engineering proceedings

MDPI

Proceeding Paper

On the Context-Aware GNSS Navigation: Test of a k-Nearest Neighbors Classifier in Different Environments †

Giovanni Cappello [1,*], Antonio Angrisano [2], Ciro Gioia [3], Antonio Maratea [4] and Salvatore Gaglione [1,4]

[1] International PhD Programme, UNESCO Chair "Environment, Resources and Sustainable Development", Department of Science and Technology, University of Naples "Parthenope", Centro Direzionale Isola C4, 80143 Naples, Italy; salvatore.gaglione@uniparthenope.it
[2] Department of Engineering, University of Messina, 98122 Messina, Italy; angrisanoa@unime.it
[3] European Commission, Joint Research Center, 21027 Ispra, Italy; ciro.gioia@ec.europa.eu
[4] Department of Science and Technology, University of Naples "Parthenope", Centro Direzionale Isola C4, 80143 Naples, Italy; antonio.maratea@uniparthenope.it
* Correspondence: giovanni.cappello001@studenti.uniparthenope.it
† Presented at the European Navigation Conference 2025 (ENC 2025), Wrocław, Poland, 21–23 May 2025.

Abstract

GNSS navigation can be challenging in urban environments, especially when low-cost devices are adopted. Among the possible solutions, in more recent years, approaches based on Machine Learning became popular. In this work, features based on geometry, satellite visibility and carrier-to-noise ratio are used in combination with k-Nearest Neighbors classifier to distinguish between open-sky and obstructed environments. The purpose of this research is to develop a reliable context classifier, to evaluate its recognition capabilities in static and dynamic environments and to assess its applicability in real-time positioning. Several performance metrics have been used, i.e., accuracy, precision, recall, F1-score, and multiple tests have been carried out to demonstrate the reliability of such algorithm with validation data. More than 98% of classification accuracy for the static tests has been obtained in average, evidencing the detection capabilities of such an algorithm.

Keywords: GNSS; Machine Learning; adaptive navigation; context-awareness; kNN

1. Introduction

Over the years, Global Navigation Satellite Systems (GNSS) became a fundamental positioning means in different sectors of navigation, e.g., agriculture, automotive, aerospace, maritime, pedestrian, survey and so on. Depending on the application segment, different performance requirements must be met in terms of accuracy, availability, reliability, and continuity [1–3]. The achievement of the sector requirements can be challenging, especially depending on the adopted devices and techniques—on the one hand, professional receivers are equipped with high-quality hardware and advanced software capable of mitigating the presence of outliers or interferences; on the other hand, low-cost receivers are constituted by cost-effective antennas and simpler software, that do not allow achieving the same performance of the abovementioned devices. In personal navigation, users with low-grade devices and smartphones frequently face limitations such as limited number of acquired GNSS, frequencies, and lower measurement quality [4]. Additionally, the navigation in complex environments constitutes another challenge; the presence of buildings and obstacles around the receiver limits the satellite visibility, the signal quality, and the probability of outliers in the measurement set is much higher. Efforts have been made by researchers

