IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 25, 2020, accepted November 30, 2020, date of publication December 4, 2020,
date of current version December 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042469

On the Use of Differential Correction Clustering for Facing Spoofing Attacks to GNSS Augmentation Networks

SARA BALDONI[1], (Graduate Student Member, IEEE), FEDERICA BATTISTI[2], (Senior Member, IEEE), AND ALESSANDRO NERI[1,3], (Member, IEEE)

[1]Department of Engineering, Roma Tre University, 00146 Rome, Italy
[2]Department of Information Engineering, University of Padova, 35131 Padua, Italy
[3]Radiolabs, 00198 Rome, Italy

Corresponding author: Sara Baldoni (sara.baldoni@uniroma3.it)

This work has been partially developed under the grant of the "HELMET" project of the H2020-SPACE-EGNSS-2019 Program (project number 870257).

ABSTRACT Nowadays, Global Navigation Satellite Systems are the main source for high accuracy positioning and timing. For this reason, they are essential both for everyday activities and services, and for the industrial and critical infrastructure sectors. Moreover, the spread of increasingly autonomous vehicles results in strict accuracy and integrity requirements. This leads to the need for additional infrastructure to send corrections to the end users and mitigate the measurement errors, the Augmentation Networks. However, due to the increasing exploitation of localization functionalities, the Augmentation Networks could become a primary target for attackers resulting in a high financial and safety cost. Among the possible attacks, spoofing, that is the generation of a fake satellite signal which is seen as genuine by the receiver, is one of the most powerful and tricky. In this contribution, a detection and mitigation strategy for Augmentation Network spoofing attacks is proposed. We introduce two attack models and present a technique based on K-means clustering to counteract them. More in details, our approach is based on the computation of the number of clusters formed by the Augmentation Network corrections. Starting from the hypothesis that under nominal conditions only one cluster is present, the effects of the attacks on the clustering procedure are analyzed, and several attack simulations are performed to evaluate the algorithm performances. The proposed method has been compared both to an Augmentation Network attack detection technique, and to a receiver-level spoofing mitigation approach, showing comparable or better performances. Moreover, to the best of our knowledge, this is the first work addressing mitigation for spoofing attacks which target an Augmentation Network.

INDEX TERMS Augmentation networks, global navigation satellite system, GBAS, K-means clustering, SBAS, security, spoofing attack.

I. INTRODUCTION

The Global Navigation Satellite Systems (GNSSs) like Global Positioning System (GPS), Galileo, GLObal NAvigation Satellite System (GLONASS), and BeiDou Navigation Satellite System (BDS), constitute pervasive, enabling technologies for a large amount of today activities and services, including those related to critical infrastructures. Due to the cross-sector dependency on Position, Navigation and Timing (PNT) functionalities provided by GNSS, the research in this field is focusing on the fulfillment of the requirements concerning performance, with particular attention to accuracy and integrity, and on the resolution of GNSS security flaws.

Concerning the performances, stand-alone GNSS limitations have to be overcome. The basic idea of satellite positioning is to localize an object by exploiting the knowledge of its distance from a set of satellites. To this aim, the satellites are equipped with Radio Frequency (RF) transmitters that send to the ground a signal composed of ranging code and navigation data. The former allows to compute the time of flight of each signal which is exploited by the receiver to compute the distance, named pseudorange, from each satellite. The pseudorange, in fact, is obtained by multiplying the signal time of flight, given by the difference between the time of arrival and the nominal time of departure, by the velocity

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

of light in the vacuum. The satellite position can then be computed from the ephemerides contained in the navigation data. Since the satellite clock is not ideal and the RF signal propagates through the atmosphere, the navigation data include also the parameters for computing the deviation of the satellite clock from its nominal value and the incremental delays due to the propagation of the GNSS signals through the ionosphere and the troposphere, [1]. The main weakness of this approach for localization purposes is represented by the dependency on the navigation message broadcasted by the satellites. In fact, the corrections computed on the basis of the navigation message do not cover all the phenomena affecting the transmission, propagation and reception of the GNSS Signal in Space (SIS). In addition, the effective bandwidth of the navigation data channel, as well as the one of the satellite uplink channel, do not allow a timely correction of fast changes of satellite ephemerides, clock errors, and propagation delays. To fulfill the needs of those applications that require a higher accuracy than the one achievable by GNSS stand-alone mode, like those related to autonomous and connected transport systems, Augmentation Networks have been introduced. They exploit a set of ground Reference Stations (RSs), georeferenced during their deployment, to compute accurate corrections (e.g. Differential GNSS, Precise Point Positioning) and/or reference signals (Networked Real-Time Kinematic) to be sent to the end users to fix their measurements, either directly or through terrestrial or satellite broadcast channels, like the U.S.A. Wide Area Augmentation System (WAAS), and the European Geostationary Navigation Overlay Service (EGNOS). Satellite Based Augmentation Systems (SBASs) use RSs located in an entire continent to measure GNSS errors and transfer them to a central computing centre where differential corrections and integrity messages are computed. This information is then broadcasted through geostationary satellites to the final users. In this work, on the contrary, we focus on the use of Differential GNSS (DGNSS) implemented through ground RSs which provide the differential corrections via a terrestrial link.

As for the security issue, the US Department of Homeland Security has recently included PNT in the set of ''National Critical Functions'' [2]. Meanwhile, a UK study indicates that disruption of GNSS would cost 1 billion of £ per day, [3]. Unfortunately, the inherently low power of received GNSS signals (approx. -127 dBm on Earth) causes the receivers to be extremely susceptible to all types of unintentional and intentional interferences, including jamming, spoofing and meaconing. Whereas jammers are used for denial-of-service attacks, spoofers and meaconers pose an even bigger threat, because they can lead a receiver to estimate a forged position and/or time. In fact, spoofers transmit signals that appear as genuine, but with mutual delays such that the receiver will appear as located in any place decided by the attacker [4]. While receivers for military applications are usually designed for operating in hostile environments and fully exploit the features of the military signals addressing authentication issues, low cost receivers for civilian applications lack of robustness against such kind of attacks [5]. Moreover, in addition to spoofing targeting individual receivers, the current threat includes attacks directed toward the Augmentation Networks. Once the position accuracy is entrusted to Augmentation Networks, in fact, such systems become a perfect target.

Since such a kind of attack may produce large scale effects while staying undetected, in this article we address two research questions:

- R1) Is it possible to design a clustering-based method for detecting and mitigating spoofing attacks directed to the Augmentation Networks?
- R2) Is it possible to outperform existing approaches which aim at detecting spoofing against Augmentation Networks or at counteracting the attack at the user receiver level?

To answer the first research question (R1) we propose a technique based on the computation of the number of clusters formed by the corrections sent by the RSs. Each RS generates a correction vector that can be seen as a point of the correction space. The clusters formed by such points, then, are directly related to the RSs which computed them, thus allowing the identification of the attacked ones. The core of our method is the hypothesis that under nominal conditions the differential corrections should group in a single cluster, so that whenever two or more clusters are present an anomaly can be declared. Although in principle each clustering algorithm could be used, we apply the K-means clustering. We exploit the GAP statistic [6] to identify the optimal number of clusters, and propose and compare two re-labeling techniques to reduce the false alarms introduced by the GAP statistic. Details about the proposed technique are provided in the following, and its effectiveness is proven against two attack models using a simulation setup involving the measurements of twenty-one RSs of the IGS (International GNSS Service) network.

To give an answer to the second research question (R2), we compare the proposed method both with the detection strategy for Augmentation Network spoofing proposed in [7], and with the clustering-based user spoofing mitigation technique proposed in [8].

The remainder of the paper is organized as follows. In Section II the security aspects of positioning are presented. In Section III the analytical framework is introduced. In Section IV, the attack models are presented and in Section V the proposed detection and mitigation techniques are detailed. Then, in Section VI the experimental setup and the benchmark approaches are introduced. In Section VII the simulation results and the answers to the addressed research questions are provided and, finally, in Section VIII, the conclusions are drawn.

II. SECURITY ASPECTS OF POSITIONING

Due to the wide range of applications based on GNSS PNT functionalities, satellite-based positioning is becoming an attractive objective for attackers.

The feasibility of the spoofing attack is a well studied topic in the literature. A first implementation of a

portable software-defined spoofer was proposed in [9], where Humphreys *et al.* demonstrated their capability in realizing the attack, thus highlighting the need for effective countermeasures. The requirements for a successful attack have been analyzed also in [10]. More in details, the authors studied GPS spoofing both for single and multiple victims, and for civilian and military receivers. More recently, a portable low-cost spoofer for the road scenario has been implemented in [11]. The attack was tested in a real-world driving scenario, and its feasibility has been proven thanks to a simulation test with 40 participants. At last, in [12], we simulated a spoofing attack in a railway scenario. We showed how an attacker can spoof the SIS received by the RSs to produce a desired offset in the train position computed by the on-board receiver. It is important to highlight that a tampering on the RS SIS directly impacts on the differential corrections sent to the on-board receiver. Along with the possibility to implement radio spoofing attack another threat is arising: software attacks. An overview about this issue is provided in [5], where the authors propose several kinds of attack, including a software attack directed to a set of RSs. More recently, Luo *et al.* in [13] highlighted the security threats concerning the development of autonomous vehicles, and proposed a software spoofing attack targeting three mobile navigation apps. In addition, due to the spread of deep learning techniques, adversarial attacks on GNSS have been proposed in [14].

In order to counteract the spoofing threat, different techniques have been proposed in the literature and an extensive review of existing countermeasures can be found in [15]. For instance, outputs from a commercial receiver are combined to detect spoofing in [16]. More in details, the Automatic Gain Control (AGC) behavior is monitored along with a Signal Quality Metric (SQM). The two parameters are then combined through an AND operation to detect spoofing. The use of SQM monitoring for spoofing detection has been proposed also in [17], where multiple metrics are fused in order to overcome single metric weaknesses, showing a significant performance improvement. Other works propose to exploit subsequent measurements, such as [18] and [19]. The former paper proposes a spoofing detection technique which exploits the double differences between the pseudoranges computed in two time instants for a pair of satellites. In [19], the authors use the wavelet transform to mitigate the spoofing effect. Their approach is based on computing the pseudorange diversity, between two consecutive observations, and use it as input of the wavelet transform to identify the deviation due to the attack.

Another possibility is to exploit multi-sensor approaches for spoofing suppression. The combination of GNSS and Inertial Navigation Systems (INS) for spoofing detection is proposed for instance in [20]. The authors aim at exploiting the complementary nature of the involved technologies to perform a consistency check. The use of GPS-INS hybrid positioning is considered also in [21]. More specifically, Support Vector Machines (SVM) are used for GPS spoofing attack detection and the distribution of the error between the two systems is analyzed to identify the attack. Supervised machine learning has been exploited for spoofing detection showing promising results also in [22]. The authors performed a correlation analysis to identify the statistically significant variables and used them to train the SVM. Moreover, Shafiee *et al.* compared the performances of K-Nearest Neighbourhood and naive Bayesian classifiers with the ones of a neural network for spoofing detection in [23], showing that the neural network achieves the best results.

In this work, to answer to the research question number one (R1), we propose a clustering-based spoofing detection and mitigation algorithm for attacks which target the Augmentation Networks. There are two main differences between state-of-the-art methods and the one we are proposing:

1) the described techniques address attacks which target the end user receivers, whereas we are considering a spoofing implemented against the Augmentation Network. The reason why an attacker may choose to alter the differential corrections is that if the satellite signals received by a group of receivers are spoofed, all of them will appear in a unique position. If the receiver positions are monitored by a common service center, as in the case of car sharing and fleet management, or shared among them, as in case of cooperative driving, the attack becomes easily detectable [10]. When the spoofing attack is addressed to the Augmentation Network, on the contrary, the relative positions among the receivers will remain consistent and the attack detection based on position coincidence will be ineffective. Consequently, we decided to define a countermeasure to this kind of attack and we selected one of the very few works addressing the problem of spoofing directed to Augmentation Networks [7] to answer to research question number two (R2). More in details, the method proposed in [7] detects spoofing by defining a threshold for the magnitude of the difference between the positions computed with and without DGNSS corrections. Differently from [7], we aim at detecting anomalies in the differential correction domain instead of analyzing the computed positions.
2) Data-driven methods, such as [21]–[23], exploit supervised learning. In our case, on the contrary, an unsupervised technique based on K-means clustering is proposed. This is the reason why, to give an answer to research question number two (R2), we compared our method to the one presented in [8]. More specifically, the authors proposed a clustering-based solution separation algorithm to detect spoofing attacks against the end user receiver. Their algorithm checks the consistency of the positions obtained with different subsets of satellites since the satellites whose signals have not been altered should create a cluster in the position domain thus allowing their identification. Let us note that the method proposed in [8] exploits a multi-constellation approach whereas our method aims

at facing spoofing efficiently even with a single constellation.

III. NOTATIONS AND DEFINITIONS

For a quantitative analysis of the effects produced by an attack targeting the Augmentation Network, let us introduce the analytical framework. For sake of simplicity and compactness, here we refer to the case of PNT computed on the basis of code pseudoranges. Thus, given a receiver (RX), the measured code pseudorange of the i-th satellite in view for the k-th epoch is given by:

$$\begin{aligned}\rho_i(k) = &\|\mathbf{X}_i^{sat}[T_i(k)] - \mathbf{X}^{RX}\| + c\delta t^{RX}(k)\\ &-c\delta t_i^{sat}(k) + c\Delta\tau_i^{trop}(k) + c\Delta\tau_i^{ion}(k)\\ &+n_i(k), \quad i = 1, \ldots, N_{sat}\end{aligned} \tag{1}$$

where:

- $T_i(k)$ is the time instant on which the signal of the k-th epoch is transmitted from the i-th satellite;
- $\mathbf{X}_i^{sat}[T_i(k)]$ is the coordinate vector of the i-th satellite at $T_i(k)$;
- $\mathbf{X}^{RX}$ is the coordinate vector of the receiver;
- $\delta t^{RX}(k)$ and $\delta t_i^{sat}(k)$ are the RX and i-th satellite clock offset, respectively;
- $\Delta\tau_i^{trop}(k)$ and $\Delta\tau_i^{ion}(k)$ are the tropospheric and ionospheric incremental delays along the path from the i-th satellite to the receiver, respectively;
- $n_i(k)$ is the pseudorange error due to the time of arrival estimation algorithm generated by multipath, GNSS receiver thermal noise, radio frequency interference and other possible local effects. Let us note that the error on the time of arrival estimation determines an error in the computed distance if multiplied by the velocity of light, so that $n_i(k)$ is expressed in meters;
- N_{sat} is the number of visible satellites.

