GNSS Anti-spoofing Detection based on Gaussian Mixture Model Machine Learning

Zejian Feng[1], Chee Kiat Seow[1], Qi Cao[1]

Abstract—Nowadays, the security of Global Navigation Satellite System (GNSS) has raised much more concerns due to the reliance on its position, velocity, and timing (PVT) information, which is of vital importance to various Internet of Things (IoT) systems, robotics, 5G technology and many applications of Intelligent Transportation Systems (ITSC). It has been shown that GNSS system can be easily spoofed and masqueraded to provide ill intent payload damages. This paper proposes a novel algorithm based on unsupervised machine learning Gaussian Mixture Models (GMM) to provide anti-spoofing capability of GNSS signal such as GPS signal. It segregates GPS signals that are not under spoofing, from spoofed GPS signals that will result in malicious changes of pseudo-range measurements. It has been found out that the proposed GMM clustering algorithm is able to cluster the positions generated by the un-spoofed GPS signals properly and return the PRN (pseudo-range noise) codes of the satellites without spoofing effectively. The proposed GMM clustering algorithm could cluster the position points generated by non-spoofed signals properly by more than 90% and 77% accuracy for one and three spoofed satellites respectively.

I. Introduction

Currently, position, velocity and timing (PVT) information of Global Navigation Satellite System (GNSS) has been utilized by various Internet of Things (IoT) systems, robotics, autonomous vehicle (AV) and 5G technology applications [1]. With the rapid development of applications with GNSS positioning, the concerns have been raised on the security of GNSS signals which are used to provide PVT information [2]. For example, robots used for logistics could be spoofed and misled to another target position, which is likely to be utilized as a place to make illegal behaviors such as robbery or smuggling. Apart from that, autonomous driving requires the integrity and reliability of the position information, otherwise it will cause serious accidents by a spoofed PVT information without detection[3].

On top of that, more concerns have been raised on the security and integrity of GNSS signals especially by research community in Intelligent Transportation Systems (ITSC). To be specific, there are various applications in ITSC that require reliability and integrity in the accuracy and security of PVT information, such as vehicle-to-everything (V2X) communication [4], pedestrian detection based on GNSS sensors [5], collision avoidance for automotive based on navigation satellites [6]. For these applications that have high reliance in positioning solution, it is of vital importance to make sure the PVT information such as the position that the GNSS receivers acquire from satellites are not only highly accurate but also trustable.

In this paper, we propose an machine learning algorithm based on Gaussian Mixture Model (GMM) and least squares method for localization, to detect and exclude false GNSS signals of which pseudo-range measurements have been altered for malicious intent. The GMM algorithm pivots on finding the mean value and variance of each cluster, which is quite suitable to detect false signals, due to the differences of the mean and variance between the positions generated by spoofed signals and non-spoofed signals. We also find a general rule to obtain essential parameters of the GMM as the number of clusters in different scenarios, which is to ensure optimality in clustering performance.

This paper provides a brief introduction of the GNSS ant-spoofing aims and presents the proposed algorithm to achieve the goals. Section II gives the literature review on related works about GNSS anti-spoofing methods which are mainly for attacks on altering pseudo-range measurements in a malicious way. Section III outlines the details of positioning algorithm based on GPS observation and navigation data from NASA. The proposed GMM model is also depicted. Section IV evaluates the performance of our proposed GMM clustering algorithm compared with other clustering algorithms. Finally, conclusion and discussion on the proposed GMM algorithm in the typical scenarios are provided in Section V.

II. Related Works

A. GNSS vulnerability to spoofing

In general, the vulnerability of GNSS for spoofing is classified into three main categories from the perspectives of GNSS receivers: (a) the GNSS navigation message (NM) data bits, (b) GNSS signal processing, and (c) the position and navigation solution [7,8].

For the GNSS NM data bits that are transmitted from the respective satellite to receivers, its structure is openly available to public. Therefore, it is quite easy to get the common structure of each GNSS satellite's NM publicly and regenerate a similar one to mislead the GNSS receivers. The navigation data is composed of various parameters such as satellite ephemeris, almanac, time, telemetry information, and authentication keys [9]. Most of the information in navigation messages do not change very often or not fast enough. For example, the satellite ephemeris usually changes for every

[1]All authors are with School of Computing Science, University of Glasgow, United Kingdom (e-mail: 2535290F@student.gla.ac.uk, CheeKiat.Seow@glasgow.ac.uk, Qi.Cao@glasgow.ac.uk)

12.5 minutes [7]. As a result, the spoofer can forge the GNSS signal with this feature, and change navigation information that is essential for computing PVT solutions to apply a spoofing effect on the GNSS receivers.