Check for updates

Academic Editor: Tomasz Hadas

Published: 11 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

to enhance the positioning performance of low-cost GNSS receivers in signal-degraded environment. Common solutions include the adoption of Fault Detection and Exclusion (FDE) [5] and the integration of GNSS with complementary external sensors [1,2]. In more recent years, Artificial Intelligence (AI) has taken steps forward in multiple scientific sectors, including satellite navigation, inspiring novel and innovative applications in GNSS research. In fact, Machine Learning (ML) has found different use cases in the GNSS domain, i.e., signal acquisition, detection and classification, precise positioning, ionosphere with Neural Networks (NN), k-Nearest Neighbors (kNN) and Support Vector Machine (SVM) being the most popular choices [6]. Among all the possible use cases, context recognition is the specific focus of this paper. Context-awareness in navigation constitutes an essential information to improve the GNSS positioning performance; such information allows applying adaptive strategies at the navigation engine level depending on the context. The basics of context-adaptive integrated navigation are found in [7], providing a framework aimed at categorizing context and user behavior. In [8], the context-awareness was explored on smartphone devices using data collected in different types of environments and features based on Carrier-to-Noise Ratio (CN0), the visibility, and the pseudorange residuals. In [9], the same authors extended the concept by proposing a framework that includes a behavior detector to recognize vehicle or pedestrian navigation, and the adopted classification models were presented in [10]. In [11] an SVM-based context recognition model trained with GNSS signal features with the aim to recognize six different environments is proposed. The main limitation in recognizing more and more scenarios is the decreasing detection accuracy; the more environments one tries to classify, the more the boundary between one context and another tends to become thinner. Naturally, the choice of the features used to recognize the scenario is crucial and, more importantly, the choice of proper statistical indicators applied to the features must be taken carefully, in order to obtain the most descriptive set of instances possible. Also, the choice of the dataset is essential because it must be made in order to avoid the overfitting problem [12,13]. In this work, a kNN is tested to perform an automatic scenario classification. The proposed research is an extension of our past study [14], where a novel set of features has been explored on different grade of devices in open sky and urban canyon environments. These features are now exploited for the first time with a kNN classifier. The performance of the classifier is evaluated in terms of accuracy, precision, recall, and F1-score. The labelled reference dataset includes 4 h of GPS data collected in an open-sky environment and 4 h of GPS data in multiple urban areas. For the test, two static and one kinematic dataset were gathered in order to demonstrate the effectiveness of the kNN classifier with new data and its applicability for real-time positioning purposes.

The paper is structured as follows: in Section 2, the kNN algorithm, the selected features and the evaluation metrics are discussed; in Section 3, the training procedure is described; Section 4 presents the results; finally, Section 5 concludes the paper.

2. Materials and Methods

2.1. k-Nearest Neighbors Classification

The kNN is an ML algorithm that allows classifying data by finding for each test instance the majority label among its closest k labelled instances. The number of k closest instances is a free parameter, an integer number set by the user. The selection of the k nearest instances is based on the calculation of a distance metric which in this study, is the Euclidean one. In detail, given two points (a, b) and considering a total of n features

https://doi.org/10.3390/engproc2026126010

used for data classification $(a_1, a_2, \ldots a_n)$ and $(b_1, b_2, \ldots b_n)$, the Euclidean Distance can be expressed as follows [12]:

$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \quad (1)$$

It is good practice to normalize the data to avoid that a feature dominates the classification process with respect to the other ones. For this purpose, a z-score normalization has been performed as reported in (2). For example, for a vector data $\underline{x}$:

$$\underline{x}^+ = \frac{\underline{x} - \mu}{\sigma} \quad (2)$$

where (μ, σ) indicate the mean and the standard deviation, respectively, of the considered feature.

The main drawback of using kNN is the computational complexity which is $O(nd)$ for each point to be classified, with n being the number of labelled reference instances and d being the number of features [15]. For this study features are normalized, and a binary classifier is used, with label 1 that corresponds to urban canyon and label 0 that corresponds to open sky, the number of neighbours $k = 10$ and the number of features $d = 3$.

2.2. Feature Description

For this study, the features presented in [14] are used to perform a scenario classification. Starting from Positional Dilution of Precision (PDOP) and satellite visibility (SV), a moving window of 10 consecutive samples has been considered for each epoch and the standard deviation in this window has been used as the feature suitable for classification. So, the first feature, f_1, is the standard deviation of PDOP and the second feature, f_2, is the standard deviation of SV, both in a time window of 10 consecutive measurements. The third feature, f_3, is for each epoch the median difference of CN_0 for each couple of visible satellite (ΔCN_0). Their combination has been chosen based on an ablation study.

2.3. Performance Metrics

To evaluate the performance of the kNN classifier and to prevent overfitting, k-fold cross validation is used [12]. The accuracy is derivable from the confusion matrix that reports the number of instances correctly (True Positives and True Negatives) and incorrectly (False Positive and False Negatives) classified. This last consists of a table where the columns represent the classification made by the ML algorithm and the rows represent the ground truth [12], as reported in Table 1.