The satellites themselves provide, through the navigation message, the models' parameters required to approximate the satellite clock offset and the ionospheric and tropospheric incremental delays. Let $\Delta\hat{\tau}_i^{ion}(k)$, $\Delta\hat{\tau}_i^{trop}(k)$ and $\delta\hat{t}_i^{sat}(k)$ be the estimates of the ionospheric and tropospheric delays and of the satellite clock offset. Moreover, the term $\|\mathbf{X}_i^{sat}[T_i(k)] - \mathbf{X}^{RX}\|$ represents the geometrical distance $r_i(k)$ between satellite and receiver. However, the true satellite position is unknown and its estimate, $\hat{\mathbf{X}}_i^{sat}[T_i(k)]$, computed on the basis of the ephemerides contained in the navigation data, is employed. This implies that the corresponding geometric range $\hat{r}_i(k)$ differs from $r_i(k)$ by a quantity $\epsilon_i^{eph}(k)$ usually addressed as ephemeris pseudorange error. Equation 1 can be now written as:

$$\begin{aligned}\rho_i(k) = &\hat{r}_i(k) + c\delta t^{RX}(k) - c\delta\hat{t}_i^{sat}(k)\\ &+c\Delta\hat{\tau}_i^{trop}(k) + c\Delta\hat{\tau}_i^{ion}(k) + n_i(k)\\ &+c\epsilon_i^{ion}(k) + c\epsilon_i^{trop}(k) - c\epsilon_i^{\delta t}(k) + \epsilon_i^{eph}(k),\end{aligned} \tag{2}$$

where ϵ_i^{ion}, ϵ_i^{trop} and $\epsilon_i^{\delta t}$ are the model residual errors. These residuals can be grouped, together with $n_i(k)$, in a generalized error term $v_i(k)$. Equation 2 then becomes:

$$\begin{aligned}\rho_i(k) = &\hat{r}_i(k) + c\delta t^{RX}(k) - c\delta\hat{t}_i^{sat}(k)\\ &+c\Delta\hat{\tau}_i^{trop}(k) + c\Delta\hat{\tau}_i^{ion}(k) + v_i(k),\end{aligned} \tag{3}$$

where, apart from $v_i(k)$, the only unknowns are the receiver's position $\mathbf{X}^{RX}$ and clock offset $\delta t^{RX}(k)$. Equation 3 can be solved, if at least four pseudorange measurements are available, with the least square estimation method [1], [24].

The main goal of DGNSS is the compensation of the model residuals which contribute to $v_i(k)$, thanks to the exploitation of their spatial correlation. To this aim, the signals received by a set of RSs located in the user's area at known and fixed locations are exploited. Let us observe that satellite clock errors are the same for any receiver, independently from its location. On the other hand, the ephemeris error depends on the projection of the satellite position error onto the line of sight connecting the receiver to the satellite itself. Although this projection varies with the receiver location, its value can be well approximated by the one computed at a nearby RS. In the same way, even if the atmospheric features are not exactly the same at different locations, their value is similar enough for a user-RS pair [25].

To support differential positioning, each RS computes a correction vector. To this aim, for each satellite in view, the pseudorange, $\rho_{i,n}(k)$, measured by the n-th RS is corrected for the satellite clock offset, $c\delta\hat{t}_i^{sat}(k)$, and for the atmospheric incremental delays, $c\Delta\hat{\tau}_{i,n}^{trop}(k)$ and $c\Delta\hat{\tau}_{i,n}^{ion}(k)$. Then, the difference between the corrected pseudorange and the geometric range, $\hat{r}_{i,n}(k)$, evaluated on the basis of the known fixed RS position, is computed as:

$$\begin{aligned}\Delta\rho_{i,n}(k) = &\rho_{i,n}(k) - \hat{r}_{i,n}(k)\\ &+c\delta\hat{t}_i^{sat}(k) - c\Delta\hat{\tau}_{i,n}^{trop}(k) - c\Delta\hat{\tau}_{i,n}^{ion}(k).\end{aligned} \tag{4}$$

Thus, from Equations 1 and 2 we obtain:

$$\begin{aligned}\Delta\rho_{i,n}(k) = &\epsilon_i^{eph} - c\epsilon_{i,n}^{\delta t}(k) + c\epsilon_{i,n}^{ion}(k)\\ &+c\epsilon_{i,n}^{trop}(k) + c\delta t_n^{RS}(k) + n_{i,n}(k).\end{aligned} \tag{5}$$

Since $\Delta\rho_{i,n}(k)$ is affected by both the receiver clock offset and the receiver noise, the differential correction is computed by filtering out these two components. For sake of compactness, without loss of generality, in the following we consider the case of memory-less differential correction computation. Thus, denoting with $\mathbf{K}_{\Delta\hat{\rho}_n}(k)$ the gain of the filter, the correction vector of the n-th RS, $\Delta\hat{\boldsymbol{\rho}}_n(k)$, containing the differential corrections for all the satellites in view, can be written as

$$\Delta\hat{\boldsymbol{\rho}}_n(k) = \boldsymbol{K}_{\Delta\hat{\rho}_n}(k)\Delta\boldsymbol{\rho}_n(k). \tag{6}$$

Since model residual errors and observation noise are expected to be zero mean random variables, the easiest way to obtain an estimate, $\overline{c\delta t_n^{RS}}$, of the RS clock offset is to average $\Delta\rho_{i,n}(k)$ of Equation 5 with respect to the $N_{sat,n}$ satellites visible by the n-th RS:

$$\overline{c\delta t_n^{RS}} = \frac{1}{N_{sat,n}}\sum_{j=1}^{N_{sat,n}}\Delta\rho_{j,n}(k). \tag{7}$$

The i-th element of the vector $\Delta\hat{\boldsymbol{\rho}}_n(k)$ is then computed as

$$\Delta\hat{\rho}_{i,n}(k) = \Delta\rho_{i,n}(k) - c\overline{\delta t_n^{RS}}. \quad (8)$$

For a RS network, the corrections forwarded to the end user are usually computed as a weighted sum of the individual differential corrections $\left\{\Delta\hat{\rho}_{i,n}(k)\right\}$ provided by a subset $\mathbb{I}_{vRS}$ of the RS set $\mathbb{I}_{RS}$. This procedure mimics the presence of a virtual RS (vRS) which provides the following corrections:

$$\Delta\hat{\rho}_{i,vRS}(k) = \sum_{n\in\mathbb{I}_{vRS}} \lambda_{i,n}\Delta\hat{\rho}_{i,n}(k). \quad (9)$$

The rationale behind the definition of a virtual RS is that, if none of the real RSs is close enough to the end user receiver, the corrections that would be sent by such a RS have to be approximated. To this aim, the contributions given by the subset $\mathbb{I}_{vRS}$, usually composed of the RSs which are closer to the user, can be combined through an appropriate interpolation technique, such as Kriging [26].

The major security issue of this scenario is that, if the RSs have been attacked, the corrections will be altered and the end user will not be able to compute his real position and time. These aspects will be detailed in the following sections.

IV. ATTACK MODELS

To define the adversary strategy we introduced two different kinds of attack: coherent and non-coherent. The coherent scenario occurs when the attacker wants to cause a specific displacement in the position estimated by the targeted receiver, whereas the non-coherent attack is designed to introduce significant errors in the PNT estimation process. Since our detection and mitigation strategy works on the differential corrections extracted from the pseudoranges, the attack models will be presented showing their impact on the pseudoranges. Details about the signal model before and after the attack are provided in Appendix A. In the following we will denote with superscripts S and H the quantities referred to spoofing and normal condition, respectively. For quantities and relations valid in both cases subscript will be dropout. Moreover, let us indicate as $\mathbb{I}_{V,n}$ the set of visible satellites for the n-th RS. Under spoofing attack, the pseudoranges $\rho_{i,n}(k)$ of the attacked RSs are modified by terms $b_{i,n}(k)$ so that:

$$\Delta\rho_{i,n}^{S}(k) = \Delta\rho_{i,n}^{H}(k) + b_{i,n}(k), \quad \forall n \in \mathbb{I}_{RS}^{S} \wedge i \in \mathbb{I}_{V,n}, \quad (10)$$

where $b_{i,n}(k)$ will be 0 if the i-th satellite has not been attacked. The alteration introduced in $\Delta\rho_{i,n}$ directly affects the final differential correction computed through Equation 9. Let us now detail how the bias $b_{i,n}(k)$ in Equation 10 can be determined for the two attack models.

Concerning the coherent attack, the spoofer has to transmit a set of satellite signals, appearing as the true ones, such that the corresponding pseudoranges equal the geometric distances $\{r_i^{spoofed}(k)\}$ between the set of attacked satellites and the location where it wants the receiver to appear. Incidentally we recall that the satellite positions are known to all receivers based on the ephemerides broadcasted by the satellites. On the other hand, when the spoofer knows the receiver location, it can compute the current geometric distance $r_i^{real}(k)$ between the i-th satellite and the target receiver. Thus, to spoof the receiver position, the attacker may choose to alter the final differential corrections, $\Delta\hat{\rho}_{i,vRS}(k)$, by the quantity $\Delta r_i(k)$ corresponding to the difference between the true and the spoofed geometric distances:

$$\Delta r_i(k) = r_i^{real}(k) - r_i^{spoofed}(k). \quad (11)$$

In fact, the same offset will affect the measurements of the end user after applying the differential corrections. Given the differential correction increments $\Delta r_i(k)$, the attacker can determine the spoofing offsets $b_{i,n}(k)$ to be applied to the pseudoranges of the n-th RS contributing to the computation of the virtual RS differential corrections. For instance, when planning to attack a subset $\mathbb{I}_{RS}^{S} \subseteq \mathbb{I}_{vRS}$ of RSs, the easiest choice is to apply the same bias to each of them neglecting the contribution of the RS clock offset. In this case, then, the bias $b_{i,n}(k)$ is independent from the n-th RS, so that the subscript n will be omitted in the following. Therefore, based on Equation 9, the attacker can compute for the i-th satellite

$$b_i(k) = \frac{\Delta r_i(k)}{\sum_{m\in\mathbb{I}_{RS}^{S}\cap\mathbb{I}_{vRS}} \lambda_{i,m}}. \quad (12)$$

The feasibility of such an attack, however, depends on the capability of knowing the target position in real-time. Let us consider three different targets: a train, a public transport system, and a car. We analyzed the first scenario in [12] where the attacker was supposed to know the position of the train. Since the rail coordinates can be obtained and the train velocity can be inferred depending on the train type, the assumption that the attacker knows the target position is plausible. Concerning the public transport system, several scenarios are possible. First of all, as for the train, an approximate path can be known. Moreover, thanks to the huge number of mobile apps which provide up-to-date arrival information for buses, we can assume that the attacker is able to obtain the target position. In addition to that, an attacker's partner on-board could relay the position provided by a smartphone GNSS receiver, or he/she can leave on-board a position relay unit. All these assumptions, however, do not hold for the car scenario which becomes too challenging if the attacker targets a specific vehicle. Nevertheless, we observe that the attacker may be interested in causing car crashes or traffic chaos in a specific road area instead of misleading a single car. For this reason, he/she may choose to alter the differential corrections using the coordinates of the road point where the attack will be performed, so that a victim car passing by that point will appear in the fake position. Let us note that, as shown by Equations 11 and 12, the alteration computed by the attacker depends on $r_i^{real}(k)$ and $r_i^{spoofed}(k)$. Consequently, being the satellite-receiver distance very large, the biases computed for a set of points in the attack area will be very similar. The result will be that a vehicle entering the attack area will appear in a fake position as distant as desired by the attacker from the true one.

In the non-coherent scenario, the attacker does not aim at causing a specific position displacement, but at altering the PNT estimation process. As a consequence, the complete knowledge about the target location is not needed. The attacker, in fact, can simply introduce signal delays corresponding to predefined pseudorange alterations for all the attacked satellites. In this case, the bias introduced in the pseudorage measurements is independent both from the n-th RS and from the i-th satellite, and equals a constant value $b(k)$. The same kind of alteration has been considered in [8]. This kind of attack is easier to realize than the coherent one, but it is equally dangerous.

Moreover, irrespective of the attack type, another parameter noticeably affects the attack complexity: the number of attacked RSs. Concerning the coherent scenario, an attack involving a large number of RSs is challenging due to the need of synchronization between the different spoofers and the victim, whose position has to be followed in real-time. However, as shown by Equation 12, the bias can be modified according to the number of targeted RSs, thus making the attack effect in the position domain almost independent from the amount of attacked stations. As a consequence, the attacker may choose to alter a small number of RS signals, while obtaining the same position shift. On the other hand, according to Equation 12, this implies that the bias injected in the differential correction will grow, thus allowing an easier attack detection in the differential correction domain, so that the attacker has to find a trade-off between the attack complexity and its stealthiness. However, let us note that attack detection becomes easier when the number of attacked RSs is small only if the detection strategy works in the differential correction domain. If, on the contrary, the spoofing countermeasure exploits the position information, as in [7], the attack stealthiness does not change with the number of attacked RSs. Concerning the non-coherent case, the bias injected in the RS measurements is independent from the number of attacked stations. As a consequence, the alteration of Equation 9 grows, thus making the attack more disruptive, when the number of attacked RSs increases. For the non-coherent scenario, however, the attack is independent from the target position, and no synchronization is needed between the spoofed RSs. As a consequence, the number of attacked stations may change in time causing a larger or smaller alteration in the PNT estimation process.