Similarly, for GNSS signal processing, the structure of most GNSS signals are published as the common frame, which includes pseudo-random noise (PRN) signals, the modulation type, transmit frequency, signal bandwidth, Doppler range, and signal strength [7]. Therefore, it is not difficult for spoofers to generate fake signals with the general structure of GNSS signals and implement malicious spoofing for ill intent.

For solution of position and navigation, the pseudo-range measurements could be changed in several ways, including jamming the authentic satellite signals, injecting fake pseudo-range measurements by spoofers [7], or altering the time-offsets of a signal [9]. There are various methods used to change the pseudo-range measurements, which lead to the deviations from the right PVT solutions in assorted ways. In this paper, we will focus on the impact and extent due to the changes in pseudo-range measurement rather than the methods of the attacks that lead to the changes in pseudo-range measurements.

B. *GNSS spoofing and anti-spoofing with ML algorithms*

(1) Spoofing simulation by ML

Nowadays, a few of studies have been made on spoofing simulation by machine learning. For example, a GNSS spoofing method with adversary attacks model as a deep learning model was reported [10], which can mislead the GNSS receiver to a malicious target position without being detected by the receiver autonomous integrity monitoring (RAIM) algorithm. The paper [10] applies GAN to generate forged signals with its generator and train its discriminator to detect forged signals from original signals in order to counterfeit fake signals that can't be detected easily. However, it can only work within a limited range if the spoofing goal is to mislead the victim to a targeted position.

(2) Anti-spoofing methods by ML algorithms

Anti-spoofing against satellite NM and signal attack have been well-researched through authentication of the NM and signal [8]. This paper will focus on attack that cannot be protected through authentication. Such research focusses on anti-spoofing methods with ML algorithms. A possible solution that utilizes Support Vector machines (SVM) to classify the spoofed signals and original signals was introduced in [11]. The results were robust with the validation on the unintentional spoofed subsets. However, it was also reported in [11] that the separation of spoofed signals and original signals based on different multiple measurements of GNSS signals using SVM is very complicated and thus difficult to be deployed. Apart from that, a method to detect spoofed signals based on an improved RAIM algorithm was introduced in [12], which combined Density-Based Spatial Clustering of Applications and Noise (DBSCAN) algorithm for single constellation. Nevertheless, RAIM algorithm combined with DBSCAN get poor performance very often when it comes to data sets in high dimensions especially with varying density clusters [12]. What's more, A clustering-based solution separation algorithm (CSSA) has been presented in [1] to detect spoofing in multi-constellation as well as single constellation. The CSSA can identify the small changes of pseudo-range measurements generated by spoofer. However, CSSA is not customized for single constellation anti-spoofing, especially when there is only one GNSS system.

In this paper, we propose a novel anti-spoofing algorithm that overcomes the above-mentioned limitations. We use the Gaussian Mixture Models (GMM) to cluster original signals from spoofed signals. As such, the mean and the variance of the original signals can be taken into account and thus make it more likely to sort out the original signals among spoofed signals.

III. GNSS Positioning Algorithm and GMM

A. *GNSS Positioning Algorithm*

GNSS receivers use measurement observations that are received from GNSS satellites and satellites orbital position information to calculate the positions of GNSS receivers. The observations mainly contain three main components: phase, time and pseudo-range. These values are stored in the Observation data file [13]. These observations should be corrected to avoid the external effects such as atmospheric refraction, clock offsets, etc. [13]. The positions of satellites can be computed with navigation messages sent from GNSS satellites, which are stored in Navigation file. The time of observations represent the time of the GNSS signals received by the receiver, which can be affected by clock offsets between satellites time stamp and that of receivers. Pseudo-range of observations is defined as the distance between the receiver and the satellite with the consideration of clock offsets and other biases such as atmospheric delays [13]. The equation of pseudo-range is shown in Eq. (1) as follows.