Table 1. The confusion matrix for a binary classification problem.

	Predicted Positive	**Predicted Negative**
Positive	True Positives (TP)	False Negatives (FN)
Negative	False Positives (FP)	True Negatives (TN)

Given this matrix, several metrics can be calculated to evaluate the performance of the classifier. In particular, the accuracy, a, the precision, p, the recall, r, and the F1-score, F_1, ref. [12] have been calculated as reported in Table 2.

https://doi.org/10.3390/engproc2026126010

Table 2. Classifier performance metrics.

Accuracy	Precision	Recall	F1 Score
$a = \frac{TP+TN}{N}$	$p = \frac{TP}{TP+FP}$	$r = \frac{TP}{TP+FN}$	$F_1 = 2 * \frac{p*r}{p+r}$

Where N is the total number of samples. The recall is also referred as True Positive Rate (TP Rate) and it can be used together with the False Positive Rate (FP Rate) to generate the Receiver Operating Characteristics (ROC) graph that serves in comparing the TP Rate with the FP Rate. A good classifier tends to be placed in the upper zone, toward the point $(0, 1)$, i.e., TP Rate $>$ FP Rate. A classifier placed on the diagonal of the graph has FP Rate $=$ TP Rate and is defined as random classifier. Finally, a classifier located below the diagonal, toward the point $(1,\ 0)$ has a FP Rate $>$ TP Rate. For that kind of algorithm, better performance is obtained reverting the classifications [12].

Finally, to assess the real-time applicability of the proposed kNN model, a processing time analysis is conducted using an Intel Core i7-10750H CPU at 2.60 GHz [16].

3. Experimental Setup and Data

3.1. Reference Data

This section describes how the labelled dataset is built for the two possible target scenarios, urban canyon (labelled as "1") and open sky (labelled as "0"). For this study, a u-Blox ZED F9P multi-GNSS receiver (u-blox (SIX:UBXN), Thalwil, Switzerland) with its standard antenna, manufactured by u-Blox company, based in Thalwil, Switzerland, has been used [17,18].

3.2. Open Sky

In Figure 1, a description of the open sky scenario is reported in terms of PDOP and visibility, in box (a), and in ΔCN_0 in box (b). The dataset includes four hours of GPS data collected at 1 Hz. Looking at the box (a), the open-sky scenario is characterized not only by the high satellite visibility and the favourable geometry, but also by their slow variation over time. Moving the focus on the box (b), it can be noticed how similar the between-satellites C/N_0 values are, indeed most of the ΔCN_0 samples can be found between 0 dB-Hz and 4 dB-Hz.