As previously mentioned, we focus on DGNSS implemented through ground RSs which provide the differential corrections for code pseudoranges via a terrestrial link. Nevertheless, let us note that every augmentation system (either ground or satellite-based) may show similar vulnerabilities.

V. PROPOSED METHOD

The core idea of the proposed method is to exploit the differential correction spatial correlation properties. Let N_{sat} be the number of satellites visible by all the RSs, then each RS differential correction vector $\Delta\hat{\boldsymbol{\rho}}_n(k)$ can be seen as a point of a N_{sat}-dimensional space. Since all the RS points represent the model deviations mainly due to the atmospheric behaviour, we expect these points to be close to each other. Without attacks, then, the differential corrections should form a dense cluster in the N_{sat}-dimensional space. As a consequence, when more than one cluster is identified an anomaly can be declared. The way the clusters behave because of a coherent and a non-coherent attack will be discussed in the following. Here, let us first formalize the adversary-Augmentation Network interaction. To do so, we adapted to the GNSS scenario the state machine security model proposed in [27], where clock synchronization for industrial applications has been addressed. More in details, in [27], three strategies have been defined to handle synchronization packets: "pass", "drop", and "quarantine". This strategy allows a system to determine if it is under attack and, if so, to employ mitigation techniques before it breaches clock synchronization. The overall system adapted to our scenario can be depicted as the state machine shown in Figure 1.

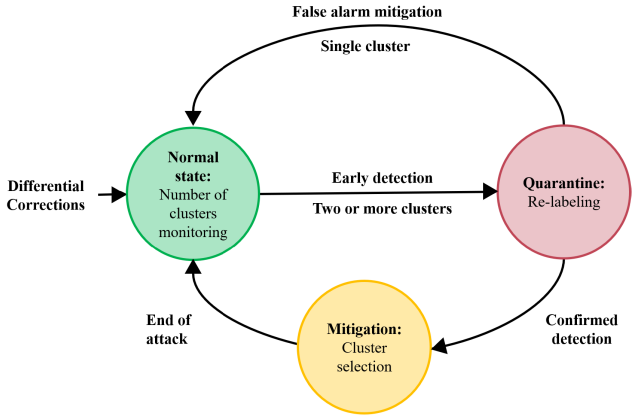


FIGURE 1. State machine model.

While in [27] the authors used rules, based on thresholds, to switch from one state to the other, in our approach state transitions are driven by a clustering procedure.

In the *normal state* the detector continuously monitors the differential correction clusters. If only one cluster exists, the system remains in this state. On the contrary, when a larger number of clusters is detected, the system moves to the *quarantine state*. This state can be seen as an "early warning stage" in which a further analysis is performed in order to confirm or cancel the alarm. This stage has been introduced in order to reduce the false alarm probability and consists in merging clusters too close to each other to indicate the presence of an attack. When the attack is confirmed, the system moves to the *detected attack state*. In this phase the mitigation strategy, aimed at identifying the subset of healthy RSs, is deployed. Therefore, the clusters are classified either as "reliable" or "unreliable". Even in the *detected attack state* the system performs the monitoring on all the stations and, once the multi-cluster condition ends, the system goes back to the *normal state*. Since the monitoring procedure has been defined through a state machine model, the rules used

for the transition from one state to the other are of utmost importance as will be detailed in Subsections V-A and V-B.

A. DETECTION

In order to detect the attack, the number of clusters has to be determined. Well-known validity indices to perform this task are, for instance, the Davies-Bouldin Index [28], the Calinski-Harabasz Index [29] and the Silhouette Index [30]. All these indices, however, are based on the computation of both inter-cluster and intra-cluster measures. For this reason, they cannot be applied to our single-cluster hypothesis verification. As a consequence, we resorted to a metric based only on intra-cluster features: the GAP statistic [6].

Given a set of points which can be grouped in a variable number K of clusters, the GAP statistic was introduced as a statistical procedure for the determination of an elbow for the within-cluster dispersion W_K, defined as in [6]. If K increases, in fact, W_K naturally decreases until the decrease flattens, creating an elbow. The authors in [6] proposed to compare the $log(W_K)$ with its expected value under a predefined null hypothesis. The value of K is selected as the one for which $log(W_K)$ falls the farthest below the reference curve, namely the one for which the GAP value is maximum:

$$GAP(K) = E\{log(W_K)\} - log(W_K), \tag{13}$$

where $E\{\}$ denotes expectation. The null hypothesis is defined as the case of a single cluster. Moreover, the reference distribution has to be defined and there are two possibilities:

- generate each reference feature uniformly over the range of values observed for it;
- generate the reference features from a uniform distribution over a box aligned to the data principal components.

The second model considers the shape of the data distribution and makes the procedure rotationally invariant. For this reason, we decided to use the second model for the reference distribution. Even if the GAP statistic alone allows to achieve good results, as will be shown in Section VII, the method based only on this statistic can be improved, as described in Subsection V-B.

B. QUARANTINE AND MITIGATION

A potentially anomalous behaviour is declared whenever the differential corrections group into two or more clusters. In this case the system state switches from *normal* to *quarantine*. In this state a re-labeling procedure is deployed to reduce false alarms. An example of clustering re-labeling is the one presented in [31] where security issues for Supervisory Control And Data Acquisition (SCADA) systems are addressed. More in details, in [31], the re-labeling process is realized by computing the Euclidean distance from each critical state to each normal cluster centroid. For SCADA, in fact, normal data should cluster in a set of dense clusters, while altered data should behave like outliers. In our case, on the contrary, normal data should create a single cluster, whereas after an attack, a multi-cluster scenario will occur. As a consequence, our re-labeling process consists in merging the clusters erroneously separated by the GAP statistic. To do so, we explored two techniques: a time consistency check and a standard deviation (STD) criterion.

As for the former method, to reduce latency in anomaly detection, we limited the length of the consistency check interval to three epochs. More in details, the multi-cluster scenario is confirmed, thus leading to the *detected attack state*, whenever at least two out of three clustering outputs agree. Otherwise, the current scenario is re-labeled as *normal* and the system goes back to the *normal state*. Moreover, a similar technique can be used in the *detected attack state* to reduce missed anomalies: the absence of an anomaly has to be confirmed by at least two out of three epochs to go back to the *normal state*. Although the time consistency check is an effective solution, it introduces a significant vulnerability in the system since, once the attacker is aware of it, he/she could create a non-persistent attack. The effect of such an attack on the end user position would be the one of a non-coherent attack which makes the receiver position continuously hopping between true and fake location. To reduce this vulnerability, we resorted to the STD-based re-labeling. Given the assumption that, under nominal conditions, the differential corrections form a dense cluster and follow a Gaussian distribution, a threshold based on the dispersion of such a cluster can be inferred. More specifically, we select a threshold (Γ) proportional to the standard deviation of the differential corrections under nominal conditions. During the monitoring, when the multi-cluster scenario occurs, the centroids of all the clusters are computed and the distances between all the centroid pairs are evaluated. Then, the minimum distance is computed and it is compared to the threshold Γ. If the minimum distance is smaller than the threshold, the corresponding clusters are merged and the process is repeated until when the minimum distance is larger than Γ. This technique is time independent so that it does not contribute to the described vulnerability and its success does not depend on how the detection algorithm behaves in other epochs. The procedure is summarized in Algorithm 1 and details about the threshold computation are provided in Appendix B.

When the multi-cluster scenario is confirmed, the mitigation strategy is deployed. As previously mentioned, the corrections are essentially the model error residuals and, for this reason, their value should be small. The goal of the attack can be described as the displacement of the global centroid from the value it had before the attack. The distance of the centroid of the attacked cluster from the N_{sat}-dimensional space origin, as a consequence, should be greater than the one of the healthy cluster. Therefore, a simple mitigation technique consists in selecting the cluster whose centroid is closer to the origin. If the attacker aims at causing a big position shift, the mitigation technique will correctly select the healthy cluster. In order to make the algorithm select the attacked cluster, in fact, the attacker should modify the corrections so that the attacked cluster centroid moves towards the origin. This implies that the mean differential correction would tend

Algorithm 1 State Machine Model With the STD-Based Re-Labeling

Compute the number of clusters, *Kopt*, based on the GAP statistic.
if *Kopt* ≥ 2 **then**
  Go to *quarantine state*.
  while *Kopt* > 1 **do**
    for $i \leftarrow 1$ to Number of cluster pairs **do**
      Extract the differential corrections belonging to the cluster pair
      Compute the cluster centroids
      Compute the distances between the centroids building a vector of distances *dist*
    end for
    Set $dist_{min}$ to the minimum of the vector *dist*
    if $dist_{min} < \Gamma$ **then**
      Merge the two clusters;
      set $Kopt = Kopt - 1$.
    else
      Go to *detected attack state.*
    end if
  end while
  Go to *normal state.*
else
  Stay in *normal state.*
end if

to zero leading back to the absolute positioning and canceling the attack effect. During the mitigation phase, a subset of the RSs is used to compute the final differential corrections. In the meanwhile the monitoring continues and, when only a single cluster is detected, the system goes back to the *normal state*.

C. ATTACK EFFECT ON THE CLUSTERING PROCEDURE

The effect of the attack on the clustering procedure mainly depends on how the differential corrections are computed. Due to the expression in Equation 8, a key role is played by the δt_n^{RS} estimate which under attack is given by:

$$\begin{aligned} \overline{c\delta t_n^{RS,S}} &= \frac{1}{N_{sat,n}} \sum_{j=1}^{N_{sat,n}} [\Delta\rho_{j,n}^{H}(k) + b_{j,n}(k)] \\ &= \overline{c\delta t_n^{RS,H}} + \frac{1}{N_{sat,n}} \sum_{j=1}^{N_{sat,n}} b_{j,n}(k). \end{aligned} \tag{14}$$

From Equations 8, 10, and 14, for the differential corrections the following relation holds:

$$\begin{aligned} \Delta\hat{\rho}_{i,n}^{S}(k) &= \Delta\hat{\rho}_{i,n}^{H}(k) + b_{i,n}(k) \\ &\quad - \frac{1}{N_{sat,n}} \sum_{j=1}^{N_{sat,n}} b_{j,n}(k), \quad \forall\, n \in \mathbb{I}_{RS}^{S} \wedge i \in \mathbb{I}_{V,n}. \end{aligned} \tag{15}$$

As a consequence, the final corrections for the i-th satellite provided to the user by the virtual reference station will be:

$$\begin{aligned} \Delta\hat{\rho}_{vRS,i}^{S}(k) &= \Delta\hat{\rho}_{vRS,i}^{H}(k) \\ &\quad + \sum_{n \in \mathbb{I}_{RS}^{S}} \lambda_{i,n} \left[b_{i,n}(k) - \frac{1}{N_{sat,n}} \sum_{j=1}^{N_{sat,n}} b_{j,n}(k) \right]. \end{aligned} \tag{16}$$

Equations 15 and 16 can be used to infer how the coherent and non-coherent attacks will affect the clustering procedure.

More in details, since in the coherent case the bias is equal for all the RSs, and $b_i(k)$ is given by Equation 12, we have that the differential corrections of the spoofed RSs are altered by the quantities

$$\begin{aligned} \Delta\hat{\rho}_{i,n}^{S}(k) - \Delta\hat{\rho}_{i,n}^{H}(k) = b_i(k) - \frac{1}{N_{sat,n}} \sum_{j \in \mathbb{I}_{V,n}} b_j(k), \\ \forall\, n \in \mathbb{I}_{RS}^{S} \wedge i \in \mathbb{I}_{V,n}. \end{aligned} \tag{17}$$

Since the visible satellites are not always the same for all the stations, the second term may vary from RS to RS. However, only slight variations are expected since, in order to make the target appear in the spoofed position, the biases $b_j(k)$ for different satellites will have different signs. When the sum is computed, then, they will compensate each other thus making the influence of the term $\frac{1}{N_{sat,n}}$ negligible. Moreover, a smarter spoofer may account for the lack of visibility of some satellites from the attacked RSs, and compensate for it. Therefore, we expect the differential corrections to be split mostly into two clusters after a coherent attack.

Concerning the non-coherent attack, setting the bias to a constant value $b(k)$ for all the RSs and satellites, implies that the differential corrections of the spoofed RSs are altered by the quantities

$$\begin{aligned} \Delta\hat{\rho}_{i,n}^{S}(k) - \Delta\hat{\rho}_{i,n}^{H}(k) &= b(k) - \frac{1}{N_{sat,n}} \sum_{j \in \mathbb{I}_{V,n}} b(k) \\ &= b(k) \left[1 - \frac{N_{att,n}}{N_{sat,n}} \right], \end{aligned} \tag{18}$$

where $N_{att,n}$ is the number of spoofed satellites visible by the n-th RS. From Equation 18 it is clear that, since the injected bias is equal for all the satellites, the alteration experienced by the differential corrections varies for each station and depends on the ratio $\frac{N_{att,n}}{N_{sat,n}}$. The main consequence is that a variable number of clusters may arise for every epoch depending on the number of satellites seen by each RS. In this case, therefore, instead of having a two-cluster partition, a multi-cluster scenario may occur.

The experimental proof of the effects of the attack on the clustering process, along with the behaviour of the detection and mitigation strategy with and without attack, are presented in Sections VI and VII.