$$\rho^i=\sqrt{\left(p_{ix}^S-p_x^R\right)^2+\left(p_{iy}^S-p_y^R\right)^2+\left(p_{iz}^S-p_z^R\right)^2}+c(\Delta t+\text{other biases}) \quad (1)$$

where $\left[p_{ix}^S \quad p_{iy}^S \quad p_{iz}^S\right]$ and ρ^i represent the three- dimensional position of i^{th} satellite and its pseudo-range respectively, where the position of satellites can be computed with the observation data. $\left[p_x^R \quad p_y^R \quad p_z^R\right]$ represents three-dimensional position of the receiver [10], c is the speed of light in vacuum, and Δt is the clock offsets [13]. In this paper, to get the position of the receiver, the least-square method is used to calculate the position iteratively with pseudo-range measurements and clock offsets [14,15]. There are four unknown variables, namely $\left[p_x^R \quad p_y^R \quad p_z^R\right]$ and the clock offset Δt [16]. As such, there should be at least four satellites' pseudo-range being used to calculate the position of receiver. The least squares method implemented on calculation of positions of the receiver can be decomposed into several steps, the core steps can be written in Eq. (2), Eq. (3), Eq. (5) and Eq. (6). Because pseudo-range measurements of i^{th} satellite ρ_o^i can be extracted from observation data, and the pseudo-range measurements of i^{th} satellite i.e., ρ^i, can also be calculated in Eq. (1), there are always differences between these two pseudo-range measurements. The equation of the differences can be written as in Eq. (2):

$$r^i = \rho_o^i - \rho^i \quad (2)$$

where r^i is the difference between these two kinds of pseudo-range measurements of i^{th}satellite, where $i = 1 \cdots I.$; I is the number of all available satellites used to calculate positions of the receiver. And the difference can also be written in the matrix form in Eq. (3):

$$r = H_q \Delta z + \varepsilon \tag{3}$$

$$H_q = \begin{bmatrix} \frac{p_{1x}^S - p_{qx}^R}{\rho^1} & \frac{p_{1y}^S - p_{qy}^R}{\rho^1} & \frac{p_{1z}^S - p_{qz}^R}{\rho^1} & 1 \\ \frac{p_{2x}^S - p_{qx}^R}{\rho^2} & \frac{p_{2y}^S - p_{qy}^R}{\rho^2} & \frac{p_{2z}^S - p_{qz}^R}{\rho^2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{p_{Ix}^S - p_{qx}^R}{\rho^I} & \frac{p_{Iy}^S - p_{qy}^R}{\rho^I} & \frac{p_{Iz}^S - p_{qz}^R}{\rho^I} & 1 \end{bmatrix} \tag{4}$$

where $r \in R^{I\times 1} = [r^1 \quad \cdots \quad r^I]^T$. The subscript q refers to the iteration number where $q=1\cdots Q$, p_{qx}^R represents the position of receiver in x coordinate in the q^{th} iteration; $\Delta z \in R^{4x1} = [\Delta p_x^R \quad \Delta p_y^R \quad \Delta p_z^R \quad \Delta t]^T$ is the difference of receiver state between two iterations of the least square method. $\varepsilon \in R^{Ix1}$ is the biases such as Gaussian measurement noise [10]. The solution of Eq. (3) can be written in Eq. (5):

$$\Delta z = (H^T H)^{-1} H^T r \tag{5}$$

And a new state of the receiver is updated based on the solution of Eq. (5) as follows:

$$z^q = z^{q-1} + \Delta z \tag{6}$$

where $z \in R^{4x1} = [p_x^R \quad p_y^R \quad p_z^R \quad \Delta t]^T$.The value of receiver position in z^1 is initialized to be zero Eq. (5) and Eq. (6) will go through q^{th} iteration till Δz is below a certain threshold [17]. In this paper, the threshold is set as 0.001. The differences between real position and averaged position of positions calculated by this positioning algorithm are [0.0432 m, 0.5336 m, 2.7406 m] in three dimensions, which is the best performance in our experiments.