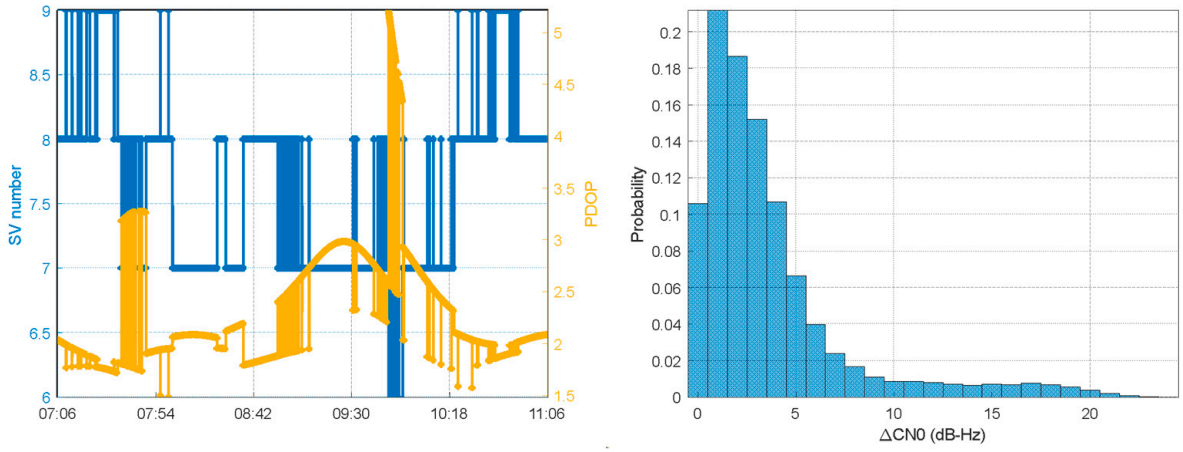


Figure 1. Feature-based description of the open sky environment. (**a**) Time series of satellite visibility (in blue) and PDOP (in yellow); (**b**) distribution of ΔCN_0 values.

So, the first half of the reference dataset is built with 4 h of data (features f1, f2, and f3) gathered in open sky and labelled "0".

3.3. Urban Canyon

Differently from open sky, the urban-canyon context is extremely inhomogeneous due to the variability of number, size, type and arrangement of surrounding obstacles. In this case, a composite dataset has been created concatenating data collected in eight different locations, named from P01 to P08, for a total of 4 h. This is the second half of the reference dataset, labelled as "1".

At each of the eight chosen locations, 30 min of GPS data at 1 Hz were collected in static mode. From the box (a) of Figure 2, the fast variation over the time of the satellite visibility and the PDOP can be noted, together with the different behavior of the single locations. Furthermore, from box (b) of Figure 2, the ΔCN_0 distributions evidence that large between-satellite differences in signal strength can be detected in obstructed areas, assuming values between 10 and 20 dB-Hz, as demonstrated especially in the case of P03 and P06, corresponding to severe urban canyons.

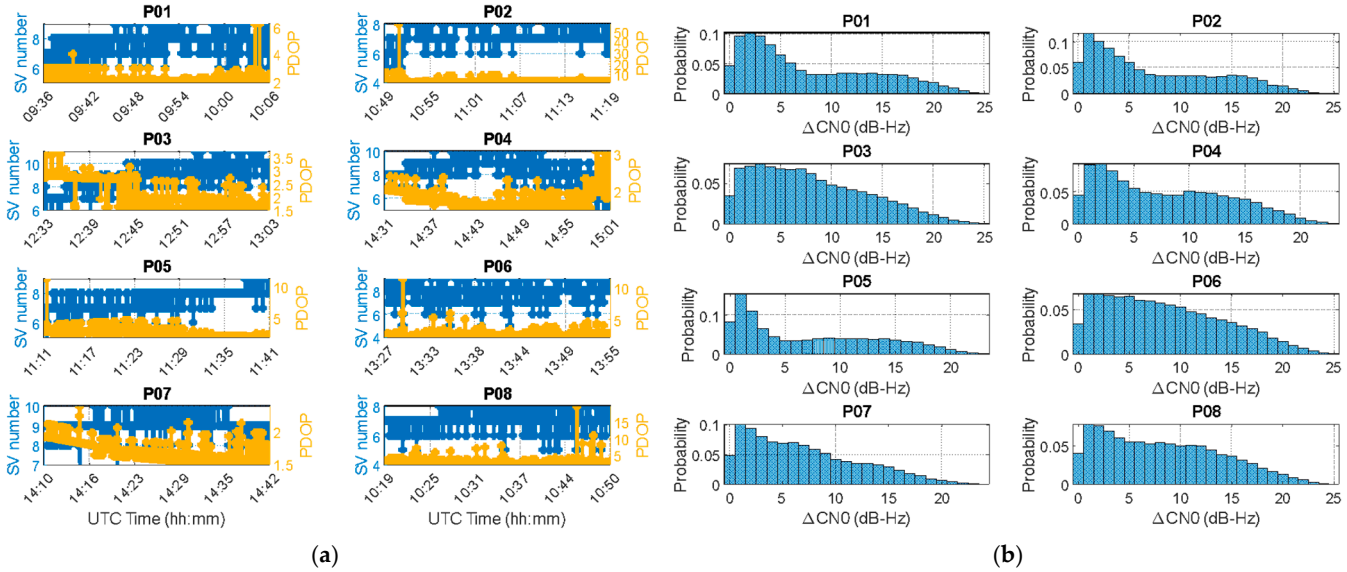


Figure 2. Feature-based description of the urban-canyon environments. (**a**) Time series of PDOP (in yellow) and satellite visibility (in blue) across locations; (**b**) ΔCN_0 distribution per urban site.

