IUST
Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

GPS Spoofing Detection Based on Variance and Singular Values Analysis of Cross Ambiguity Function and Machine Learning Algorithms

M. J. Jahantab*, S. Tohidi*, M. R. Mosavi*(C.A.) and D. M. De Andrés**

Abstract: Global Positioning System (GPS) spoofing poses serious threats to navigation systems, as it transmits false GPS signals that cause receivers to compute incorrect positions. To address this issue, our research in this study focused on leveraging the Cross-Ambiguity Function (CAF) along with advanced machine learning techniques to effectively detect spoofing attacks. A further challenge in using CAF for spoofing detection is its high dimensionality, which demands powerful hardware and considerably slows down the detection process. Detecting spoofing signals with delays of less than 0.5 chips relative to the authentic signal is particularly difficult. To overcome this, the SVD_Var dimensionality reduction algorithm, which leverages the variance of CAF data through Singular Value Decomposition (SVD), is proposed to enhance both speed and detection performance. The reduced-dimensionality data are subsequently used to train a basic Multi-Layer Perceptron (MLP) neural network and the k-Nearest Neighbors (kNN) algorithm. The effectiveness of the proposed method is validated using the widely recognized Texas Spoofing Test Battery (TEXBAT) dataset. Results indicate that the method achieves an average detection rate exceeding 80% across various TEXBAT scenarios, demonstrating enhanced sensitivity and robustness in spoofing detection compared to both traditional and state-of-the-art approaches. Also, this approach accomplishes a dimensionality reduction ranging from 99.69% to 99.99% in terms of the number of pixels which significantly accelerates the processing speed.

Keywords: GPS, Spoofing Detection, CAF, SVD, MLP, kNN, Dimension Reduction Algorithm.

1 Introduction

FOR the effective functioning and deployment of a multitude of contemporary applications, including intelligent transportation systems and location-based services, a seamless and accurate supply of navigation, positioning, and timing data is indispensable. In this context, Global Navigation Satellite Systems (GNSS) stand as the foremost provider of critical information, thereby establishing the foundational infrastructure for all Positioning, Navigation, and Timing (PNT) requirements, contingent upon their availability [1-6]. A GPS receiver measures the time delay of signals received from satellites to calculate its distance from each one. Utilizing distance measurements from a minimum of four satellites, the receiver accurately determines its three-dimensional position in space [7,8]. Among the various types of attacks, spoofing is considered the most dangerous form of intentional interference with GPS. In this type of attack,

Iranian Journal of Electrical & Electronic Engineering, YYYY.
Paper first received DD MONTH YYYY and accepted DD MONTH YYYY.
* The authors are with the School of Electrical Engineering, Iran University of Science and Technology (IUST), Narmak, Tehran 16846-13114, Iran.

** Department of Computer Science, Universidad de Valladolid.
E-mails: jahantab_m77@elec.iust.ac.ir, s_tohidi@alumni.iust.ac.ir, m_mosavi@iust.ac.ir, and diego.martin.andres@uva.es.
Corresponding Author: M. R. Mosavi

the GPS receiver is deceived by tracking counterfeit signals. Spoofing poses a greater risk compared to jamming due to the receiver's inability to recognize the attack. Essentially, spoofing is a stealthy form of assault wherein the attacker deceives the receiver's spatial and temporal data by transmitting counterfeit signals that closely mimic genuine ones. Research examining various GPS receivers' reactions to spoofing threats demonstrates that these attacks can significantly compromise the accuracy of receiver measurements [9-11].

