GNSS Spoofing Detection using Multiple Sensing Devices and Decision Tree Classifier

Xin QI[†a)], Toshio SATO[†], Zheng WEN[†], Masaru TAKEUCHI[††], Yutaka KATSUYAMA[†], Kazuhiko TAMESUE[†], Kazue SAKO[†], *Members*, Jiro KATTO[†], *and* Takuro SATO[†], *Fellows*

SUMMARY For next-generation logistics systems using autonomous vehicles and drones, spoofing of the GNSS location data will induce serious problems. Although signal-based anti-spoofing has been studied, it is difficult to apply to current commercial GNSS modules in many cases. We investigate possibilities to detect spoofing of GNSS location data using multiple sensing devices and a decision tree classifier. Multiple features using the GNSS, beacons, and the IMU are defined and create a model to detect spoofing. Experimental results using a learning-based classifier indicate higher performances and generalization capability. The results also show that distance from beacons is useful for detecting GNSS spoofing and indicate prospects of installation for future drone highways.
keywords: *location, spoofing, GNSS, beacons, drones.*

1. Introduction

Location information captured by Global Navigation Satellite Systems (GNSS) is important for next-generation logistics systems using autonomous vehicles and drones. GNSS is susceptible to interference from jamming and spoofing due to its open signal structure and low signal power. As a result, the safety and integrity of GNSS are threatened. Recently, actual cases of spoofing have been reported [1] and countermeasures are being considered.

Most of the spoofing detection methods reported so far are based on signal-level analysis [2][3]. Spoofing detection by signal level analysis requires special circuitry and cannot take advantage of the low-cost GNSS modules that are currently available to the embedded devices. Another approach is to detect GNSS spoofing by integrating multiple sensing devices, including Inertial Measurement Units (IMU) and Inertial Navigation Systems (INS) [4][5]. However, in practice, the available devices and methods differ depending on the environment. The generalization of integration of multiple sensing devices is a challenge.

We will investigate methods to detect GNSS spoofing using multiple low-cost devices and aim to generalize the method using a machine learning approach.

2. Problem Statement

2.1 Spoofing GNSS Signals

From banks to smart power grids, data centers, logistics giants, high-frequency trading firms, 5G, and digital television broadcasting, several industries critically rely on global navigation satellite systems, GNSS, for precise synchronization of time and position. Sadly, due to the low power level of satellite signals, GNSS receivers are highly susceptible to RF interference. Closely located powerful transmitters like digital television or cellular base stations can easily degrade the quality of GNSS signals, which can be critical for time-sensitive applications like 5G. Widely distributed low-cost jammers easily block GNSS signals hundreds of meters away. But it is nothing compared to the threat posed by GNSS spoofing.

GNSS signal spoofing can be easily performed with low-price software-defined radios in the market [6][7]. An example is shown in Fig. 1. When a drone is following its GNSS location-based route and the received GNSS spoofing signal strength is stronger than the real GNSS signal. The drone will pick the stronger signal by default and let the spoofing data affect its navigation system. The drone then may be navigated towards incorrect locations by false GNSS data and falls into the wrong hands.

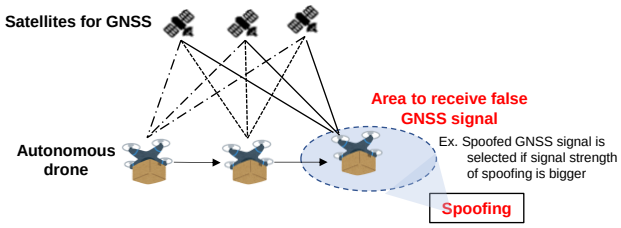


Fig. 1 Spoofing GNSS signals.

2.2 Detection of GNSS Spoofing using Multiple Sensing Devices

In order to maintain correct GNSS information continuously, an important question is how to detect spoofing data. In modern IoT devices, manufacturers usually ship their chips with several sensors and communication modules all-in-one style. We can usually find not only GPS modules but also Bluetooth modules and IMUs, including accelerometers and barometers. These sensors and modules can provide valuable data that potentially confirms the device's current movement status and further validate GNSS status. Fig. 2 shows an example of using a Bluetooth module for its received beacon signals. The beacon's GNSS data can be referenced to validate the drone's GNSS data.