The final size of the labelled dataset is [28,769 × 3].

4. Results

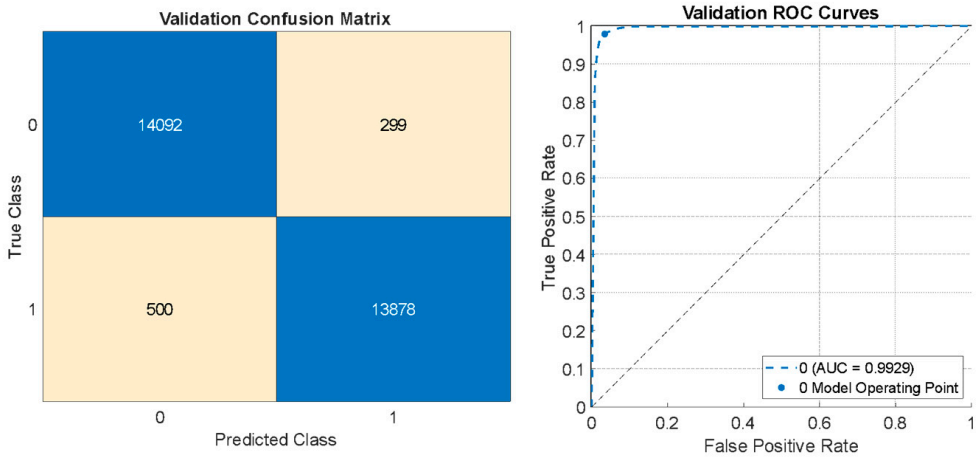
4.1. k-Fold Cross Validation

As a first attempt, classification based on *k*-fold cross validation ($k = 50$) is performed on the reference labeled data and reported in Figure 3. The box (a) represents the confusion matrix with the aggregated TP, TN, FP, FN. As can be seen, only a few hundred of FP and FN, i.e., wrong classifications, are obtained. The potentiality of the kNN is also demonstrated by the ROC curves in the box (b), showing that the Area Under Curve (AUC) is close to 1.

In Table 3, the accuracy, the precision, the recall and the F1-score are reported in percentage. All the values are greater than 96%, demonstrating the viability of the kNN classifier with the chosen reference dataset.

Table 3. KNN performance metrics values in percentage.

Accuracy	Precision	Recall	F1 Score
97.22%	96.57%	97.92%	97.24%



(**a**) (**b**)

Figure 3. Training performance of the KNN classifier. In box (**a**) the confusion matrix describing correct and incorrect classifications is reported. In box (**b**) the ROC curves reporting the True/False Positive Rates for both classes is depicted.

4.2. Static Tests

To validate the performance of the kNN in terms of accuracy and computational overhead, such classifier has been integrated into a Single Point Positioning (SPP) algorithm; in order to evaluate the context information before the solution estimation and how much more time is required to the algorithm to process the GPS data with the addition of the kNN.

For this purpose, two static tests lasting about 2 h each wereperformed, i.e., one in open sky and one in urban canyon.

- To validate the accuracy of the kNN classifier in open sky, 7189 epochs are considered. A total of 7076 correct instances were obtained and only 113 instances were incorrect. Overall, the 98.4% of instances were classified correctly.
- To validate the accuracy of the kNN classifier in urban canyon, 7209 epochs are considered. A total of 7115 correct instances were obtained and only 94 instances were incorrect. Overall, the 98.7% of instances were classified correctly.

