximity-based estimator's ℓ_1-norm behavior dominated since it observed more valid users, leading to near-optimal results.

Based on these simulation results, the robust proximity-based estimator Eq. (15) serves as an effective alternative to the proximity-based estimator Eq. (5). The robust proximity-based estimator provides near-optimal results when the number of outliers is less than the number of valid users. Under these conditions, it requires fewer valid users to achieve near-optimal results. In contrast, the proximity-based estimator's ℓ_2-norm properties significantly hinder its performance, as it is highly sensitive to outliers. In environments that may desire to use such proximity-based solutions, having many valid users may not always be feasible due to resource limitations. Therefore, when valid users are limited and outliers are relatively low, the robust proximity-based estimator is a better option for estimating the position of the p-th user.

V. CONCLUSION

A proximity-based service can estimate a user's position by utilizing its neighboring users who are willing to take part of such service. By leveraging the observable measurements of these neighboring users, a highly accurate position can be computed for the user whose location is unknown. However, this approach relies on several assumptions, including the distribution of the uncertainty in the measurements and the trustworthiness of neighboring users. If the uncertainty is assumed to follow a Gaussian distribution, an ℓ_2-norm estimator can be used to compute the estimated position. However, these estimators are highly sensitive to outliers, which can arise for various reasons. Outliers may result from malicious users trying to skew the position estimates or from unreliable users providing inaccurate measurements. In such cases, the proximity-based service becomes vulnerable to a Byzantine failure.

Therefore, an alternative estimator was introduced to address the vulnerability of a Byzantine failure that an ℓ_2-norm may experience. To this end, the methodology of a robust estimator was integrated into the proximity-based estimator, resulting in a robust proximity-based estimator. This estimator penalizes the presence of outliers within a dataset, thereby reducing their influence on the estimated value. This estimator was evaluated under two simulated environments in order to analyze the accuracy performance. It was compared directly to the proximity-based estimator. The results showed the proximity-based estimator performance decreased almost immediately with the presence of a single outlier. As for the robust proximity-based estimator, its performance surpassed the proximity-based estimator. In both environments, it produced near-optimal results with the presence of outliers. After the number of outliers was more than 50 percent of the observable measurement values, the estimator lost its robustness properties and it began to behave more like the proximity-based estimator. From these results, the robust proximity-based estimator showed that it can serve as an alternative solution to the proximity-based estimator when the outliers accounted for less than 50 percent of the total number of neighboring users who are willing to collaborate in the proximity-based solution.

These results represent a significant step forward, providing a solid foundation for further investigation. In particular, they offer an opportunity to explore how the proposed proximity-based solution performs when processing data that contains sensitive information. Future research could examine how incorporating privacy measures impacts the robustness of the proposed solution while still effectively identifying and managing anomalies or outliers. Additionally, the proposed solution can be extended to evaluate its performance under different model assumptions, as seen in Eq. (1). As a benchmark, this solution paves the way for addressing these important considerations.

ACKNOWLEDGEMENTS

This work has been partially supported by the National Science Foundation under Awards ECCS-1845833 and CCF-2326559.

REFERENCES

Borio, D., Li, H., and Closas, P. (2018). Huber's non-linearity for gnss interference mitigation. *Sensors*, 18(7):2217.

Faragher, R. and Harle, R. (2015). Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428.

Garello, R., Samson, J., Spirito, M., and Wymeersch, H. (2012). Peer-to-peer cooperative positioning. *Inside GNSS*.

Hernandez, G., LaMountain, G., and Closas, P. (2023a). Privacy-preserving cooperative gnss positioning. *NAVIGATION: Journal of the Institute of Navigation*, 70(4).

Hernandez, G., LaMountain, G., and Closas, P. (2023b). Proximity-based positioning scheme with multi-layer privacy. In *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 235–242. IEEE.

Huber, P. J. and Ronchetti, E. M. (2011). *Robust statistics*. John Wiley & Sons.

Lamport, L., Shostak, R., and Pease, M. (2019). *The Byzantine generals problem*, page 203–226. Association for Computing Machinery, New York, NY, USA.

Medina, D., Li, H., Vilà-Valls, J., and Closas, P. (2019). Robust statistics for gnss positioning under harsh conditions: A useful tool? *Sensors*, 19(24):5402.

Minetto, A., Bello, M. C., and Dovis, F. (2022). DGNSS Cooperative Positioning in Mobile Smart Devices: A Proof of Concept. *IEEE Transactions on Vehicular Technology*, 71(4):3480–3494.

Minetto, A., Cristodaro, C., and Dovis, F. (2017). A collaborative method for positioning based on GNSS inter agent range estimation. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2714–2718. IEEE.

Rousseeuw, P. J. and Croux, C. (1993). Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88(424):1273–1283.

Zoubir, A. M., Koivunen, V., Ollila, E., and Muma, M. (2018). *Robust statistics for signal processing*. Cambridge University Press.