[†]The authors are with Waseda University, Japan.
[††]The author is with Japan Datacom Co., Ltd, Japan.
[a)] E-mail: samqixin@aoni.waseda.jp

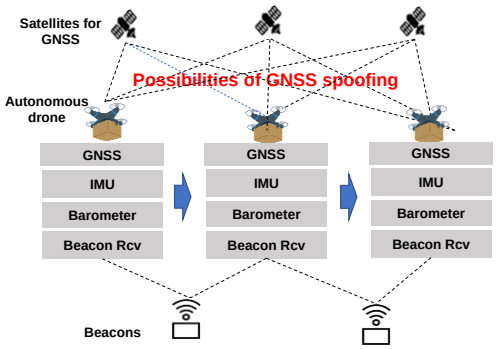


Fig. 2 Detection of GNSS spoofing using multiple sensing devices.

3. Methods

3.1 Experimental System and Data Acquisition

To verify our proposal, an experimental system is designed to detect simulated GNSS spoofing data.

The experimental system is made with several parts, a hand-held compact computer powered by a battery, a GNSS module, and several Bluetooth beacons.

As shown in Fig. 3, the hand-held computer is based on the Raspberry-Pi model 4B. An accelerometer and barometer embedded sensor-hat is attached to the Raspberry-Pi to provide sensor data for GNSS spoofing detection, and a touch-screen display is connected to the Pi to provide real-time data feedback. The GNSS module is connected via USB to provide GNSS data while in the field.

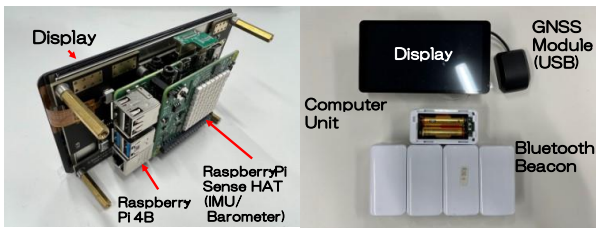


Fig. 3 Experimental system.

A total of five Bluetooth beacons are used in the experiment. The embedded Bluetooth module in the Pi can scan and find the closest beacon and use the beacon's preset location data to detect GNSS spoofing data.

Table 1 shows the acquisition data in the experiment. The GNSS data records the latitude, longitude, and altitude values during the experiment. The Bluetooth receiver scans and collects the Bluetooth beacon in range and records the beacon ID and its RSSI value. The beacon ID points to a table that holds the other beacons' GNSS data, which is preset in the initial process of the experiment. The RSSI value helps to choose the closest beacon to reference its GNSS data. The onboard accelerometer records the real-time acceleration values in the three axes to verify if the device has moved by GNSS data. The barometer records barometric pressure values that also will help verify GNSS data. The data acquisition cycle is one second.

Table 1 Acquisition data

Device	values	Remarks
GNSS	latitude longitude altitude	
Bluetooth Receiver	beacon ID RSSI	The nearest beacon is selected, its location data is obtained
IMU	acceleration X acceleration Y acceleration Z	
Barometer	barometric pressure	