VI. SIMULATION SETUP

For the experimental tests the data recorded by 21 stations in Europe have been downloaded from the NASA's space geodesy data center [32]. Among them, $N_{vRS} = 20$ have been used for the computation of the vRS corrections, and a central Europe RS has been employed as target user receiver. In fact, using a receiver in a known and fixed position as target allowed to better analyze the attack effects. A number of 1000 epochs (with a 30 second interval between them) have been extracted from a one day recording (01/01/2019) to perform the MATLAB simulations. The pseudoranges recorded by the RSs are stored in the RINEX files available at [32]. To verify that the detected anomaly was caused by the attack, and not due to irregularities in the data, we employed the same original pseudoranges to show the algorithm performances with and without the attack. To assess the method performance in absence of attack, we computed the differential corrections without altering the pseudoranges and we verified that the proposed method was able to recognize the presence of a single cluster. To evaluate the method performance when the spoofing was simulated, we altered the pseudoranges of the analyzed 1000 epochs and computed the corresponding differential corrections. We provided them as inputs to the proposed algorithm and verified that it was able to identify the presence of more than one cluster. We considered the attack detected if more than one cluster was identified. For mitigation, on the contrary, we considered the attack properly mitigated only if the cluster chosen to compute the final differential correction vector included all and only the healthy RSs.

The RSs selected for the simulation are shown in Figure 2 in blue, and the attack target is highlighted in red.

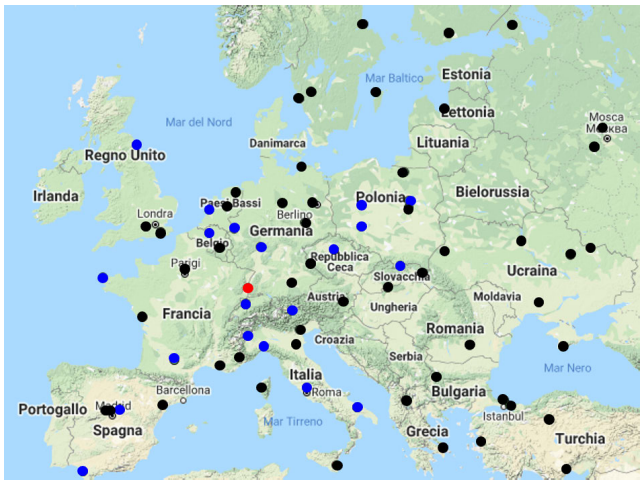


FIGURE 2. Reference Station map, as provided by [33].

Since the number of attacked RSs strictly depends on the capabilities of the attacker, the cost he/she can sustain and the effectiveness of the attack he/she wants to achieve, we analyzed a wide range of scenarios in which a variable number of RSs was targeted. More specifically, for each scenario, a different percentage of RSs has been involved in the attack, and the following set of percentages has been considered: {5%, 10%, 40%, 50%, 60%, 90%, 95%}. For 20 stations this implies a minimum of one attacked RS and a maximum of 19 attacked RSs. For each percentage we randomly selected the sets of healthy and attacked RSs. Let us note that an attack involving all the available RSs would not be detectable because a single cluster would appear in the differential correction space. In addition, we considered as candidate number of clusters, K, the values from 1 to 5. Moreover, to compute the threshold for the STD-based re-labeling, we downloaded the data for a different day (16/12/2018) and we considered 2000 epochs. First of all, we tested the Gaussian distribution hypothesis with the Kolmogorov-Smirnov test. To do so, we extracted the differential corrections for all the satellites in view from all the RSs. From the two thousand analyzed samples, we had a total of 12925 differential corrections (for each time sample, the number of satellites visible from all the RSs was different). The Gaussian distribution hypothesis was rejected only 17 times, so that the assumptions made in Appendix B can be considered valid. We show an example of the obtained normal probability plot in Figure 3 (note that if the sample data has a normal distribution, the data points appear along the reference line of the plot).

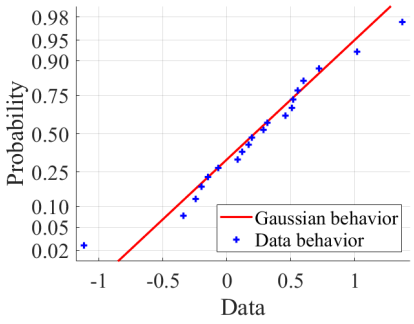


FIGURE 3. Normal probability plot example.

Starting from these data, we computed the standard deviation as detailed in Appendix B. Moreover, for the experimental tests, we set $\Gamma = 2\gamma_c$ with γ_c equal to three times the estimated standard deviation. Knowing that the 99.73% of samples extracted from a Gaussian distribution fall within $\pm 3\sigma$, we assume that the threshold Γ is representative of the minimum distance between the centroids of the two clusters needed to consider them as separate entities.

For the simulations we considered the GPS signal only, but the attack and the results can be extended to other GNSSs as well.

Concerning the coherent attack, four different position shifts have been considered: 1, 10, 50 and 100 meters. Both forward (North) and lateral (East) shift directions have been used. As for the set of altered satellite signals, we considered an attacker which, thanks to its prior knowledge on the attack area, spoofs the satellites potentially visible both from the RSs used in the differential correction computation and from the user position, without accounting for masking or shadowing. In this way, the attacker is sure that the satellites

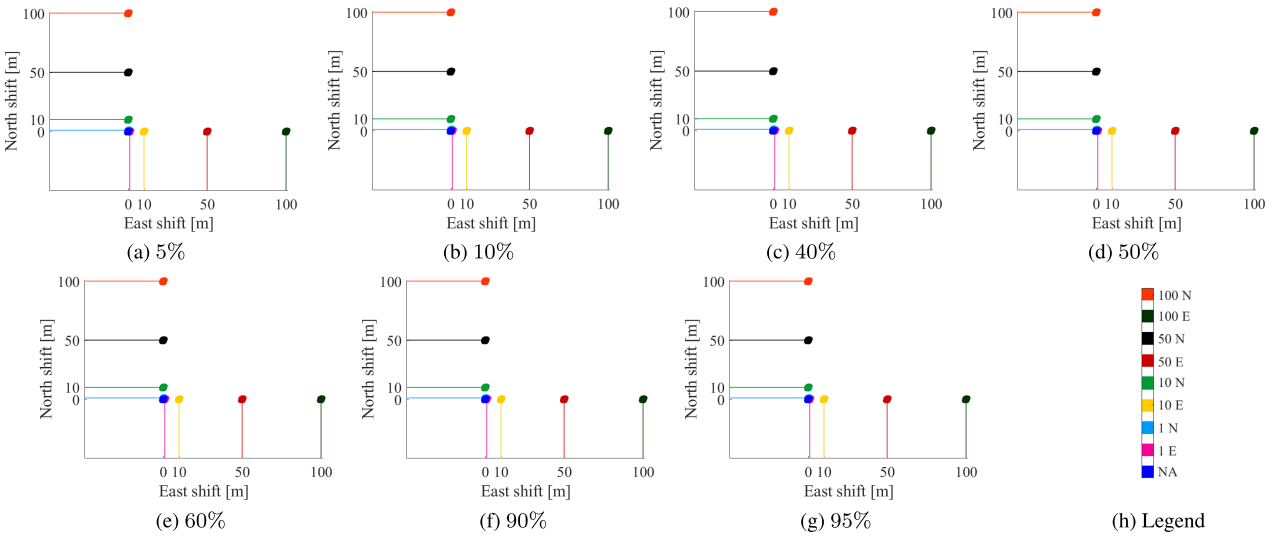


FIGURE 4. Coherent attack effect on the final user position. Different colors represent different kinds of attack both for North (N) and East (E) directions. The blue plot represents the case in which there is no attack (NA).

for which corrections are made available have been altered. For the selected site, the number of spoofed satellites varied between 5 and 10. Concerning the spoofed differential corrections, for sake of simplicity, we simulated uniform weighting of the corrections provided by the N_{vRS} reference stations used in the vRS correction computation. Therefore, based on Equation 12, we have that:

$$b_i(k) = \Delta r_i(k) \frac{N_{vRS}}{N^S_{vRS}}, \tag{19}$$

where N^S_{vRS} denotes the number of spoofed RSs contributing to the vRS corrections. The effect of the coherent attack on the final user is shown in Figure 4, where the shifts between true and computed positions under attack are depicted. As clearly shown, the attacker is able to cause the desired displacement independently from the percentage of attacked RSs.

As for the non-coherent attack, we used a fixed alteration $b(k)$ for all the attacked RSs and satellites. In order to perform a fair comparison with the approach proposed in [8], we used the same alteration of the pseudorange measurements: 50, 100, 200, 300 and 500 meters. Concerning the number of attacked satellites we aim at showing that our technique is almost independent from this choice. For this reason we considered two opposite situations: the case of the alteration of a single satellite measurement, and the case in which all the satellites visible by all the RSs have been attacked. When all the shared satellites are altered, so that $N_{att,n}$ increases, the differential correction alteration tends to zero, as clearly shown from Equation 18. As a consequence, this is the attack with the smallest impact on the differential corrections and, in turn, on the final receiver position. This attack, as a consequence, has been introduced with the aim of comparing the methods with and without re-labeling under the most challenging circumstances more than to show a realistic attack scenario. On the other hand, the attack which targets a single satellite is the easiest to realize, and thus the most realistic, since a single measurement has to be altered for each RS. This kind of attack, in addition, has a relevant impact on the position accuracy of the final receiver, as shown in Figure 5.

Concerning the method performance assessment, we computed the accuracy defined as:

$$A = \frac{tp + tn}{tp + tn + fp + fn} = \frac{tp + tn}{N_{sample}}, \tag{20}$$

where tp are the true positives, tn are the true negatives, fp are the false positives, fn are the false negatives, and N_{sample} is the number of samples. Since we identify as true positives the cases in which the attack is present, and as true negatives the ones in which there is no attack, in absence of attack true positives do not exist. Thus, we computed the number of true negatives and divided it by the total number of samples (1000). We presented these results as percentages. A similar procedure was repeated when the attack was present. In this case, since we altered the pseudoranges for all the epochs, the true negatives did not exist. Let us note that having considered two separate testing scenarios (i.e. presence and absence of attack) does not impair the performance assessment. In fact, we considered the case of memory-less differential correction computation and the weakness of time-consistency re-labeling with respect to non-persistent attack in order to select the best mitigation strategy.

In the following subsection we provide some details about the methods used as baseline to answer to the research question number two (R2).

A. BENCHMARK APPROACHES

As stated in Section II, we compared our method to two state-of-the-art approaches to answer to the research question number two (R2).

First of all, we considered the approach proposed in [7], where the authors address the issue of spoofing against an Augmentation Network. More specifically, the attack

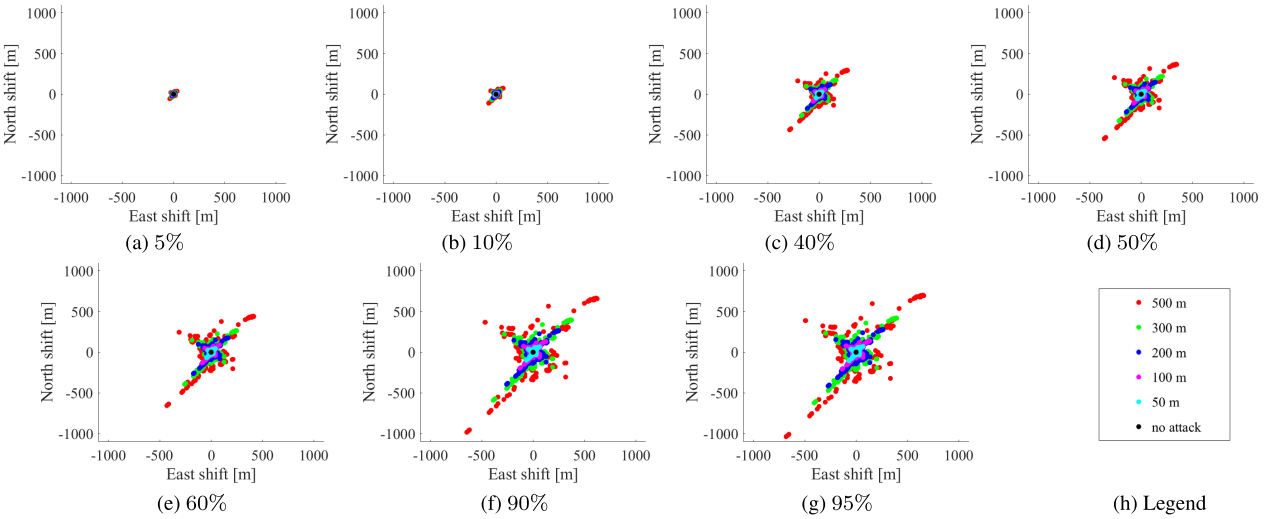


FIGURE 5. Non-coherent attack effect on the final user position when one satellite has been attacked. Different colors represent the different pseudorange alterations. The black plot represents the case in which there is no attack.

is detected by comparing the magnitude of the difference between the positions computed with and without the aid of the differential corrections to a threshold. However, they focus on attack detection only, so that when an attack is declared, no mitigation strategy is deployed. Although in [7] the criterion for the threshold selection has not been indicated, and a performance assessment has not been carried out, we can observe that an available choice for threshold setting could be the Neyman-Pearson criterion. Considering for sake of simplicity 2D localization, as in case of road vehicles, and assuming that, in nominal conditions, the North and East components of the difference $\Delta\mathbf{P}$ between DGNSS and stand-alone estimated positions can be overbounded by zero mean, uncorrelated Gaussian random variables with variance σ_d^2, we have that the magnitude ζ of $\Delta\mathbf{P}$ has a Rayleigh probability density function:

$$p_\zeta(\zeta) = \frac{\zeta}{\sigma_d^2} e^{-\frac{\zeta^2}{2\sigma_d^2}} \quad (\zeta \geq 0), \tag{21}$$

and the Probability of False Alarm P_{fa} is then given by

$$P_{fa} = \int_{\gamma_D}^{\infty} \frac{\zeta}{\sigma_d^2} e^{-\frac{\zeta^2}{2\sigma_d^2}} d\zeta = exp\left(-\frac{\gamma_d^2}{2\sigma_d^2}\right). \tag{22}$$

Thus, for a given P_{fa} the threshold is given by

$$\gamma_d = -\sqrt{2}\sigma_d \ln P_{fa}. \tag{23}$$

Considering that for $P_{fa} = 10^{-5}$ we have that $-\sqrt{2}\ln P_{fa} = 16.28$, and that σ_d^2 is essentially dominated by the variance of the stand-alone position error so that $\sigma_d > 1$ meter, a threshold γ_d fairly exceeding 16 meters can be expected. This result will be compared to our performances in the next section.