B. Gaussian Mixture Models

The GMM is a typical unsupervised machine learning algorithm used as a clustering algorithm for data points that can be clustered into multiple Gaussian distributions. In the GMM, data points are considered to be generated by multivariate Gaussian distributions with relative mean and variance [17]. Additionally, GMM is a probabilistic model to predict the distribution each data point arises. The probability of each data point is calculated by Eq. (7).

$$P(x_i) = \sum_{j=1}^{K} \pi_j N(x_i | \mu_j, \Sigma_j) \tag{7}$$

where $N(x_i|\mu_j, \Sigma_j)$ is the Gaussian distribution of x_i conditioned on μ_j and Σ_j. x_i is the i^{th} data point and K is the number of Gaussian clusters. It is assumed that the total number of all data points is S, and μ_j and Σ_j are the mean and the covariance matrix of j^{th} cluster respectively [18]. π_j is the weight of j^{th} cluster to be learnt by GMM algorithm where the sum of all weights is equal to 1. Furthermore, $\{\pi_j, \mu_j, \Sigma_j\}$ are estimated by maximizing log-likelihood function in Eq. (8).

$$\ln P(x_1, x_2, \ldots x_S | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{S} \ln \{ \sum_{j=1}^{K} \pi_j N(x_i | \mu_j, \Sigma_j) \} \tag{8}$$

Expectation-Maximization (EM) algorithm is used to maximize Eq. (8) [19]. The EM algorithm begins with an initialization of the parameters π_j, μ_j, Σ_j for j^{th} cluster with a predefined number of clusters, i.e., K. The EM algorithm consists of the E-step and M-step.

E-step:

$$E_{i,j} = \frac{\pi_j N(x_i|\mu_j, \Sigma_j)}{\sum_{j=1}^{K} \pi_j N(x_i|\mu_j, \Sigma_j)} \tag{9}$$

As shown in Eq. (7), the E-step aims to estimate how likely each data point will be assigned into a particular cluster. The probability of i^{th} data point being assigned into j^{th} cluster is calculated with $N(x_i|\mu_j, \Sigma_j)$.

M-step:

The M-step is mainly for updating parameters π, μ, Σ based on maximizing Eq. (8). For example, the new weight of each cluster (π_j) is calculated in Eq. (10).

$$\pi_j = \frac{M}{S} \tag{10}$$

where *S* is the number of all data points and *M* is the number of data points assigned to j^{th} cluster. The mean and covariance matrix are also updated with data points in each cluster, which is also updated by maximization of Eq. (8) [20]. After the updates of these parameters, it will turn into the E-step again for the next iteration. This iterative process will stop until the log-likelihood function reaches the maximum, which means the EM algorithm converges. In this paper, we cluster the data of combinations of positions by using GMM package in scikit-learn [21].

IV. Implementation and Evaluation

A. Simulation setup

We use the GPS navigation data and observation data of a GNSS receiver downloaded from NASA's archive of Space Geodesy Data in RINEX format [22] to compute the position of this receiver with the positioning model. We choose a static position to experiment, at which one of the GNSS receiver stations (i.e., ZIMM00CHE, Switzerland) of International GNSS Service (IGS) locates. The data contains the GPS navigation data and observation data on 4th March 2021 with 30 seconds interval.

There are 10 visible satellites for this part of recorded data of which the PRN sequence is {1, 31, 3, 6, 21, 9, 17, 22, 19, 4}. We simulate the attacks by manipulating pseudo-range measurements of different numbers of satellites that are randomly chosen as being spoofed. The changes in pseudo-range measurement are set as various values of {30, 40, 50, 100, 200} meters. For instance, if we have three spoofed satellites, {1, 31, 3} can be the PRN of spoofed satellites in one run, and {31, 6, 21} is possible to be selected as the PRN of spoofed satellites in another run. In this scenario, we combine five satellites for each positioning, by which we generate three-dimensional position data points. The clustering is based on the data points of positions generated by the combinations. It is easy to calculate that the number of all

possible combinations is C_{10}^{5}. The reason for the combination size of five satellites are as follows: First, the least number of satellites required to generate a position is four. Secondly, we have compared the root mean square error (RMSE), mean deviation and variance performance between the calculated positions and real positions with combination size of 5, 6 and 7 satellites, as shown in Fig. 1. It is observed that the performance knee point is satellite size of 5. As such, in order to improve the ability of anti-spoofing, i.e. the number of spoofed satellites can be detected, 5 satellites is chosen as the size of combination.