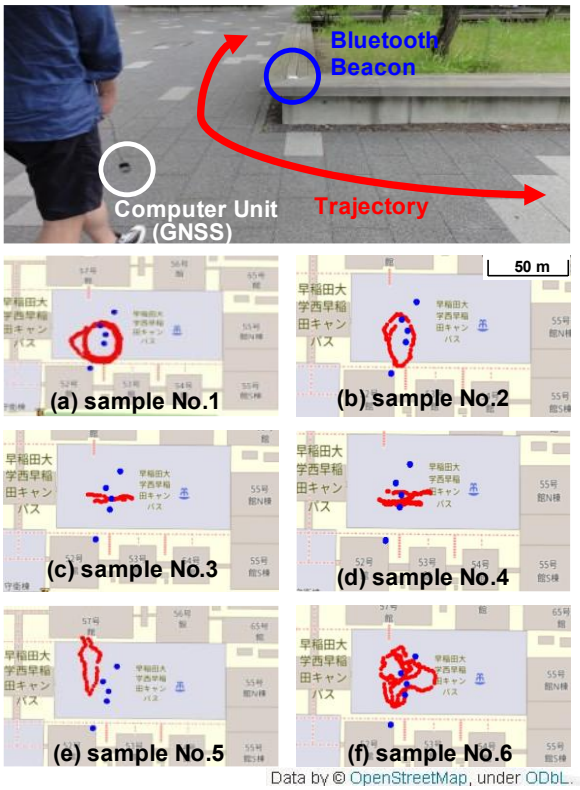


Fig. 4 Acquisition of location data.

The GNSS data is collected in the field of Waseda University. In the experiment, we hand-held the battery-powered computer and moved in different patterns to emulate drone movement. Each movement pattern provides a set of acquisition data. Fig.4 shows collection samples consisting of GNSS location data (red dots) and the positions of Bluetooth beacons (blue dots). As shown in Fig. 4, there are six sets of data categorized into (a)~(f). In data (a) and (b), movement data is in circles and in (c) and (d), it's in straight lines. In (f), it represents random movement. With the collected GNSS data in the field, we can simulate spoofing data by adding malice data in different styles and using acquisition data to detect spoofing data.

3.2 Simulation of GNSS Spoofing

There are many methods of spoofing, ranging from simple to complex implementations [2]. In this study, we first assume that spoofing is an event that replaces false location information in the target area. As shown in Fig. 5, a 5 x 3 radio emission points array is set in our simulation. The distance between the emission points is approximately 5 m, and one of the launch points is activated. In the area centered on the launch point, the GNSS information is replaced by spoofing data. The radius r is set to 10 m in this study.

The spoofing data is designed to maintain continuous movement changing in 9 directions. Latitude and longitude change by 0.000005 degrees per second, which is close to the actual walking movement. Small random perturbations are also added to avoid unnatural movements.

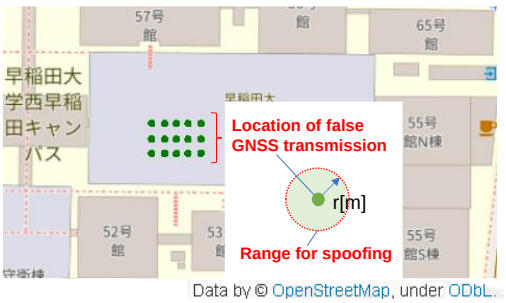
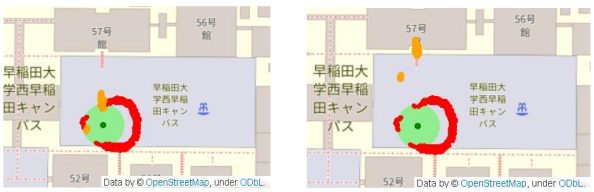


Fig. 5 Areas for spoofing GNSS signals.

By changing emission points and moving directions, the simulation transforms a single acquired walking trajectory into 135 trajectories that include spoofing data. As shown in Fig. 6, two different offsets are set to spoof latitude and longitude. Fig. 6(a) and (b) show the cases of offset = 0 (spoof A) and offset = 25 m (spoof B), respectively. The red dots indicate GNSS positions, and the green circle is the area to be spoofed. The spoofed position data are indicated by orange dots with different amounts of shift.



(a) spoof A (b) spoof B

Fig. 6 Spoofing GNSS data (red dots: GNSS, green area: spoofing area, orange dots: spoofed location)

3.3 Detection of Spoofing

We analyze the properties of features in spoofing data by using a decision tree classifier [8]. Also, the decision tree classifier can be applied to detect spoofing in Section 4.2.