In this study, we employ spoofing detection leveraging the Cross-Ambiguity Function (CAF) within the receiver's acquisition unit—a method initially introduced by Borhani [12]. The CAF is a two-dimensional function that measures the correlation between the received signal and a locally generated code replica across all possible delay/Doppler pairs, and is maximized to detect valid signals. During the acquisition phase, the presence of spoofing interference can be identified by detecting correlation peaks that exceed a predefined threshold. However, a major challenge in using CAF for spoofing detection lies in its high dimensionality, which demands significant computational resources and results in slow detection speeds. Furthermore, identifying spoofed signals with delays of less than 0.5 chips relative to the authentic signal is particularly difficult. To address these limitations, this paper proposes an algorithm that analyzes the dispersion pattern of CAF data using singular values from SVD and peak maximization, thereby improving both the performance and the speed of the spoofing identification process. The overall approach of the proposed idea is presented in Fig. 1.

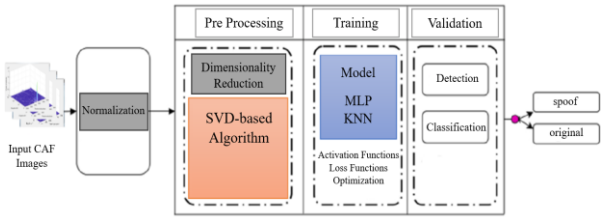


Fig. 1 Overall Approach to Spoofing Detection.

As shown in Fig. 1, the initial phase involves gathering the dataset, comprising both authentic and spoofed signals. Subsequently, following data normalization, the relevant parameters are extracted for signal classification (such as phase, Doppler frequency, variance, signal peak, etc.) are extracted so that dimensionality reduction or detection algorithms can be applied to them. The final stage consists of implementing machine learning techniques, wherein the model is trained and tested using the collected dataset. Consequently, the model categorizes the input signals as either authentic or spoofed based on the extracted classification parameters.

The structure of the rest of the paper is as follows: Section 2 provides an overview of recent developments in spoofing detection and mitigation strategies. Section 3 introduces the spoofing signal model. Section 4 outlines the data collection process. Section 5 presents the proposed detection method, while Section 6 details the machine learning framework employed. The experimental results are discussed in Section 7, and Section 8 concludes the paper.

2 Related Works

Spoofing detection techniques for GNSS signals can be broadly classified into four principal categories: (1) signal processing–based methods, (2) data bit–level approaches, (3) position-based techniques, and (4) methods utilizing machine learning and deep learning models.

Within the signal processing category, various techniques such as correlation peak analysis, antenna array processing, and signal power monitoring are commonly applied. Important examples include the use of SQM-based methods and phase differences between authentic and spoofed signals [13–15], detection with the KNN algorithm [13], utilization of GAN networks with over 98% accuracy for phase differences greater than 0.5 chips [14], and spoofed correlation peak cancellation methods [15]. Power monitoring is also highly effective against meaconing attacks [16], which involve rebroadcasting authentic signals with higher power. In addition, C/N_0-based methods [17-20] and their improved versions, combined with pseudo-ranges and correlation distortion functions [19,20,21], have achieved up to 98% detection rates in scenarios where the spoofed signal power is only a few dB higher than that of the authentic signal.

In data-bit-based methods, the main focus is on Time of Arrival (TOA) and Direction of Arrival (DOA) of signals. The differences in the spoofed signals' TOA compared to authentic signals [22,23], as well as discrepancies in their DOA compared to satellite signals [24], are reliable indicators for spoofing detection. Simulations have shown that this approach performs satisfactorily even in multi-source scenarios.

Machine learning techniques can be integrated with traditional observational parameters and implemented using software-defined radios. A variety of datasets have been employed in machine learning research, which may be either publicly accessible (verifiable) or private (restricted from sharing with other researchers). Among the publicly available datasets, TEXBAT and OAKBAT are the most widely recognized for spoofing scenarios and detection studies. Numerous publications utilize these datasets for the validation and verification of their proposed methodologies. Generally, datasets can be divided into three main categories [25]: (1) real data: raw data from smartphones, GNSS stations, and receivers, (2)

simulated data: generated by software-defined radios and receivers, such as simulators, and (3) hybrid datasets: a combination of simulated and real data, which is the most common case.