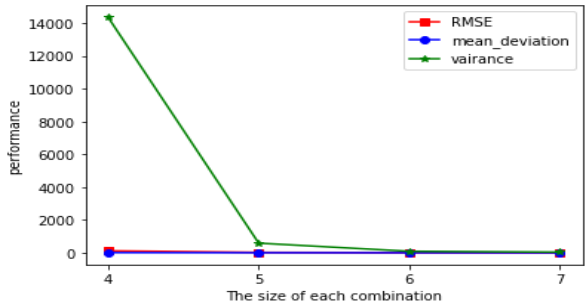

Fig. 1. Performance of different combination size (Units of Mean_deviation and RMSE are in meters, Variance is in square meters)

B. GMM implementation

The number of spoofed satellites is unknown in the real spoofing attack is made. Therefore, the number of spoofed satellites is chosen to be one at the beginning, and then it's added one until the algorithm converges. The metric used to evaluate the performance of clustering is mainly RMSE, with the mean deviation and variance as the supplement, where the mean deviation represents differences between the real position and the mean of calculated positions in one dimension on average. The cluster number is firstly set to 2. Then, we check if the smallest RMSE of all the clusters is below 10, where 10 is the threshold to determine whether we have found the clean cluster. If the smallest RMSE doesn't meet the threshold, the process will be iterative with adding one to the cluster number till we find a cluster that meets the threshold.

It is observed from the experiments that the number of clusters is highly relative to the number of spoofed satellites. It can be proven that when the number of clusters is equal to 2^n or larger around 2^n, where n is also the number of spoofed satellites, the algorithm converges faster. For example, if we have two spoofed satellites, PRN of which is {1, 31}, C_2^0 represents there are no spoofed satellites in the combination of satellites like {3, 6, 21, 9, 17}. C_2^1 represents there are two kinds of spoofed satellites, for example, {1, 3, 6, 21, 9}. C_2^2 represents the combination contains these two spoofed satellites, such as {1, 31, 6, 21, 17}. It's been proved by our experiments that GMM always groups the position data points generated by combinations of satellites with the same number of spoofed satellites into the same cluster. Therefore, by computing $C_2^0+C_2^1+C_2^2 = 2^2 = 4$, there should be four clusters in ideal conditions. For other scenarios that the number of spoofed satellites is not 2, the number of clusters still meets the rule. In our experiments, sometimes the number of clusters should be set larger than 2^n to make our algorithm converge faster. There might be one or two more categories than ours in the perspective of ML, therefore, we usually first set the number of clusters into 2^n and add by 1 to check for convergence. The ceiling is usually less than 2^{n+1}. The rule is to make it convenient to set the number of clusters at the beginning in a roughly proper range. Therefore, in this paper, it helps a lot to make our method more efficient to get a great performance.

C. Simulation results and analysis

We simulate the spoofing attacks by manipulating pseudo-range measurements with additional different numeric values: {30, 40, 50, 100, 200} meters. In the experiment, three satellites are manipulated by changing pseudo-range measurements from a set of clean satellites signals. Therefore, there are three situations in the experiment: (1) One spoofed satellite, (2) Two spoofed satellites, (3) Three spoofed satellites.

Taking the three spoofed satellites as an example, the results of clustering based on the GMM in two-dimensional (2D) vision applied with PCA projection are shown in Fig. 2. They clearly show that the cluster with purple color is more compact than other clusters and thus it's the clean cluster. We separate out the clean cluster by comparing the RMSE of different clusters with mean deviation and variance as supplement. Fig. 3 shows the performance of the clustering result of the example with three spoofed satellites.

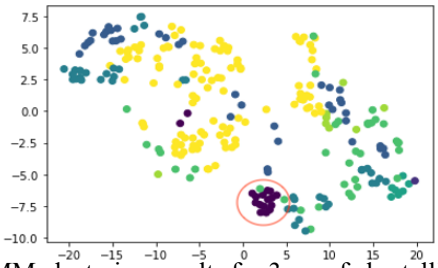

Fig. 2. GMM clustering results for 3 spoofed satellites, with pseudo range measurements changed by 50 meters 2D vision (PCA projection)

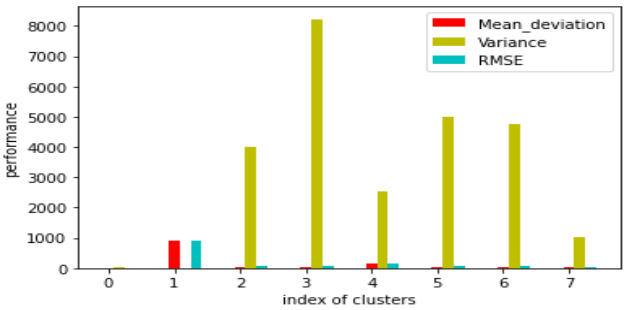

Fig. 3. Clustering performance of the example with 3 spoofed satellites (Mean_deviation and RMSE are in meters, Variance is in square meters