To evaluate from a practical point of view the computational overhead due to the introduction of the context classifier in the SPP estimation process, execution times were monitored and reported hereafter. In detail Figure 4 shows the processing time passed from an epoch to the following one, i.e., Δt_k^{k-1} for two versions of the SPP algorithm, i.e., with and without the classification layer and for the two static data collections, i.e., open sky and urban canyon. As can be seen, an increase in the 90th percentile is detected, going from circa 1 ms for the sole classical SPP to circa 6.5 ms when the kNN layer is added, adopting the same hardware (Intel Core i7-10750H CPU). About 45 s in total are needed to process 2 h of data with the addition of the classification layer, compared to about 6 s for the native SPP without the context classifier.

https://doi.org/10.3390/engproc2026126010

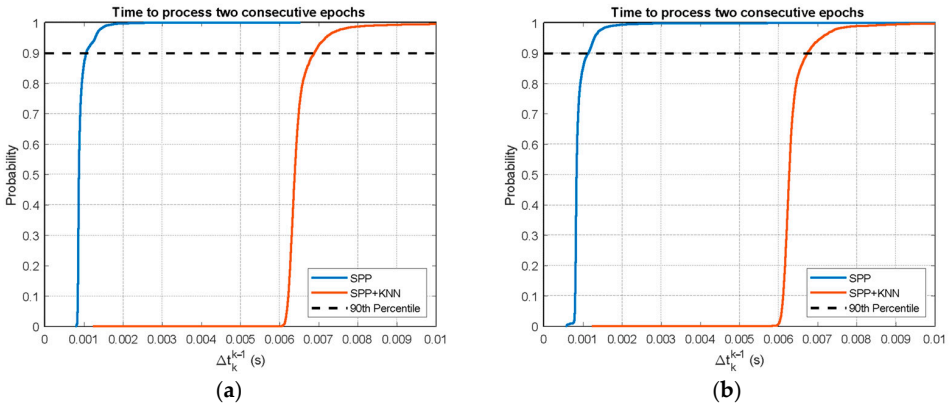


Figure 4. Time to process two consecutive epochs. The orange line indicates the customized algorithm adopting the KNN classification layer; the blue line indicates a classical SPP algorithm. Box (**a**) is for open sky and box (**b**) for urban canyons.

4.3. Dynamic Context Test

Validation in open sky and urban canyon was done with static data collections due to the difficulties of finding true labels for the context in each epoch in a dynamic scenario. Hence this final test is qualitative, it is aimed at verifying the kNN context recognition capabilities in a dynamic scenario only where an a priori knowledge of the environment surrounding the receiver is available. To this aim, the antenna of the chosen receiver was placed on top of a car following for 22 min a mixed trajectory characterized by narrow streets, skyscrapers, trees, and bridges, with a data rate of 1 Hz. The followed path is shown in Figure 5; the full trajectory is showed in the central map while, on the sides, four zoomed-in areas highlight specific segments are shown. The blue dots represent the locations classified as open-sky environments while the yellow ones represent the locations classified as disturbed areas. In total, 1233 instances have been classified. Looking at the enlarged views, it is possible to observe the very good classification capabilities of the KNN based on the chosen features.