Moreover, in order to fully answer the second research question (R2), we compared our method to the one presented in [8]. In their paper, the authors propose a clustering-based spoofing mitigation for an attack directed to a single receiver. More specifically, they propose exploiting multi-constellation signals to counteract spoofing, improving the results of a single constellation approach. As in [7], the analysis is performed in the position domain. The authors downloaded the RINEX files from the same source we used, but they analyzed one hour length data with 30 seconds interval, which corresponds to 120 epochs. Moreover, as we did, they considered as target of the attack one static station and implemented the attack through MATLAB simulation. In addition, as previously mentioned, we decided to use the same alteration of the pseudorange measurements to allow a proper comparison of the two methods. To assess the performances of the algorithm proposed in [8] when applied to the Augmentation Network scenario, in the following we will assume that each RS implements the method proposed by Zhang *et al*. In addition, futher considerations are needed. First of all, the method proposed in [8] is designed for a multi-constellation receiver. As stated in the paper, the single constellation version of the algorithm works only if the number of attacked satellites fulfills the inequality: $N_{att} \leq N - 5$. We checked, for each epoch, if the number of visible GPS satellites, when all the shared satellites had been attacked, was large enough to fulfill the requested inequality. The number of epochs for which the algorithm is suitable is 0 for 15 RSs and ranges from 5 to 20 for the remaining ones. Then, we can conclude that the single constellation version of the algorithm proposed in [8] is not suitable to mitigate a spoofing attack which targets all the shared satellites. Moreover, as stated in the paper, even when the single-constellation method can be applied, the multi-constellation one achieves better results thus leading to a more challenging comparison. Let us highlight that also the multi-constellation method presents limitations in terms of the maximum number of tolerable faults. An algorithm residing in the end user receiver, in fact, can only exploit position domain clustering and, since four

satellites are needed to obtain the position solution, a threshold for the maximum number of alterations exists. Exploiting multiple constellations this issue is reduced due to the increase in satellite availability, but it cannot be completely removed. On the contrary, working in the correction domain, our algorithm is independent from the number of attacked satellites, thus achieving good performances even with a single constellation. This gives a first answer to the second research question. Concerning the attacks performed in [8], the authors analyzed two different scenarios, namely the case in which a single constellation has been attacked, and the one in which both of them have been spoofed. In the first scenario the RS receives both Galileo and GPS signals, but only GPS satellite pseudoranges are attacked. Zhang *et al.* identified the altered signals by using the healthy constellation as reference and stated that it is not possible to achieve a false detection probability (P_{FA}) of 0.01% and a detection probability (P_D) of 99.9% for errors smaller than 50 meters. As a consequence, we considered these performances for all the analyzed alterations. Since in [8] the attack detection implies the identification and exclusion of faulty signals, and thus mitigation, to obtain the success probability when each RS implements the algorithm independently we computed the probability of mitigation (P_M) as

$$P_M = P_D^{N_{vRS}^S}(1 - P_{FA})^{N_{vRS}^H}, \tag{24}$$

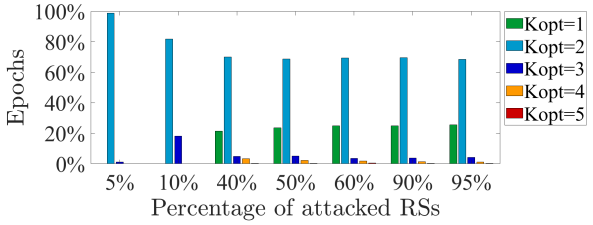
where N_{vRS}^H is the number of healthy RSs. Concerning the second scenario, the results provided in [8] show that the algorithm cannot identify the faulty signals when the pseudorange alteration is 100 m. It allows a P_{FA} of 0.1% with a P_D of 86.94% for an alteration of 200 m, and a P_D of almost 100% for an alteration greater than 300 m. These values have been used to compute the mitigation performancces with Equation 24.

VII. RESULTS AND DISCUSSION

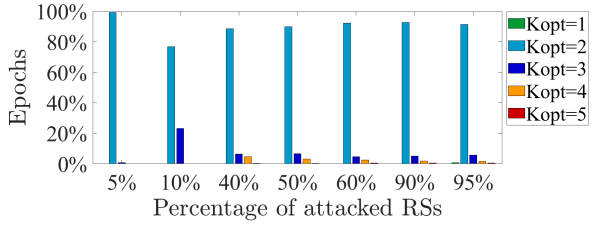
To discuss the results achieved by the proposed method we start analyzing the performances achieved in absence of attack. In normal conditions, the simple clustering procedure, without re-labeling, correctly identified the presence of a single cluster in the 99% of the epochs. However, in presence of a false alarm (i.e. at least two clusters detected) the size of the cluster used to compute the differential corrections ranged from 1 to 5 RSs. In general, this fact would lead to a degradation of the accuracy of the differential corrections. The introduction of time consistency increased the correct estimation of the number of clusters to 99.6%, while by means of the STD re-labeling the value of 100% was reached. In the following subsections, we will discuss how the three methods performed when the spoofer was active. Moreover, we aim at verifying if, besides being time independent, the STD method outperforms the time consistency re-labeling also when the system is under attack.

A. COHERENT ATTACK

First of all, we verified the assumptions made in Section V-C. The number of clusters *Kopt* estimated after the STD re-labeling process in presence of a coherent attack is reported in Figure 6(a).



(a) Results obtained including all the tested attacks



(b) Results obtained excluding the 1-meter attacks

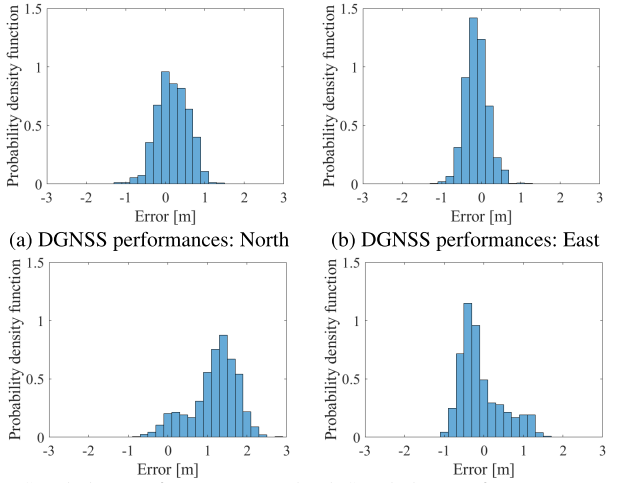
FIGURE 6. Number of clusters (*Kopt*) identified after the STD-relabeling in presence of a coherent attack.

As expected, according to the analysis performed on $c\overline{\delta t_n^{RS}}$, the number of clusters is, for the majority of the epochs, equal to 2. The large number of epochs corresponding to the single cluster is due to the 1-meter attack, as evidenced by Figure 6(b), where the results obtained excluding such an attack are shown. Let us note that an offset of ± 1 meter is comparable with the errors introduced by the receiver noise in DGNSS mode and is a small portion of it when the receiver operates without differential corrections. This is clearly illustrated by Figure 7, where the position estimation error probability density functions for both stand-alone and DGNSS mode under normal conditions are reported.

Moreover, since we used the data recorded by a RS, whose location is carefully chosen to avoid local effects impairing the receiver, such as multipath, the positioning performance of a mobile receiver in normal conditions could be even worse. As a consequence, it is not surprising that the corrections tampered by a 1-meter attack are classified as a single cluster. The reason why this phenomenon does not occur when the percentage of spoofed RSs is small is that, in order to achieve the same position shift, $b_i(k)$ grows when the number of attacked RSs decreases. This is clearly illustrated by Figure 8, where the magnitude of $b_i(k)$, averaged over the spoofed satellites and the epochs, is reported. Then, even if the position shift imposed by the attacker is the same, in the differential corrections space, a greater displacement between the healthy and attacked clusters appears when the percentage of attacked RSs is small. Moreover, Figure 8 shows the difference between the alterations required to achieve shifts of the same magnitude in the North-South and in the East-West

TABLE 1. Detection (D) and mitigation (M) performances for the coherent attack. The detection and mitigation results are expressed as percentages of the total number of epochs both for North (N) and East (E) directions.

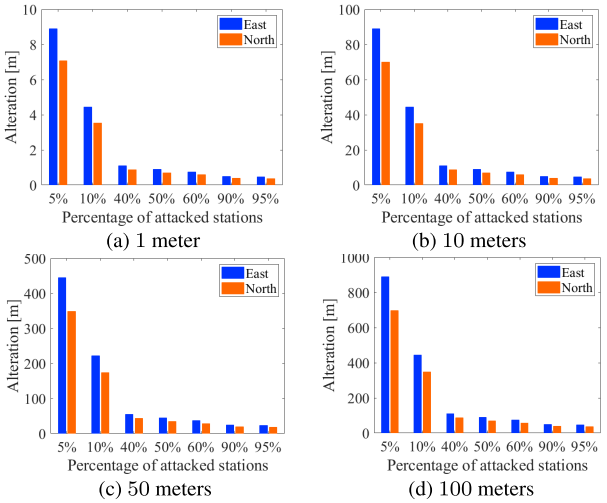
N_{vRS}^{S} (%)	Method	Position shift															
		1 m N		1 m E		10 m N		10 m E		50 m N		50 m E		100 m N		100 m E	
		D	M	D	M	D	M	D	M	D	M	D	M	D	M	D	M
5%	Simple	100	94.7	100	98.6	100	99.3	100	99.3	100	99.3	100	99.3	100	99.3	100	99.3
	TC	100	94.7	100	98.6	100	99.3	100	99.3	100	99.3	100	99.3	100	99.3	100	99.3
	STD	**100**	**95.2**	**100**	**98.6**	**100**	**99.3**	**100**	**99.3**	**100**	**99.3**	**100**	**99.3**	**100**	**99.3**	**100**	**99.3**
10%	Simple	100	50	100	95.5	100	99.4	100	99.6	100	99.3	100	99.4	100	99.3	100	99.3
	TC	100	50	100	95.5	100	99.4	100	99.6	100	99.3	100	99.4	100	99.3	100	99.3
	STD	**100**	**95.2**	**100**	**98.0**	**100**	**99.4**	**100**	**99.6**	**100**	**99.6**	**100**	**99.6**	**100**	**99.6**	**100**	**99.6**
40%	Simple	85.8	1.6	60	9.1	100	96.4	100	99.2	100	99.1	100	99.7	100	98.6	100	99.3
	TC	87.6	1.7	63.3	9.5	100	96.4	100	99.2	100	99.1	100	99.7	100	98.6	100	99.3
	STD	**2**	**0**	**26.9**	**19.8**	**100**	**97.8**	**100**	**99.2**	**100**	**99.6**	**100**	**99.7**	**100**	**99.8**	**100**	**99.9**
50%	Simple	47.2	0	33.6	1.4	100	92.8	100	99.1	100	99.3	100	99.7	100	97.7	100	99.2
	TC	47.8	0	31.8	0.9	100	92.8	100	99.1	100	99.3	100	99.7	100	97.7	100	99.2
	STD	**0.3**	**0**	**10.5**	**2.4**	**100**	**97.3**	**100**	**99.1**	**100**	**99.5**	**100**	**99.7**	**100**	**99.7**	**100**	**99.9**
60%	Simple	20.7	0	6.9	0	100	90.4	100	99.1	100	99.2	100	99.8	100	97.1	100	99.2
	TC	18.6	0	3.7	0	100	90.4	100	99.1	100	99.2	100	99.8	100	97.1	100	99.2
	STD	**0**	**0**	**1.1**	**0**	**100**	**95.4**	**100**	**99.1**	**100**	**99.6**	**100**	**99.8**	**100**	**99.9**	**100**	**100**
90%	Simple	0.1	0	0.2	0	100	100	100	100	100	100	100	100	100	100	100	100
	TC	0	0	0	0	100	100	100	100	100	100	100	100	100	100	100	100
	STD	**0**	**0**	**0**	**0**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
95%	Simple	0.4	0	1.3	0	95.7	95.7	99.7	99.7	100	100	100	100	100	100	100	100
	TC	0.1	0	0.3	0	95.9	95.9	99.9	99.9	100	100	100	100	100	100	100	100
	STD	**0**	**0**	**0.1**	**0**	**95.7**	**95.7**	**99.7**	**99.7**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**



(a) DGNSS performances: North (b) DGNSS performances: East

(c) Stand-alone performances: North (d) Stand-alone performances: East

FIGURE 7. Comparison between nominal stand-alone and DGNSS performances.



(a) 1 meter (b) 10 meters

(c) 50 meters (d) 100 meters

FIGURE 8. RS differential correction alteration.

directions. This difference is due to the geometrical properties of the satellite constellation.

The results for multi-cluster detection obtained with the clustering method without re-labeling (Simple), and with re-labeling based on time consistency (TC) and standard deviation threshold (STD) are shown in Table 1. As it can be seen from the table, the three methods perform almost equally with respect to the attack detection capability, an exception being represented by the 1-meter attack. As explained before, however, it is not surprising that, starting from 40%, the 1-meter attack is not detected.