Because there are 3 spoofed satellites, it is observed that eight clusters are achieved convergent, which meets the rule of setting the number of clusters. What is presented vividly in the bar chart is that the clean cluster (cluster 0) has the smallest RMSE among these clusters, which is below 10 meters. It's relatively hard to see RMSE of cluster 0 in the chart, so is the

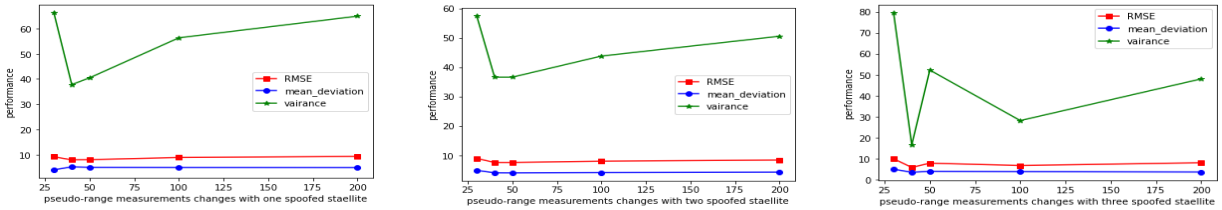

Fig. 4. GMM performance in situations with 1,2,3 spoofed satellites (Mean_deviation and RMSE are in meters, Variance is in square meters)

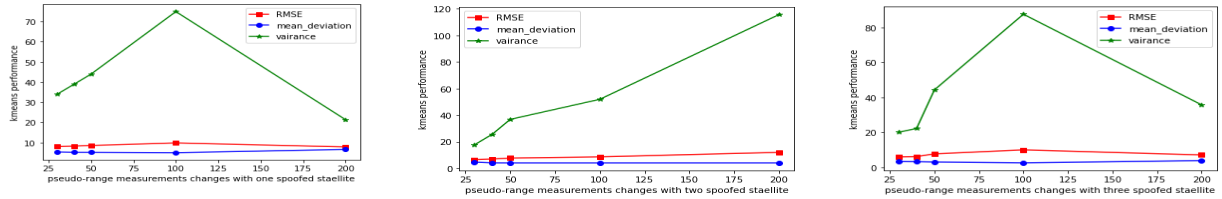

Fig. 5. K-means performance of the clean cluster in situations with 1,2,3 spoofed satellites (Mean_deviation and RMSE are in meters, Variance is in square meters)

mean deviation of clean Cluster 0. But its variance value is non-zero which is shown in Fig. 3. The mean deviation and variance of clean cluster 0 are also the smallest, compared to other clusters.

In the experiments, the spoofing attacks have been tested in these three situations with different pseudo-range measurements manipulations. The most quickly convergent clustering performance of the proposed GMM are shown in Fig. 4, while those of the K-Means are shown in Fig. 5.

It is observed that the performance of GMM with changes in pseudo-range measurements by 30 meters in all three situations are relatively worse than others. So when the pseudo-range measurements are manipulated by more than 30 meters, GMM can achieve a good performance. The variances in all circumstances are always below 60, while the mean deviation in all conditions are always below 5 meters and sometimes under 2 meters. The RMSE must be under 10 because the algorithm converges once RMSE is below 10, and thus RMSE is always close to 10.

What's more, there might be some prediction errors in clean cluster, such as missing clean data points or outliers. To be specific, some clean data points might be assigned into spoofed clusters instead of the clean cluster due to common predict errors of GMM, therefore, the issue of missing clean data points occurs in all circumstances and it's the main source of prediction errors when more than 50 meters pseudo-range measurements are manipulated. Outliers refers to some spoofed data points are assigned into the clean cluster by mistakes, which leads to a more serious prediction error than missing data points. A clean cluster should be composed of clean data points that give arises to correct receiver position. Outlier data points will affect the originality of the receiver position. As such, we put more weight on the number of outliers than that of missing clean data points especially when comparing performance of two algorithms in the same experimental condition.