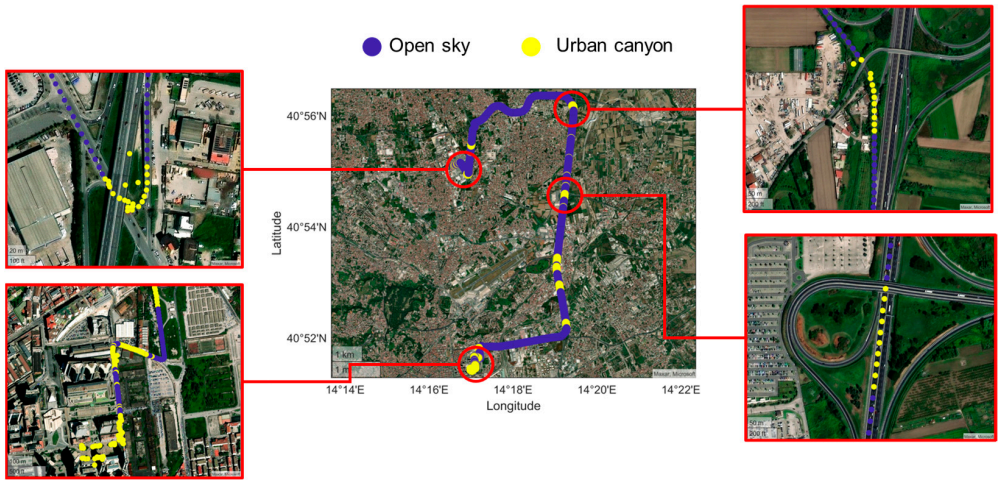


Figure 5. Demonstration of KNN context recognition capabilities in dynamic environment. The blue dots refer to an open sky; the yellow ones refer to an urban canyon. The main trajectory is reported in the central picture.

The total number points classified as open sky is 1006 and 227 for urban canyon. As expected, a higher number of open sky environments is reported, since most of the trajectory is conducted on a highway, typically free from obstacles. However, a substantial number of urban canyon environments were detected, especially at the beginning and at

the end of the test, in compliance with the test plan, including narrow streets, bridges, and tall buildings around the antenna.

Finally, the processing time overhead analysis is reported in Figure 6. The results confirm the higher amount of time needed to process GNSS data in SPP when the classification layer is also activated; an increase in the 90th percentile of Δt_k^{k-1} was detected, passing from about 2.5 ms for the classical SPP to about 12 ms when adopting the kNN.

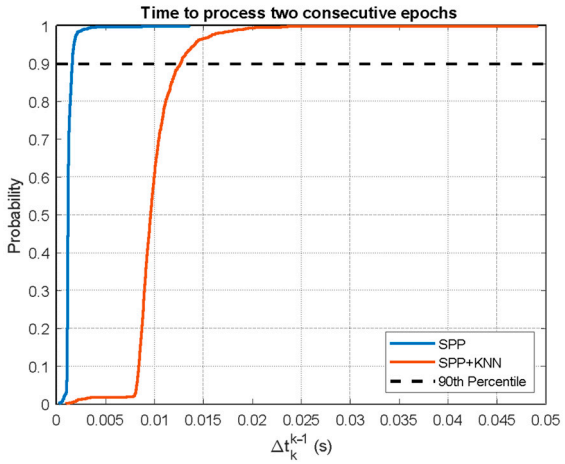


Figure 6. Time to process two consecutive epochs for the dynamic test. The orange line indicates the customized algorithm adopting the KNN classification layer; the blue line indicates a classical SPP algorithm.

5. Conclusions

In this work, a kNN classifier was tested to perform an automatic context recognition with real GNSS data, based on the features introduced in our previous study [14]. To obtain labelled data with sufficient variety and representativeness, several data collections from different contexts have been combined. In detail, 4 h of open-sky and 4 h of urban-canyon data were collected and used as labelled reference for the kNN for binary classification. The validation was conducted with k-fold cross validation in terms of accuracy, precision, recall and F1-score, all resulting higher than 96%. Additionally, three tests were carried out to evaluate the capabilities of the trained ML algorithms with new data; a static test in open sky, a static test in urban canyon, and a qualitative dynamic automotive test in mixed environments were performed. Finally, the ML algorithm was included into a SPP algorithm to check its applicability in real-time navigation and its computational overhead.

From the static tests, it emerged that accuracies higher than 98% were obtained for both cases. On the other hand, the applicability in real-time positioning is challenging, since a strong increase in the processing time was observed, evidencing the need for reducing the size of the reference data to increase the efficiency. In the dynamic test, where a large number of instances had to be classified under varying conditions, only a qualitative analysis was feasible, since no true labels were available. The classifier maintained a consistent recognition performance, even in highly variable environments.

In future studies, dimensionality reduction and pruning techniques will be tested on the reference dataset; also, transition areas will be modelled and the same features will be provided to other ML classification algorithms, characterized by a lower computational complexity.