Among such studies in deep learning methods, references [21,26] employed Multi-Layer Perceptron (MLP) neural networks for spoofing detection. In [12], the CAF was processed as an image, and spoofing interference was detected using a Convolutional Neural Network (CNN). Similarly, [13] investigated the CAF within the receiver acquisition unit to detect spoofing signals with delays of less than two chips. In this approach, a CNN was used to analyze the merged peaks of spoofed and authentic signals, thereby identifying spoofing attacks. Reference [27] estimated the clock bias using an MLP neural network and detected spoofing attacks by comparing the estimated value with the measured one. In [28], the authors conducted a comparative analysis between various unsupervised and supervised models, evaluating them across several metrics, including memory consumption, prediction latency, training duration, processing time, false alarm rate, spoofing detection rate, detection probability, and accuracy. The findings indicated that classification and regression trees outperformed other various unsupervised and supervised approaches in both classifying and detecting GPS spoofing attacks. The authors in [29], performed a K-fold analysis to select the best machine learning algorithm among several candidates. Their results demonstrated that the polynomial-kernel Support Vector Machine (SVM) outperformed other methods. On the other hand, the findings in [30] indicated that decision tree–based algorithms achieve better results than SVM, KNN, and other analyzed methods. In [31], the authors proposed GNSS spoofing detection using an SVM-based machine learning approach, combining real and simulated datasets for validation and verification. Similarly, [32] employed deep neural networks for spoofing attack detection, using CAF images for spoofing signal recognition. They implemented a data-driven classifier through image segmentation and also considered a Gaussian mixture model to estimate the number of spoofing signals. However, the proposed approach involved the use of multiple neural network models, which significantly increased its computational complexity. Despite this, the method demonstrated a notably high spoofing detection success rate compared to previously established techniques, especially under moderate to high Signal-to-Noise Ratio (SNR) conditions (36 dBm-45 dBm). In [33], the authors used a multilayer perceptron neural network for classifying artificial spoofing scenarios based on selected features and hybrid datasets. Reference [21] also applied an MLP neural network model enhanced through Particle Swarm Optimization (PSO) to improve spoofing detection performance. This method demonstrated accuracy improvements of 4 % and 2% compared to the optimal Bayesian rule-based classifier and the multi-hypothesis Bayesian classifier, respectively. The features employed for classification included signal power and correlation distortion functions, which allowed classification of signals into multi-path, jammed, spoofed, or interference-free categories.

3 Signal Model

In the process of navigation data extraction, the first step in digital signal processing section is acquisition. The purpose of this stage is to identify the observable satellites and to obtain coarse estimates of the code phase and carrier frequency in order to generate a local replica signal and remove the input components. Satellites are distinguished by 32 different Pseudo-Random Noise (PRN) codes. The Doppler effect caused by satellite motion relative to the receiver can shift the carrier frequency by up to ±10 kHz from its nominal value; therefore, estimating the frequency offset is essential. Acquisition is typically performed using one of three methods: (1) serial search, (2) parallel search, or (3) parallel code-space search [7, 34]. The discrete-time signal observed by the receiver, as expressed in Eq. (1), results from down-conversion and sampling (at a rate of $fs=1/T_s$) of the combined signals transmitted from M satellites, along with additive noise.

$$y[n] = \sum_{i=1}^{M} x_i[n;\theta_i] + \eta[n],$$

$$x_i[n;\theta_i] = \alpha_i b_i(nT_s - \tau_i)c_i(nT_s - \tau_i)e^{j2\pi f_d nT_s + j\phi_i} \quad (1)$$

The signal received from the i^{th} satellite is defined by several key parameters: data bits $b_i(\cdot)$, amplitude α_i, spreading code $c_i(\cdot)$, carrier phase term ϕ_i introduced by the channel, Doppler frequency f_d, variable time delay τ_i, and complex Gaussian noise $\eta[n]$ (with zero mean and variance $\sigma 2$). For brevity, the signal parameters are represented in the vector $\theta_i=(\alpha_i,\phi_i,\tau_i,f_{d,i})^\top$.