Furthermore, we define the accuracy as the proportion of position data points generated by clean satellites in the clean cluster. The accuracy is directly affected by the number of missing clean data points and outliers. When the accuracy of two situations are equivalent, we will compare the number of outliers, which are shown in parentheses in Table I.

D. Comparisons with K-means

It is shown that K-means has a comparable performance with GMM from the perspective of RMSE and mean deviation. However, if there are a huge number of missing points, it might degrade the effectiveness of original GNSS signals. What's more, if there are too many outliers in the clean cluster, it can disable the detection of PRN of satellites without spoofing. It will also be affected with too many lost clean data points in the clean cluster. However, there are always many outliers and missing data points at the same time in the clean cluster generated by K-means. In this circumstance, though the mean, the RMSE and variance of the clean cluster are relatively good, the spoofed signals are recognized as the clean signals based on K-means, which might cause failed detection. The accuracy and number of outliers shown in Table II clearly demonstrate this feature of K-means. It is observed from Table I and II that GMM has a higher average accuracy in situations with one spoofed satellite with same amount of outliers on average. Although the GMM averaged accuracies of situations with two or three spoofed satellites are slightly lower than those of K-means, the amounts of outliers of the proposed GMM are much smaller than those of K-means.

In addition, if the size of clean cluster is equal or larger than the value of C_{K+2}^{K}, and $(K + 2)$ are the number of satellites without being spoofed, K is the lower bound number of satellites, our method is always effective. In summary, GMM is a more effective method of anti-spoofing for GNSS signals than K-means. Meanwhile, the proposed GMM method is easily implemented without equipment of anti-spoofing technique, which decreases the complexity of anti-spoofing and makes it widely used in many appropriate scenarios.

V. Discussion and Conclusion

This paper proposes a novel algorithm to detect spoofing GNSS signals, especially for manipulations of pseudo-range

Pseudo-range changes(m) / Situation	30	40	50	100	200	average
One spoofed satellite	0.8968 (14 outliers)	0.8968 (5 outliers)	0.8968 (0 outlier)	0.9286 (0 outlier)	0.9603 (0 outlier)	0.9159 (3.8outliers)
Two spoofed satellites	0.7857 (6 outliers)	0.8929 (4 outliers)	0.8036 (0 outlier)	0.8929 (0 outlier)	0.9286 (0 outlier)	0.8607 (2 outliers)
Three spoofed satellites	0.6667 (0 outlier)	0.7143 (0 outlier)	0.7619 (0 outlier)	0.8095 (0 outlier)	0.9048 (0 outlier)	0.7714 (0 outlier)

TABLE I. ACCURACY OF GMM IN SITUATIONS WITH 1,2,3 SPOOFED SATELLITES

Pseudo-range changes(m) / Situation	30	40	50	100	200	average
One spoofed satellite	0.9286 (7 outliers)	0.8810 (6 outliers)	0.9127 (5 outliers)	0.6270 (1 outlier)	0.7698 (0 outlier)	0.8238 (3.8 outliers)
Two spoofed satellites	0.8750 (9 outliers)	0.8750 (9 outliers)	0.8929 (4 outliers)	0.9286 (2 outliers)	0.9286 (0 outlier)	0.9000 (4.8 outliers)
Three spoofed satellites	0.8095 (6 outliers)	0.8095 (5 outliers)	0.8095 (4 outliers)	0.9048 (0 outlier)	0.8571 (0 outlier)	0.8381 (3 outliers)

TABLE II. ACCURACY OF K-MEANS IN SITUATIONS WITH 1,2,3 SPOOFED SATELLITES

measurements based on the GMM, which is a widely used unsupervised machine learning clustering algorithm. Several spoofing attack scenarios have been evaluated with different amount of spoofed satellites and various degrees of pseudo-range measurements manipulations. It has also been found out the GMM clustering algorithm could cluster the position points generated by clean signals properly with more than 90% accuracy on average when there is one spoofed satellite. We also validate the proposed algorithm by comparing with another clustering algorithm K-means. The performance of the K-means is worse with several problems, especially when the changes of pseudo-range measurements are no more than 200 meters with each satellite. References

[1] K. Zhang and P. Papadimitratos, "Secure Multi-Constellation GNSS Receivers with Clustering-Based Solution Separation Algorithm," 2019 IEEE Aerospace Conference, 2019, pp. 1-9.