Author Contributions: G.C., A.A., C.G., A.M. and S.G. contributed equally. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kaplan, E.D.; Hegarty, C.J. *Understanding GPS/GNSS—Principles and Applications*, 3rd ed.; Artech House: Norwood, MA, USA, 2017.
2. Teunissen, P.J.; Montenbruck, O. *Springer Handbook of Global Navigation Satellite Systems*; Springer International Publishing: Cham, Switzerland, 2017.
3. European Union Agency for the Space Programme. EUSPA EO and GNSS Market Report. 2024/Issue 2. Available online: https://www.euspa.europa.eu/sites/default/files/euspa_market_report_2024.pdf (accessed on 24 April 2025).
4. Robustelli, U.; Baiocchi, V.; Pugliano, G. Assessment of dual frequency GNSS observations from a Xiaomi Mi 8 Android smartphone and positioning performance analysis. *Electronics* **2019**, *8*, 91. [CrossRef]
5. Kuusniemi, H. User-Level Reliability and Quality Monitoring in Satellite-Based Personal Navigation. Ph.D. Thesis, Tampere University of Technology, Tampere, Finland, 2005.
6. Siemuri, A.; Kuusniemi, H.; Elmusrati, M.S.; Välisuo, P.; Shamsuzzoha, A. Machine learning utilization in GNSS—Use cases, challenges and future applications. In Proceedings of the 2021 International Conference on Localization and GNSS (ICL-GNSS), Virtually, 1–3 June 2021; pp. 1–6.
7. Groves, P.D.; Martin, H.; Voutsis, K.; Walter, D.; Wang, L. Context detection, categorization and connectivity for advanced adaptive integrated navigation. In Proceedings of the 26th International Technical Meeting of the Satellite Division of the institute of navigation (ION GNSS+ 2013), Nashville, TN, USA, 16–20 September 2013.
8. Gao, H.; Groves, P.D. Environmental context detection for adaptive navigation using GNSS measurements from a smartphone. *Navig. J. Inst. Navig.* **2018**, *65*, 99–116. [CrossRef]
9. Gao, H.; Groves, P.D. Improving environment detection by behavior association for context-adaptive navigation. *Navig. J. Inst. Navig.* **2020**, *67*, 43–60. [CrossRef]
10. Gao, H. Investigation of Context Determination for Advanced Navigation using Smartphone Sensors. Ph.D. Thesis, UCL (University College London), London, UK, 2019.
11. Wang, Y.; Liu, P.; Liu, Q.; Adeel, M.; Qian, J.; Jin, X.; Ying, R. Urban environment recognition based on the GNSS signal characteristics. *Navigation* **2019**, *66*, 211–225. [CrossRef]
12. Bramer, M. *Principles of Data Mining*, III ed.; Springer: London, UK, 2016.
13. Inside GNSS. What Are the Roles of Artificial Intelligence and Machine Learning in GNSS Positioning? 2020. Available online: https://insidegnss.com/what-are-the-roles-of-artificial-intelligence-and-machine-learning-in-gnss-positioning/ (accessed on 24 April 2025).
14. Cappello, G.; Maratea, A.; Gioia, C.; Angrisano, A.; Del Pizzo, S.; Gaglione, S. Toward Context-Aware GNSS Positioning: A Preliminary Analysis. *Eng. Proc.* **2025**, *88*, 14.
15. Deng, Z.; Zhu, X.; Cheng, D.; Zong, M.; Zhang, S. Efficient kNN classification algorithm for big data. *Neurocomputing* **2016**, *195*, 143–148. [CrossRef]
16. Intel. Intel Core i7-10750H Processor. Available online: https://www.intel.com/content/www/us/en/products/sku/201837/intel-core-i710750h-processor-12m-cache-up-to-5-00-ghz/specifications.html (accessed on 12 May 2024).
17. u-blox. Product Summary—ZED-F9P Series. 2024. Available online: https://content.u-blox.com/sites/default/files/ZED-F9P_ProductSummary_UBX-17005151.pdf (accessed on 24 April 2024).
18. u-blox. ANN-MB Series—L1/L2 Multi-Band, High Precision GNSS Antennas. Available online: https://www.u-blox.com/en/product/ann-mb-series (accessed on 3 April 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.