Signal acquisition is the first step of the receiver to determine the observable satellites and involves searching the code-delay and Doppler frequency space to find values that match the incoming signal. After acquisition, the tracking stage is performed to obtain more accurate estimates of τ_i and $f_{d,i}$. A spoofing transmitter is an interference source that generates GNSS-like signals with the intent of manipulating the estimated time and position of the victim receiver. The spoofer transmits a set of spoofing signals that closely replicate legitimate satellite transmissions, differing only in specific parameters. These subtle discrepancies, if left undetected, can mislead the receiver and result in an incorrect position estimate. Therefore, the GNSS signal received under a spoofing attack can be expressed as in Eq. (2):

$$y[n] = \sum_{i=1}^{M} x_i[n;\theta_i] + \sum_{j=1}^{M_s} x_j[n;\theta_{s,j}] + \eta[n] \quad (2)$$

where M_s refers to the number of spoofing signals. For a spoofing attack to succeed, every spoofing waveform must adopt the same spreading code c_i as the satellite it aims to substitute and must carry a legitimate navigation message b_i. The spoofers' parameters (namely spoofing carrier-phase, code-phase, and amplitude appearing in Eq. (2)) are grouped into $\theta_{s,j}$ for the j^{th} spoofer. Detection of spoofing is formulated as a hypothesis test between $\mathcal{H}_0$ and $\mathcal{H}_1$ as:

$$\begin{cases} \mathcal{H}_0 \ : y[n] = \sum_{i=1}^{M} x_i[n;\theta_i] + \eta[n] \\ \mathcal{H}_1 \ : y[n] = \sum_{i=1}^{M} x_i[n;\theta_i] + \sum_{i=1}^{M_s} x_i[n;\theta_{s,i}] + \eta[n] \end{cases} \quad (3)$$

The CAF is a function that depends on the local replica's time delay τ and Doppler frequency f_d. Typically, the CAF computed across a discrete search space, which consists of a grid of cells corresponding to various combinations of Doppler frequency and signal delay, which are respectively denoted by the vectors $\tau \in R^{n_\tau}$ and $f_d \in R^{n_f}$. In most practical implementations, the number of delay samples n_τ significantly exceeds the number of Doppler frequency values n_f, resulting in a search space that is denser along the delay axis [32].

The effect of a spoofing signal on the CAF is illustrated in Fig. 2. Figure 2(a) shows an example CAF under the hypothesis H0, whereas Fig. 2(b) depicts the scenario when a spoofing signal is present, resulting in a secondary peak in the CAF. In this paper, a variance-based approach is proposed, which is trained on simple MLP neural network and kNN networks as data-driven models to discriminate between received spoofed signals and clean signals.

4 Collecting Data

In this paper, the TEXBAT dataset (scenarios 4 and 7) as well as two simulated datasets with 4-second and 6-second delays were employed. The main idea behind generating delayed datasets is the combination of the authentic GPS signal with its delayed replica. The transmitted GPS L1 signal from satellites can be expressed using Eq. (4) [35]:

$$S_{L1}(t) = A_p P_i(t) W(t) D_i(t) \cos(\omega_{L1}(t+\Delta t) + \varphi_{L1}) + A_c C_i(t) D_i(t) \sin(\omega_{L1}(t+\Delta t) + \varphi_{L1}) \quad (4)$$

The spoofed (delayed) signal generated by the spoofer can be modeled as Eq. (5):

$$S_{L1,C/A}(t) = A_c C_i(t) D_i(t) \sin(\omega_{L1}(t+\Delta t) + \varphi_{L1}) \quad (5)$$

Accordingly, the final received signal at the receiver is a combination of the authentic GPS signal and the spoofed delayed signal as Eq. (6):

$$C_{L1,C/A}(t) = A_C^A C_i^A(t_A) D_i^A(t_A) \sin(\omega_{L1}(t_A + \Delta t_A) + \varphi_{L1}) + A_C^D C_i^D(t_D) D_i^D(t_D) \sin(\omega_{L1}(t_D + \Delta t_D) + \varphi_{L1}) \quad (6)$$

In the Eq. (6), the superscript A denotes the authentic signal, while the superscript and subscript D represent the delayed signal. In fact, Eq. (6) describes the final signal that the target receiver processes in order to extract its spatial and temporal coordinates.