As for the mitigation results, they are shown in Table 1 where the columns labeled with "M" report the percentage of cases in which the healthy cluster has been properly selected. We verified that the healthy cluster has never been corrupted by the presence of an attacked RS, except for the case of

	1 m N	1 m E	10 m N	10 m E	50 m N	50 m E	100 m N	100 m E
5%	-	-	-	-	-	-	-	✓
10%	✓	✓	✓	✓	-	-	✓	✓
40%	✓	✓	-	✓	-	✓	✓	✗
50%	✓	✓	-	✓	-	✓	✓	✗
60%	✓	✓	-	✓	-	✓	-	-
90%	-	-	-	-	-	-	-	-
95%	-	-	-	-	✗	✗	-	-

FIGURE 9. Mitigation performance summary for the coherent attack. Green tick indicates that the STD-method achieves the best performances, red cross indicates that it does not, and yellow dash indicates that the three methods perform equally.

1-meter attack for which, as for the detection, the main failure occurs. As shown in Table 1, the mitigation performances of the STD method grows with the increase of the alteration bias. This is not exactly true for the two other methods since, without a proper re-labeling procedure, the final clustering does not group all the healthy stations in a unique cluster. For a rapid comparison between the three methods we show a summary in Figure 9. More in details, the green tick indicates that the STD method performs better than the other two, the yellow dash indicates that the three methods perform equally, and the red cross that the STD method is not the best one.

Moreover, Table 1 shows that an increasing percentage of attacked RSs implies a better mitigation performance. The investigations carried out to understand the reason behind this phenomenon enlightened that, sometimes, the clustering procedure included one healthy RS in the cluster of spoofed stations. Moreover, a deeper analysis evidenced that the miss-classified RS was always the same, namely RS number 9. The reason why the mitigation performances seem to improve when the percentage of attacked RSs increases is that in the 90% and the 95% cases, that station was spoofed and the miss-classification event did not occur.

For a deeper insight of this unexpected behavior, in Figure 10 we reported, for each satellite, the minimum, the maximum, and the median differential correction, without attack, for RS #9 overlapped with the same statistics computed on the set of the remaining RSs. Being the analyzed data related to nominal conditions, we expected the corrections of RS #9 to lie within the variability experienced by the corrections of all the others. This is definitely not true for a subset of satellites, namely satellites 6, 11, 17, 18, 19 and 28. As a consequence, under attack, the algorithm recognizes the anomalous behaviour of RS #9. Therefore, the exclusion of this RS from the healthy cluster can be considered the right choice. The mitigation performances having considered the RS #9 as unhealthy are shown in Table 2 for the STD method. As expected, this time the performances do not increase when a larger number of RSs is spoofed. Let us note that, when no attack is present, RS #9 is not excluded from the healthy cluster because, for the epochs in which the anomalous behaviour takes place, the GAP statistic gives as output $Kopt = 1$. Following Algorithm 1, this implies that no further check on the standard deviation is performed, and the anomaly is not detected. Improving the early detection procedure in absence of attack could be the subject of further research.

The effectiveness of the mitigation with respect to localization accuracy losses caused by spoofing is illustrated by Figure 11, where the North and East components of the position error of the target RS are shown. As can be seen from the plots, until the percentage of attacked stations is smaller or equal to 90%, only the results for 1-meter attacks (cyan and dark pink dots) are significantly different from the case of absence of attack. A different outcome is obtained for the 95% cases. Under these circumstances, in fact, also the 10-meter attack (green and yellow dots) becomes partially successful. Let us highlight, however, that an attack involving the 95% of the RSs requires a remarkable effort. Moreover, the comparison with Figure 4 clearly shows that the mitigation strategy is successful.

In conclusion, considering that the detection performances are almost the same for the three methods, that the time consistency check introduces a new system vulnerability, and that the best performances both in absence of attack and in terms of mitigation are achieved by the STD method, we recommend this last as detection strategy for the coherent attack scenario. In addition, let us note that, having performed

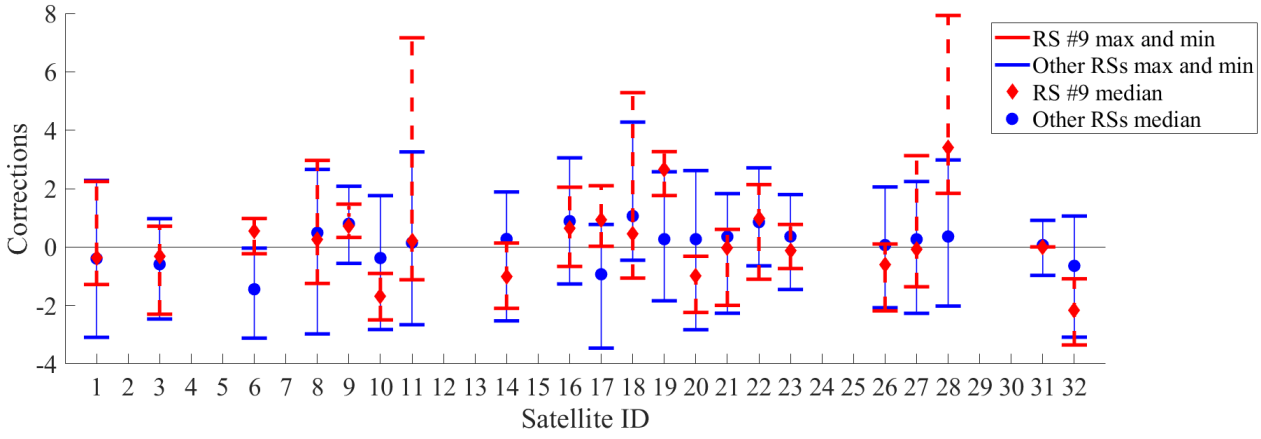


FIGURE 10. Comparison between RS #9 (in red dashed line) and the set of the other RSs (in blue thin line), without attack. The blue dot and red diamond represent the median, and the bars indicate the maximum and minimum values.

TABLE 2. Mitigation performances for the coherent attack (RS #9 excluded). The results are expressed as percentages of the total amount of epochs both for North (N) and East (E) directions.

N^S_{vRS} (%)	Method	Position shift							
		1 m N	1m E	10 m N	10 m E	50 m N	50 m E	100 m N	100 m E
5%	STD	100	100	100	100	100	100	100	100
10%	STD	97.1	99.5	100	100	100	100	100	100
40%	STD	0	20.3	99.6	100	100	100	100	100
50%	STD	0	2.4	99.1	100	100	100	100	100
60%	STD	0	0	97	100	100	100	100	100
90%	STD	0	0	100	100	100	100	100	100
95%	STD	0	0	95.7	99.7	100	100	100	100

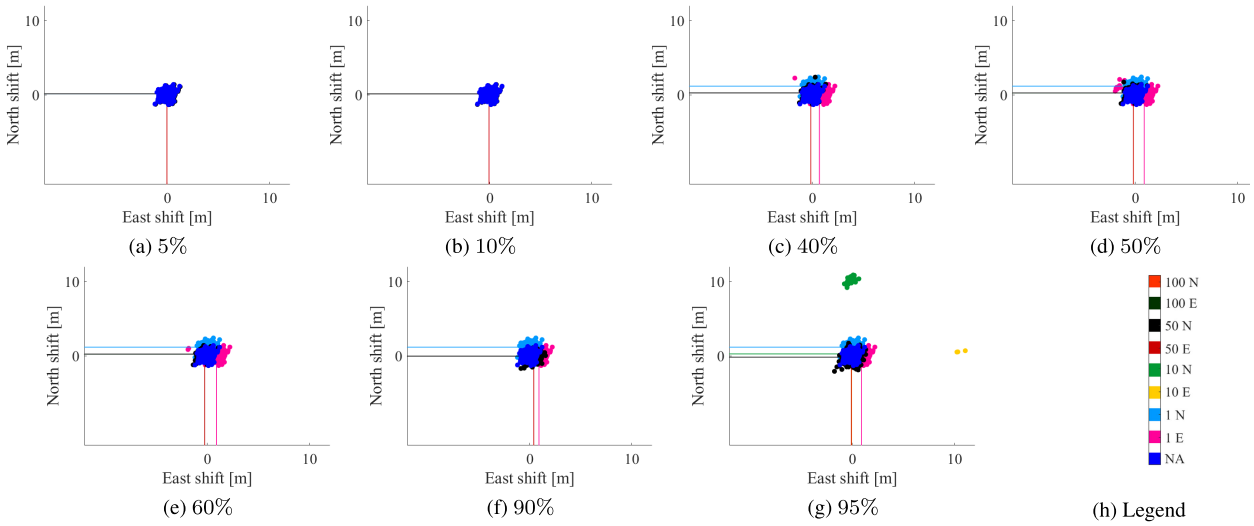


FIGURE 11. Mitigation performances for the different percentages of attacked RSs. Different colors represent different kinds of attack both for North (N) and East (E) directions. The blue plot represents the case in which there is no attack (NA).

separate tests simulating the presence or absence of attack, the performances of time-consistency re-labeling have been assessed under the best circumstances. As a consequence, the STD superiority has been proven under the most challenging conditions.

Let us now compare our method to the one proposed in [7]. According to the analysis performed in Section VI, the method proposed in [7] is unable to detected attacks producing offsets smaller than γ_d which, in turn, fairly exceeds 16 meters. As a consequence, it is completely outperformed by our method for offsets below $10 - 20$ meters. A similar situation arises for larger values, because no attack mitigation is provided in [7].

B. NON-COHERENT ATTACK

As for the coherent attack, we will start analyzing the attack effect on the clustering procedure. The number of clusters *Kopt* estimated after the STD-based re-labeling is shown in Figure 12. As clearly shown from Figure 12, a greater variability in the alteration of $c\overline{\delta t_n^{RS}}$ corresponds to a greater number of clusters in the N_{sat}-dimensional space. Moreover, based on the analysis of the behaviour of RS #9, its exclusion from the set of healthy RSs even when it has not been spoofed is considered correct, as in the case of the coherent attack.

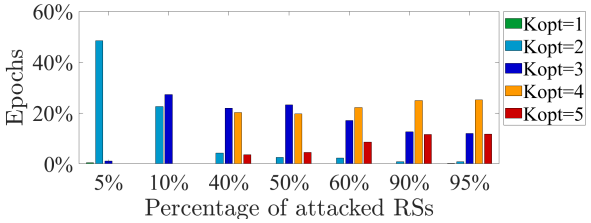


FIGURE 12. Number of clusters (*Kopt*) identified after the STD-relabeling in presence of a non-coherent attack.

The detection performances are shown in Table 3. In the table, the annotation "shared sat" in the "Method" column indicates the case in which all the satellites visible from all the RSs have been attacked. As explained in Section VI-A, these results are used to compare the performances of the simple, the TC and the STD methods. Table 3 shows that the detection performances for the 95% case are worse than for the others. The reason is that we used the GAP statistic for a value of candidate K in the range from 1 to 5, whereas in some epochs, for the 95% case, 7 clusters exist. This is motivated by a greater difference between the values of $c\overline{\delta t_n^{RS}}$ when the number of attacked RSs increases. From Equation 18, in fact, it is clear that when $N_{att,n}$ is equal for all the RSs, as in the performed attack, the differential correction

TABLE 3. Detection (D) and mitigation (M) performances for the non-coherent attack. The results are expressed as percentages of the total amount of epochs.

N^S_{vRS} (%)	Method	Position shift									
		50 m		100 m		200 m		300 m		500 m	
		D	M	D	M	D	M	D	M	D	M
5 %	Simple (shared sat)	100	97.6	100	99.1	100	99.1	100	99.1	100	99.1
	TC (shared sat)	100	97.6	100	99.1	100	99.1	100	99.1	100	99.1
	STD (shared sat)	**100**	**99.1**	**100**	**99.1**	**100**	**99.1**	**100**	**99.1**	**100**	**99.1**
	STD (1 sat)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SoA (Scenario 1)	-	99.71	-	99.71	-	99.71	-	99.71	-	99.71
	SoA (Scenario 2)	-	/	-	/	-	85.30	-	98.12	-	98.12
10 %	Simple (shared sat)	100	81.3	100	98	100	99.6	100	99.6	100	99.6
	TC (shared sat)	100	81.3	100	98	100	99.6	100	99.6	100	99.6
	STD (shared sat)	**100**	**97.6**	**100**	**99.4**	**100**	**100**	**100**	**100**	**100**	**100**
	STD (1 sat)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SoA (Scenario 1)	-	99.62	-	99.62	-	99.62	-	99.62	-	99.62
	SoA (Scenario 2)	-	/	-	/	-	74.24	-	98.22	-	98.22
40 %	Simple (shared sat)	100	72.2	100	95	100	98.9	100	99.2	100	99.3
	TC (shared sat)	100	72.2	100	95	100	98.9	100	99.2	100	99.3
	STD (shared sat)	**100**	**98.4**	**100**	**99.5**	**100**	**99.9**	**100**	**100**	**100**	**100**
	STD (1 sat)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**99.9**	**100**	**99.9**
	SoA (Scenario 1)	-	99.08	-	99.08	-	99.08	-	99.08	-	99.08
	SoA (Scenario 2)	-	/	-	/	-	32.25	-	98.81	-	98.81
50 %	Simple (shared sat)	100	70.5	100	95.2	100	98.7	100	99	100	99.2
	TC (shared sat)	100	70.5	100	95.2	100	98.7	100	99	100	99.2
	STD (shared sat)	**100**	**96.8**	**100**	**99**	**100**	**99.9**	**100**	**100**	**100**	**100**
	STD (1 sat)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**99.9**	**100**	**99.8**
	SoA (Scenario 1)	-	98.91	-	98.91	-	98.91	-	98.91	-	98.91
	SoA (Scenario 2)	-	/	-	/	-	24.43	-	99	-	99
60 %	Simple (shared sat)	100	70.3	100	92.3	100	97.2	100	97.7	100	98.3
	TC (shared sat)	100	70.3	100	92.3	100	97.2	100	97.7	100	98.3
	STD (shared sat)	**100**	**94.7**	**100**	**97.7**	**100**	**99.2**	**100**	**99.5**	**100**	**99.7**
	STD (1 sat)	**100**	**100**	**100**	**100**	**100**	**99.9**	**100**	**99.2**	**100**	**98.9**
	SoA (Scenario 1)	-	98.73	-	98.73	-	98.73	-	98.73	-	98.73
	SoA (Scenario 2)	-	/	-	/	-	18.50	-	99.20	-	99.20
90 %	Simple (shared sat)	100	100	100	100	100	100	100	100	100	100
	TC (shared sat)	100	100	100	100	100	100	100	100	100	100
	STD (shared sat)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	STD (1 sat)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SoA (Scenario 1)	-	98.20	-	98.20	-	98.20	-	98.20	-	98.20
	SoA (Scenario 2)	-	/	-	/	-	8.04	-	99.80	-	99.80
95 %	Simple (shared sat)	99.6	99.6	99.6	99.6	99.6	99.6	99.6	99.6	99.6	99.6
	TC (shared sat)	99.6	99.6	99.6	99.6	99.6	99.6	99.6	99.6	99.6	99.6
	STD (shared sat)	**99.6**	**99.6**	**99.6**	**99.6**	**99.6**	**99.6**	**99.6**	**99.6**	**99.6**	**99.6**
	STD (1 sat)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SoA (Scenario 1)	-	98.11	-	98.11	-	98.11	-	98.11	-	98.11
	SoA (Scenario 2)	-	/	-	/	-	6.99	-	99.90	-	99.90