In the first step, features for classification are defined so that they are not related to the absolute location. The distance from the previous position, the distance from the nearest beacon, and acceleration values are selected as features (see Table 2). The two types of distances from the previous position are calculated from latitude and longitude at different time intervals. In this experiment, altitude and barometric pressure are ignored as feature values because the acquired data does not involve enough altitude changes.

The distance from the beacon is obtained by selecting the nearest neighbor beacon, as explained in Section 3.1.

Table 2 Features for spoofing detection

Features	Label
Distance from previous position (1sec)	*dist*
Distance from previous position (2sec)	*dist2*
Distance from the nearest beacon	*distBle*
Acceleration X	*accX*
Acceleration Y	*accY*
Acceleration Z	*accZ*
Difference of Acceleration (1sec)	*diffAcc*

4. Results and Discussion

4.1 Properties of the Trained Decision Tree

We use the first half of data No.1 to create a model of the decision tree classifier. The training dataset consists of 67 sequences containing 18827 data. The maximum depth of the tree is set as five to create the model.

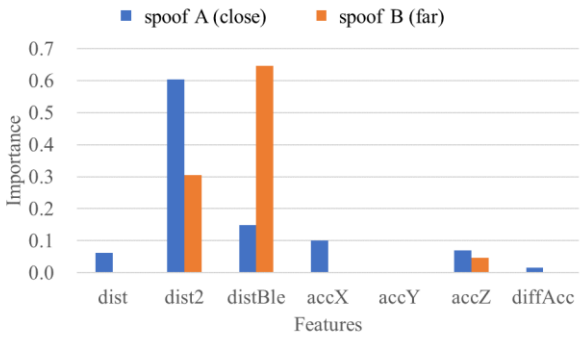


Fig. 7 **Importance of features**

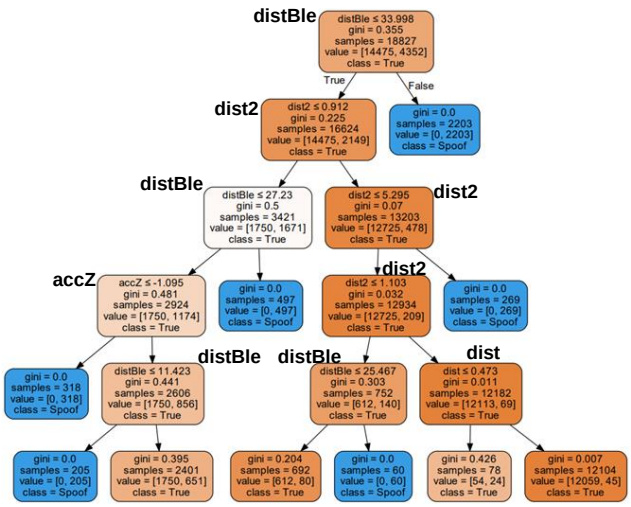


Fig. 8 Decision tree to detect "spoof B" (blue squares means the decision points of spoofing)

The importance values (the Gini importance) for features are shown in Fig. 7. If the model is trained using data "spoof B", the distance from the nearest beacon (*distBle*) becomes a higher importance. On the other hand, if the model is trained with "spoof A", the importance of *distBle* becomes lower as same as other features such as *accX* and *accZ*. This property is reasonable because "spoof B," which includes spoofed locations far from genuine locations, tends to be far from the nearest beacon as well. Fig. 8 shows the structure of the decision tree trained using "spoof B".

The contribution of *dist* is small in the results of "spoof A" and "spoof B". It is considered that amounts of *dist* are very small because the device moved slowly by walking. This feature should be revised in actual drone flight.

4.2 Performances of Spoof Detection

The six types of acquired data shown in Fig. 4 are evaluated with the trained decision trees created in Section 4.1. For the evaluation of dataset No.1, we utilized the second half of the data, which was not used for training. To obtain higher recall, the parameter of the prediction probability is changed from the default value of 0.5 to 0.95. Table 3 shows precision and recall of detecting spoofing for two kinds of spoofing patterns. In the result of "spoof B", the average of recall becomes higher at 0.986, and the average of precision is 0.704.