[2] R.Y Zhang, C.K. Seow, K. Wen and H. Zhang, "Spoofing Attack of Drone," 2018 IEEE 4th International Conference on Computer and Communications (ICCC), 2018, pp. 1239-1246.

[3] M. L. Psiaki and T. E. Humphreys, "GNSS Spoofing and Detection," in Proceedings of the IEEE, vol. 104, no. 6, June 2016, pp. 1258-1270.

[4] D. Suo and S. E. Sarma, "Real-time Trust-Building Schemes for Mitigating Malicious Behaviors in Connected and Automated Vehicles," 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 1142-1149.

[5] T. Kim, M. Motro, P. Lavieri, S. S. Oza, J. Ghosh and C. Bhat, "Pedestrian Detection with Simplified Depth Prediction," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 2712-2717.

[6] A. Barrientos, A. Mora, I. Lafoz, R. San Martin and P. Munoz, "CAWAS: collision avoidance and warning system for automotives based on satellite," Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005., 2005, pp. 480-485.

[7] Ali Jafarnia-Jahromi, Ali Broumandan, John Nielsen, Gérard Lachapelle, "GPS Vulnerability to Spoofing Threats and a Review of Antispoofing Techniques", International Journal of Navigation and Observation, vol. 2012, ArticleID 127072, 16 pages, 2012.

[8] Y.H. Chu, S.L. Keoh, C.K. Seow, Q. Cao, K. Wen, S.Y. Tan, " GPS Signal Authentication Using a Chameleon Hash Keychain", International Conference on Critical Infrastructure Protection, pp. 209-226, 2021.

[9] J. R. v. d. Merwe, X. Zubizarreta, I. Lukčin, A. Rügamer and W. Felber, "Classification of Spoofing Attack Types," 2018 European Navigation Conference (ENC), 2018, pp. 91-99.

[10] Y. Sun and L. Fu, "A New Threat for Pseudorange-Based RAIM: Adversarial Attacks on GNSS Positioning," IEEE Access, vol. 7, 2019, pp. 126051-126058.

[11] S. Semanjski, A. Muls, I. Semanjski and W. De Wilde, "Use and Validation of Supervised Machine Learning Approach for Detection of GNSS Signal Spoofing," 2019 International Conference on Localization and GNSS (ICL-GNSS), 2019, pp. 1-6

[12] K. Zhang, R. A. Tuhin, and P. Papadimitratos, 'Detection and Exclusion RAIM Algorithm against Spoofing/Replaying Attacks', in International Symposium on GNSS, 2015.

[13] Pestana, António, Reading RINEX 2.11 Observation Data Files. 2015.

[14] S. W. Chen, C. K. Seow and S. Y. Tan, "Elliptical Lagrange-Based NLOS Tracking Localization Scheme," in IEEE Transactions on Wireless Communications, vol. 15, no. 5, pp. 3212-3225, May 2016.

[15] C.K. Seow and S.Y. Tan, "Localisation of mobile device in multipath environment using bi-directional estimation," Electronics Letters, vol. 44, No 7, pp. 485-487 , Mar 2008.

[16] S. Z. Khan, M. Mohsin and W. Iqbal, "On GPS spoofing of aerial platforms: a review of threats, challenges, methodologies, and future research directions," PeerJ. Computer Science, vol. 7, 2021, pp. e507-e507.

[17] Björck Å. "Least squares methods", Handbook of numerical analysis. 1988, pp. 465-652.

[18] Misra S, Li H, He J. "Robust geomechanical characterization by analyzing the performance of shallow-learning regression methods using unsupervised clustering methods". Machine Learning for Subsurface Characterization, 2020, pp. 129-55.

[19] Ezaki T, Himeno Y, Watanabe T, Masuda N. "Modelling state - transition dynamics in resting - state brain signals by the hidden Markov and Gaussian mixture models". European Journal of Neuroscience 54.4, 2021, pp. 5404-5416.

[20] Lücke, Jörg, and Dennis Forster. "k-means as a variational EM approximation of Gaussian mixture models." Pattern Recognition Letters 125, 2019, pp. 349-356.

[21] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12, 2011, pp. 2825-2830.

[22] Crustal Dynamics Data Information System (CDDIS DAAC),International GNSS Service, Daily 30-second observation data, Dec.2021. https://cddis.nasa.gov/archive/gnss/data/daily/2021/063