alteration depends only on $N_{sat,n}$. As a consequence, when the number of attacked RSs grows, a greater number of clusters will arise due to the increase in the variability of $N_{sat,n}$. A trade-off between the number of tested Ks and performances can be defined according to a target computational cost. Moreover, even if the detection performances are the same for all the methods, the mitigation results are better for the STD technique, as shown both in Table 3, and in Figure 13. For this reason, we selected the STD method to test the algorithm when a single satellite has been attacked.

The results reported in the rows "STD (1 sat)" of Table 3 show that our algorithm is independent from the number of attacked satellites. A similar study has not been performed for the coherent attack since it does not make sense to alter a single satellite measurement to obtain a predefined position shift of the end receiver. In conclusion, we recommend the STD method as detection and mitigation strategy also for the non-coherent scenario. The attack effect on the position computed from the final user when the STD mitigation method is deployed are shown in Figure 14. The comparison with

	50 m	100 m	200 m	300 m	500 m
5%	✓	–	–	–	–
10%	✓	✓	✓	✓	✓
40%	✓	✓	✓	✓	✓
50%	✓	✓	✓	✓	✓
60%	✓	✓	✓	✓	✓
90%	–	–	–	–	–
95%	–	–	–	–	–

FIGURE 13. **Mitigation performance summary for the non-coherent attack. Green tick indicates that the STD-method achieves the best performances, and yellow dash indicates that the three methods perform equally.**

Figure 5 clearly shows the success of the mitigation strategy. Table 3 also shows the results that would be obtained if the method presented in [8] was implemented in each RS. The two scenarios described in Section VI-A are shown in Table 3 as "SoA (Scenario 1)" and "SoA (Scenario 2)". Concerning the first scenario, our results are comparable or better than the ones obtained in [8]. Moreover, working on a single constellation, we do not have any reference, performing spoofing detection and mitigation in a more challenging condition. We observe that our mitigation strategy does not depend on the number of constellations, since it would change only the dimensions of the clustering space. Finally, our false alarm probability is smaller than the one achieved in [8]. As for the second scenario, the proposed method clearly outperforms the results presented in [8]. For a better comparison, Figure 15 reports the performances of the STD method, both for the single and shared satellite attacks, and the results achieved by the state-of-the-art approaches.

C. COMPUTATIONAL COMPLEXITY

In order to provide a measure of the computational complexity of Algorithm 1, we computed the time needed to execute it when all kinds of attacks are performed, as well as when there is no attack. The mean (μ) and variance (σ^2) of the computational times are shown in Table 4. The experiments were conducted using a DELL Alienware Aurora R8 Desktop Computer, equipped with an Intel(R) Core(TM) i7-9700K CPU @ 3.60 GHz and 32 GB of RAM. Based on the results reported in Table 4, the execution time is dominated by the GAP statistic procedure, since the results are almost independent from K_{opt}. Otherwise, in fact, the execution times in absence of attack should be smaller. At last, let us note that the code is currently implemented in MATLAB and it is not optimized for computational efficiency.

TABLE 4. **Computational cost in seconds.**

	No attack	Coherent	Non-coherent (1 sat)	Non-coherent (shared sat)
μ	1.43	1.42	1.43	1.41
σ^2	2.56×10^{-4}	4.66×10^{-4}	1.49×10^{-4}	4.42×10^{-4}

D. FINAL REMARKS

The provided experimental tests prove the suitability of the proposed approach for detecting and mitigating spoofing attacks towards the Augmentation Networks. More specifically, based on the performed tests, we recommend the use of the STD re-labeling. The presented technique can face from simple to massive attacks, thus being eligible for critical infrastructures (e.g. trains and autonomous road vehicles). Although the presented method was demonstrated to be effective, further research will be focused on refining the proposed technique. More specifically, the behavior of the detection and mitigation system in absence of attack can be improved, as highlighted by the analysis concerning RS #9, and additional tests can be conducted. Moreover, further studies on the threshold computation procedure for STD re-labeling can be performed. More specifically, different settings of γ_c can be

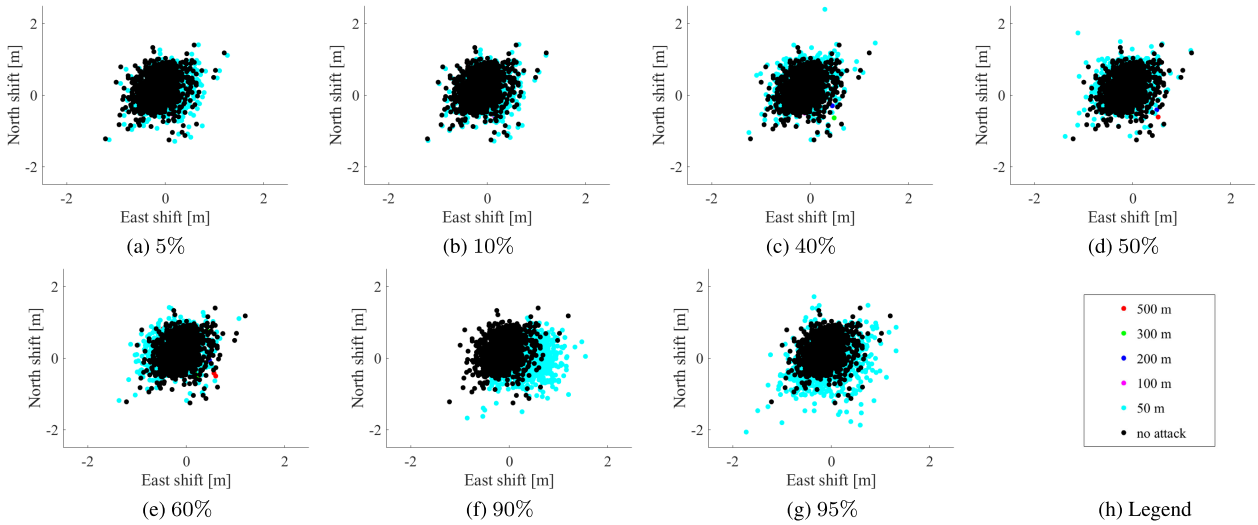


FIGURE 14. **Non-coherent attack effect on the final user position when one satellite has been attacked and STD-based mitigation has been deployed. Different colors represent the different pseudorange alterations. The black plot represents the case in which there is no attack.**

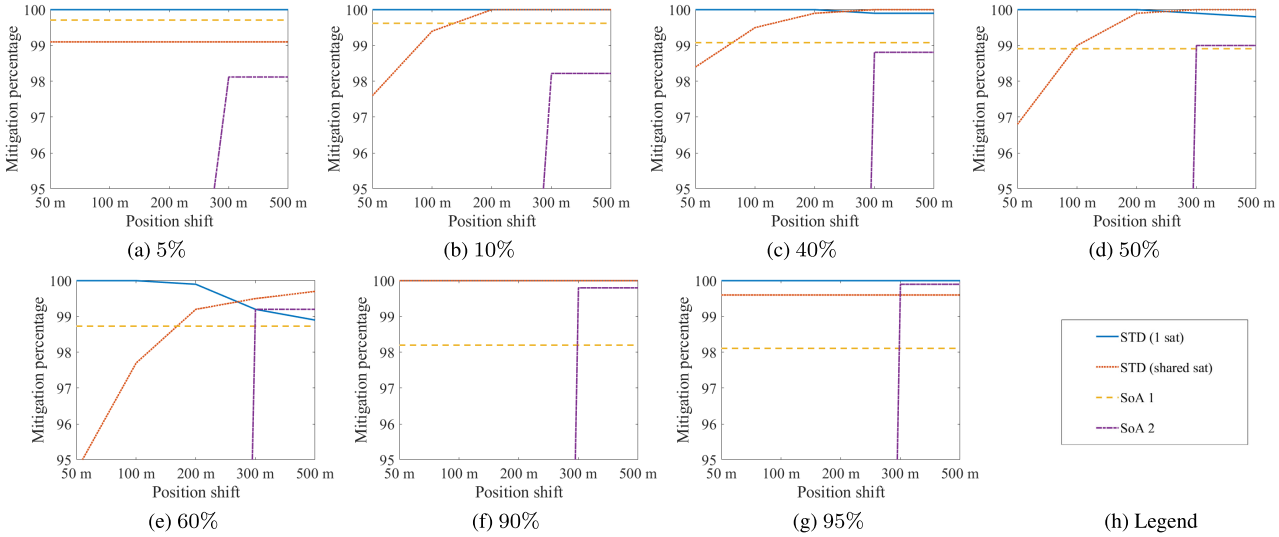


FIGURE 15. Mitigation performance comparison between the proposed method with STD-based re-labeling and state-of-the-art approaches.

investigated. For instance, with respect to the standard deviation, the proportionality constant can be changed, or more complex relations can be defined.

VIII. CONCLUSION

In this work, we proposed a detection and mitigation technique for GNSS spoofing targeting an Augmentation Network. The proposed method exploits the spatial coherence among the corrections estimated by the Reference Stations spread over the area served by the Augmentation Network. In fact, when the attacker is not able to attack all RSs at the same time, the differential corrections group into two or more clusters while, under nominal conditions, only a single cluster is present. Our technique exploits this behavior and triggers an alarm whenever a multi-cluster scenario is detected. Moreover, the proposed algorithm resorts to a re-labeling procedure to reduce false alarms. We introduced two re-labeling techniques and selected the STD-based criterion as leading strategy. At last, we proposed a mitigation method to identify the healthy cluster.

Moreover, we introduced two attack models, tested the presented method against them and we obtained promising results. In addition, the proposed method has been compared both to an algorithm for spoofing detection at the Augmentation Network level, and to a clustering-based technique to detect and mitigate spoofing at the receiver level, here applied to each RS. To the best of our knowledge, the DGNSS spoofing detection technique we used for the comparison is the only one proposed in the literature. As a consequence, the presented method introduces a key novelty by exploiting the differential correction properties to address spoofing detection. Moreover, it is the first technique which provides spoofing mitigation for attacks directed to the Augmentation Networks. Our technique provided promising results, achieving better or comparable performances with respect to receiver-level state-of-the-art approaches.

APPENDIX A
SIGNAL MODEL

The signal $x(t)$ transmitted by one satellite can be represented as

$$\begin{aligned} x(t) &= x_c(t)cos(2\pi f_L t) - x_s(t)sin(2\pi f_L t) \\ &= Re\left\{e^{j2\pi f_L t}\underline{x}(t)\right\} \end{aligned} \tag{25}$$

where $\underline{x}(t)$, $x_c(t)$, and $x_s(t)$ respectively are the complex envelope and the in phase and in quadrature components of $x(t)$ with respect to the carrier frequency f_L, for which the following expression holds:

$$\begin{aligned} \underline{x}(t) &= x_c(t) + jx_s(t) \\ &= \sqrt{2P_c}D_c(t)C_c(t) + j\sqrt{2P_s}D_s(t)C_s(t), \end{aligned} \tag{26}$$

where P_c and P_s are the signal powers of the related channels, $D_c(t)$ and $D_s(t)$ represent the navigation data, $C_c(t)$ and $C_s(t)$ are the spreading codes.

Under normal conditions, the received signal is given by the sum of the signals coming from the visible satellites, plus an additive noise term. Due to the pseudo-orthogonality of the spreading codes, however, it is possible to analyze each signal independently. In the following, then, we will refer to the signal received by a single satellite which can be modeled as

$$r(t) = Re\left\{ae^{j2\pi(f_L+f_d)t}e^{j\phi}\underline{x}(t-\tau)\right\} \tag{27}$$

where a represents the attenuation factor, f_d is the Doppler shift, τ is the propagation time delay, and ϕ is the carrier phase delay

$$\phi = 2\pi f_L \tau. \tag{28}$$

All the mentioned parameters are different for each satellite.