According to the results in Table 3, the model trained by using the first half of No.1 maintains higher performance for other samples. It is considered that the designed features and the trained classifier have generalization capability for variations in the data. As we pointed out, the generalization of integration of multiple sensing devices is a challenge. As a first step, we use a classic decision tree approach and find that learning classifiers can be one of the solutions for multi-device integration to detect spoofing.

Table 3 Performances of spoofing detection

sample No.	data volume	spoof A(close)			spoof B (far)		
		precision	recall	F1	precision	recall	F1
1	19108	0.651	0.936	0.768	0.683	0.988	0.801
2	8505	0.945	0.902	0.923	0.789	0.986	0.877
3	4995	0.930	0.912	0.920	0.959	0.988	0.974
4	11070	0.865	0.898	0.881	0.775	0.988	0.868
5	8910	0.339	0.821	0.480	0.315	0.978	0.477
6	24705	0.769	0.906	0.832	0.702	0.987	0.821
	average	0.750	0.896	0.800	0.704	0.986	0.803

The simulation results explain that distance from the nearest beacon is used to detect spoofing with large offsets. According to this result, the installation of beacons on the highways [9] is one of the effective approaches to prevent spoofing and provide trusted location information.

Although there are many studies for signal-level spoofing detection [2][3], these approaches demand additional circuits and cannot be applied to current commercial GNSS devices. It is considered that our approach is practical to prevent spoofing by using commercial GNSS devices. It is considered that our approach can be one of the realistic solutions to reduce the threat of spoofing in current systems.

5. Conclusion

In this paper, we investigated possibilities to detect spoofing of commercial GNSS devices using multiple sensing devices and a decision tree classifier. Multiple features from the GNSS, beacons and the IMU are defined and a learning-based model to detect spoofing is created. The learning-based classifier indicates adequate performance and generalization capability. The results also explain that the distance from the nearest beacon is useful for detecting GNSS spoofing and indicate possibilities of installation for future drone highways. We will continue to study further improvements, including other state-of-the-art machine learning approaches.

Acknowledgments

These research results were obtained from the commissioned research (No. 03901) by National Institute of Information and Communications Technology (NICT), Japan.

References

[1]A.J. Kerns, D.P. Shepard, J.A. Bhatti and T.E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing". J Field Robot, vol. 31 no. 4 pp.617–636, 2014.

[2]Z. Wu, Y. Zhang, Y. Yang, C. Liang and R. Liu, "Spoofing and Anti-Spoofing Technologies of Global Navigation Satellite System: A Survey," IEEE Access, vol. 8, pp. 165444-165496, 2020.

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

[4]Y. Liu, S. Li, Q. Fu, Z. Liu and Q. Zhou, "Analysis of Kalman Filter Innovation-Based GNSS Spoofing Detection Method for INS/GNSS Integrated Navigation System," IEEE Sensors Journal, vol. 19, no. 13, pp. 5167-5178, July, 2019.

[5]Y. Gao, Z. Lv and L. Zhang, "Two-Step Trajectory Spoofing Algorithm for Loosely Coupled GNSS/IMU and NIS Sequence Detection," IEEE Access, vol. 7, pp. 96359-96371, 2019.

[6]T.E. Humphreys, B.M. Ledvina, M.L. Psiaki, B.W. O'Hanlon, P.M. Kintner Jr., "Assessing the spoofing Threat: Development of a Portable GPS Civilian Spoofer," Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008), pp. 2314-2325, 2008.

[7]T. Ueki, K. Yoshii, S. Shimamoto, K. Mizuno and K. Matsufuji, "Evaluation of Impact of Intermediate GPS Spoofing to Mobile Terminals," 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), pp. 717-718, 2022.

[8]A. C. Müller, S. Guido, "Introduction to Machine Learning with Python: A Guide for Data Scientists", O'Reilly Media, 2016.

[9]M. Hamnanaka, "Optimum Design for Drone Highway Network," 2019 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 923-929 2019.