During the tracking phase, the RF signal base band conversion based on a Phase Locked Loop (PLL) driven local

oscillator, and its Analog to Digital (AD) conversion can be equivalently modeled by the sampling of the signal

$$r_{BB}(t) = Re\left\{ae^{j(\phi+\gamma_0)}\underline{x}(t-\tau)\right\} \tag{29}$$

so that

$$r_{BB}(kT) = Re\left\{ae^{j(\phi+\gamma_0)}\underline{x}(kT-\tau_k)\right\} \tag{30}$$

where γ_0 is an additional phase shift due to the receiver clock error, which is the same for all satellites, and T is the sampling period. In the following expressions, T will be omitted for sake of simplicity. The receiver estimates the code delay $\hat{\tau}_k$, at the k-th epoch, through a correlation procedure with a local generated replica of the transmitted signal. The code pseudorange can then be computed as

$$\rho(k) = c\hat{\tau}_k, \tag{31}$$

where c is the speed of light in the vacuum [24].

When spoofing is performed, the signal received by the target antenna is made of two components: the healthy signal (H) and the the attacker's one (S). As a consequence, Equation 30 becomes

$$\begin{aligned} r_{BB}(k) &= r^H_{BB}(k) + r^S_{BB}(k) \\ &= Re\left\{a^H e^{j(\phi^H+\gamma_0)}\underline{x}^H(kT-\tau^H_k)\right\} \\ &\quad + Re\left\{a^S e^{j(\phi^S+\gamma_0)}\underline{x}^S(kT-\tau^S_k)\right\}. \end{aligned} \tag{32}$$

As detailed in [9], in order to perform the attack, the counterfeit signal has to be aligned both in terms of code and phase to the authentic one. Once the spoofing correlation peak is aligned to the genuine one, the power of the fake signal can be increased gradually so that it takes control of the receiver. Once the target receiver starts tracking the spoofing signal, $r_{BB}(k)$ equals $r^S_{BB}(k)$, and the receiver can be moved from its true position. To cause the desired displacement, the attacker has to properly configure the counterfeit signal parameters. Since we introduced two code pseudorange attack case studies, in the following we will focus only on the code shift. In order to have a pseudorange alteration equal to $b_{i,n}(k)$ for the i-th satellite and the n-th RS, the attacker has to alter τ_k with a bias equal to $\frac{b_{i,n}(k)}{c}$. This alteration implies the modification of the delay bias so that

$$\tau^S_k = \tau^H_k + \frac{b_{i,n}(k)}{c}. \tag{33}$$

Concerning the navigation data, they can be properly predicted as detailed in [9].

APPENDIX B
THRESHOLD COMPUTATION

The core assumption of the threshold setting procedure is that the differential corrections computed for each satellite follow a Gaussian distribution. Our approach is based on evaluating if the clusters identified by the GAP statistic can be considered well-separated or if they have to be fused. To this aim, we compare the distance between the centroids of the identified clusters with the dispersion of the differential corrections under nominal conditions. To provide a measure of such dispersion, we consider an interval amplitude proportional to the standard deviation under nominal conditions: $\gamma_c = \kappa\sigma_c$. To obtain such value, an estimate of the dispersion of the differential correction in absence of attack is needed. To do so, a recorded time series of differential corrections under nominal conditions can be exploited. More specifically, the differential corrections for all the satellites visible from all the RSs can be extracted for a set of epochs $t = \{1, 2, \ldots T\}$. Then, the standard deviation $\sigma_{i,t}$ can be computed for the i-th satellite at epoch t. The threshold $\gamma_{i,t}$ for satellite i and epoch t, can be defined as:

$$\gamma_{i,t} = \kappa\sigma_{i,t}. \tag{34}$$

The values $\gamma_{i,t}$ can then be averaged to obtain a threshold for all the satellites as:

$$\begin{aligned} \gamma_t = \frac{1}{N_{sat}}\sum_{i=1}^{N_{sat}}\gamma_{i,t} = \frac{1}{N_{sat}}\sum_{i=1}^{N_{sat}}\kappa\sigma_{i,t} \\ \kappa\frac{1}{N_{sat}}\sum_{i=1}^{N_{sat}}\sigma_{i,t} = \kappa\sigma_t. \end{aligned} \tag{35}$$

At last, to obtain a general threshold, the values γ_t can be averaged as:

$$\gamma_c = \frac{1}{T}\sum_{i=1}^{T}\gamma_t = \frac{1}{T}\sum_{i=1}^{T}\kappa\sigma_t = \kappa\sigma_c. \tag{36}$$

Since the standard deviation is defined with respect to the mean, and we are comparing two cluster centroids, the threshold Γ used to check if the two clusters are sufficiently distant to be considered separate entities is set equal to:

$$\Gamma = 2\gamma_c. \tag{37}$$

REFERENCES

[1] J. Sanz Subirana, J. M. Juan Zornoza, and M. Hernández-Pajares, *GNSS Data Processing, Volume I: Fundamentals and Algorithms* (ESA Communications). Noordwijk, The Netherlands: ESTEC, 2013. [Online]. Available: https://gssc.esa.int/navipedia/GNSS_Book/ESA_GNSS-Book_TM-23_Vol_I.pdf

[2] J. Platt, Nat. Risk Manage. Center (NRMC) Tech. Rep., Jun. 2019. [Online]. Available: https://www.gps.gov/governance/advisory/meetings/2019-06/platt.pdf

[3] London Economics, "Economic impact to the UK of a disruption to GNSS," London Econ., London, U.K., Tech. Rep., Jun. 2017. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/619545/17.3254_Economic_impact_to_UK_of_a_disruption_to_GNSS_-_Showcase_Report.pdf

[4] R. T. Ioannides, T. Pany, and G. Gibbons, "Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques," *Proc. IEEE*, vol. 104, no. 6, pp. 1174–1194, Jun. 2016.

[5] T. Nighswander, B. Ledvina, J. Diamond, R. Brumley, and D. Brumley, "GPS software attacks," in *Proc. ACM Conf. Comput. Commun. Secur.-CCS. Assoc. Comput. Machinery*, New York, NY, USA, 2012, pp. 450–461.

[6] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. Roy. Stat. Soc., Ser. B (Stat. Methodol.)*, vol. 63, no. 2, pp. 411–423, May 2001, doi: 10.1111/1467-9868.00293.

[7] E. Ochin, "Detection of spoofing using differential GNSS," *50 Sci. J. Maritime Univ. Szczecin*, vol. 50, no. 122, pp. 59–67, 2017, doi: 10.17402/217.
[8] K. Zhang and P. Papadimitratos, "Secure multi-constellation GNSS receivers with clustering-based solution separation algorithm," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2019, pp. 1–9.
[9] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," in *Proc. 21st Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, Savannah, GA, USA, Sep. 2008, pp. 2314–2325.
[10] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *Proc. 18th ACM Conf. Comput. Commun. Secur. (CCS)*, Chigago, IL, USA, 2011, pp. 75–86.
[11] K. C. Zeng, S. Liu, Shinan, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All your GPS are belong to us: Towards stealthy manipulation of road navigation systems," in *Proc. 27th {USENIX} Secur. Symp. ({USENIX} Secur.)*, Baltimore, MD, USA, 2018, pp. 1527–1544. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/zeng
[12] A. Neri, S. Baldoni, and R. Capua, "On cyber-security of augmentation networks," in *Proc. Int. Tech. Meeting Inst. Navigat.*, Reston, VA, USA, Feb. 2019, pp. 408–422.
[13] Q. Luo, Y. Cao, J. Liu, and A. Benslimane, "Localization and navigation in autonomous driving: Threats and countermeasures," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 38–45, Aug. 2019, doi: 10.1109/MWC.2019.1800533.
[14] Y. Sun and L. Fu, "A new threat for pseudorange-based RAIM: Adversarial attacks on GNSS positioning," *IEEE Access*, vol. 7, pp. 126051–126058, 2019, doi: 10.1109/ACCESS.2019.2939141.
[15] J. Zidan, E. I. Adegoke, E. Kampert, S. A. Birrell, C. R. Ford, and M. D. Higgins, "GNSS vulnerabilities and existing solutions: A review of the literature," *IEEE Access*, early access, Feb. 13, 2020, doi: 10.1109/ACCESS.2020.2973759.
[16] E. G. Manfredini, D. M. Akos, Y.-H. Chen, S. Lo, T. Walter, and P. Enge, "Effective GPS spoofing detection utilizing metrics from commercial receivers," in *Proc. Int. Tech. Meeting Inst. Navigat.*, Reston, VA, USA, Feb. 2018, pp. 672–689.
[17] C. Sun, J. W. Cheong, A. G. Dempster, H. Zhao, and W. Feng, "GNSS spoofing detection by means of signal quality monitoring (SQM) metric combinations," *IEEE Access*, vol. 6, pp. 66428–66441, 2018, doi: 10.1109/ACCESS.2018.2875948.
[18] K. Liu, W. Wu, Z. Wu, L. He, and K. Tang, "Spoofing detection algorithm based on pseudorange differences," *Sensors*, vol. 18, no. 10, p. 3197, Sep. 2018, doi: 10.3390/s18103197.
[19] A. R. Baziar, M. R. Mosavi, and M. Moazedi, "Spoofing mitigation using double stationary wavelet transform in civil GPS receivers," *Wireless Pers. Commun.*, vol. 109, no. 3, pp. 1827–1844, Aug. 2019, doi: 10.1007/s11277-019-06654-x.
[20] A. Broumandan and G. Lachapelle, "Spoofing detection using GNSS/INS/odometer coupling for vehicular navigation," *Sensors*, vol. 18, no. 5, p. 1305, Apr. 2018, doi: 10.3390/s18051305.
[21] G. Panice, S. Luongo, G. Gigante, D. Pascarella, C. Di Benedetto, A. Vozella, and A. Pescape, "A SVM-based detection approach for GPS spoofing attacks to UAV," in *Proc. 23rd Int. Conf. Autom. Comput. (ICAC)*, Huddersfield, U.K., Sep. 2017, pp. 1–11.
[22] S. Semanjski, I. Semanjski, W. De Wilde, and S. Gautama, "Use of supervised machine learning for GNSS signal spoofing detection with validation on real-world meaconing and spoofing data—Part II," *Sensors*, vol. 20, no. 7, p. 1806, Mar. 2020, doi: 10.3390/s20071806.
[23] E. Shafiee, M. R. Mosavi, and M. Moazedi, "Detection of spoofing attack using machine learning based on multi-layer neural network in single-frequency GPS receivers," *J. Navigat.*, vol. 71, no. 1, pp. 169–188, Jan. 2018, doi: 10.1017/S0373463317000558.
[24] P. Teunissen and O. Montenbruck, *Springer Handbook of Global Navigation Satellite Systems*. Cham, Switzerland: Springer, 2017.
[25] D. Gebre-Egziabher and S. Gleason, *GNSS Applications and Methods*, Norwood, MA, USA: Artech House, 2019.
[26] R. Webster and M. Oliver, *Geostatistics for Environmental Scientists*. Hoboken, NJ, USA: Wiley, 2007.
[27] E. Lisova, E. Uhlemann, W. Steiner, J. Åkerberg, and M. Björkman, "Digital single sideband detection for interferometric sensors," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control, Commun. (ISPCS)*, Sep. 2016, pp. 1–6.
[28] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979, doi: 10.1109/TPAMI.1979.4766909.
[29] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist.-Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974, doi: 10.1080/03610927408827101.
[30] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.
[31] A. Almalawi, A. Fahad, Z. Tari, A. Alamri, R. AlGhamdi, and A. Y. Zomaya, "An efficient data-driven clustering technique to detect attacks in SCADA systems," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 5, pp. 893–906, May 2016.
[32] *RINEX Data*. Accessed: Jun. 18, 2019. [Online]. Available: ftp://cddis.gsfc.nasa.gov/gnss/data/daily/2019/001/
[33] *IGS Network Map*. Accessed: Mar. 11, 2020. [Online]. Available: http://www.igs.org/network

SARA BALDONI (Graduate Student Member, IEEE) was born in Rome, Italy, in 1994. She received the bachelor's degree in electronics engineering and the master's degree in information and communication technology engineering from Roma Tre University, in 2016 and 2018, respectively, where she is currently pursuing the Ph.D. degree in applied electronics with the Department of Engineering. Her main research interests include communication security and navigation and localization systems.

FEDERICA BATTISTI (Senior Member, IEEE) received the Master of Science degree in electronic engineering and the Ph.D. degree from Roma Tre University, Rome, Italy, in 2006 and 2010, respectively. She is currently an Assistant Professor with the Department of Information Engineering, University of Padua. Her research interests include multimedia quality assessment and security. She serves as an Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA, *EURASIP Journal on Image and Video Processing*, and *Signal Processing: Image Communication* (Elsevier).

ALESSANDRO NERI (Member, IEEE) was born in Viterbo, Italy, in 1954. He received the Ph.D. degree *(cum laude)* in electronic engineering from the University of Rome "La Sapienza," in 1977. In 1978, he joined the Research and Development Department, Contraves Italiana S. p. A., where he gained a specific expertise in the field of radar signal processing and in applied detection and estimation theory, and became the Chief of the Advanced Systems Group. In 1987, he joined the INFOCOM Department, Engineering Faculty, University of Rome "La Sapienza," as an Associate Professor in signal and information theory. In November 1992, he joined the Electronic Engineering Department, Roma Tre University, as an Associate Professor in electrical communications, where he became a Full Professor in telecommunications in September 2001. Since December 2008, he has been the President of the RadioLabs Consortium (Consorzio Università Industria–Laboratori di Radiocomunicazioni), a not-for-profit Research Center created in 2001 to promote tight cooperation on applied research programs between universities and industries, and currently linking the University of Rome "Tor Vergata," Roma Tre University, The University of Aquila, Hitachi Rail STS, and WestPole. His research interests include information theory, detection and estimation theory, digital signal processing, and image processing and their applications to both telecommunications systems, navigation, and remote sensing, including navigation and localization, systems mobile communications, the Internet of Things, communications security, machine intelligence, and multimedia. He is a member of the Institute of Navigation (ION).