According to the explanations given above, the spoofing data with delays of 4 and 6 seconds are generated as shown in Fig. 3. The data storage method is as follows: the file original contains the information from the primary source. This represents a valid GPS signal. The file spoof contains the information from the secondary source, which is obtained from the simulator. In other words, the goal is to combine the information of one location point with different delays.

In the file 4Sec, only the first source is active for about 4 seconds, after which the second source is turned on. In the file 6Sec, only the first source is active for about 6 seconds, and then the second source is activated.

The spoofed signal for the delayed datasets of 6 seconds and 4 seconds can be expressed by equations (7) and (8), respectively, where t denotes time, SA represents the authentic signal, and SS denotes the spoofing signal.

$$S_S(\mathrm{t}) = S_A(\mathrm{t}) + S_A(\mathrm{t}-6) \quad (7)$$

$$S'_S(\mathrm{t}) = S'_A(\mathrm{t}) + S'_A(\mathrm{t}-4) \quad (8)$$

The results of the acquisition process in terms of the number of acquired satellites for the 6Sec and 4Sec data are presented in Fig. 4. The tracking results for the spoofed signal with 4-second and 6-second delays are shown in Fig. 5(a) and Fig. 5(b), respectively. As shown in Fig. 5, the tracking results for the spoofed signals with 4-second and 6-second delays show clear differences in signal quality. For the 4-second signal, the discrete-time scatter plot indicates that points are concentrated along the I-axis, suggesting that Phase Lock Loop (PLL) lock is maintained but with higher noise. Navigation bits are still identifiable, though noisier, and correlation results are more variable, reflecting lower SNR, yet Delay Lock Loop (DLL) remains locked. The raw PLL discriminator fluctuates more, and the filtered PLL output shows that the PLL operates under noisier conditions. Similarly, the raw DLL discriminator is noisier, and the filtered DLL shows larger error amplitude, indicating that code tracking is maintained but with reduced quality.

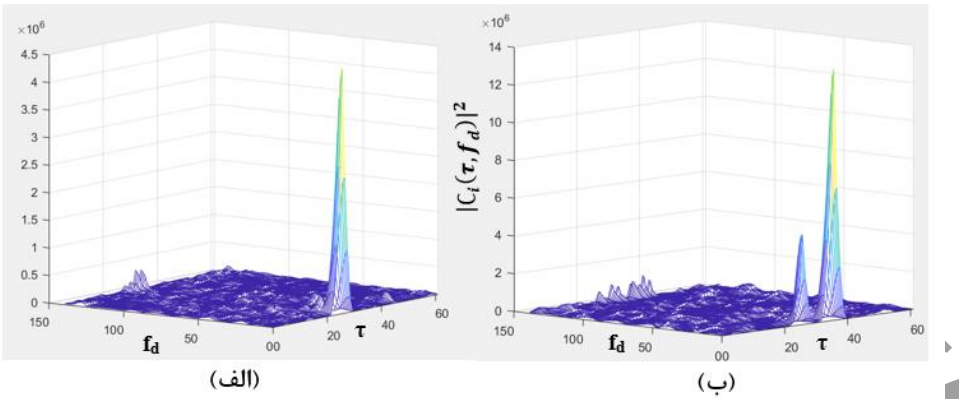


Fig. 2 Evaluation of the CAF over the delay/Doppler frequency grid with C/N0=25–30 dB-Hz and a delay of less than 0.75 chips for: (a) the hypothesis H_0, and (b) the hypothesis H_1.

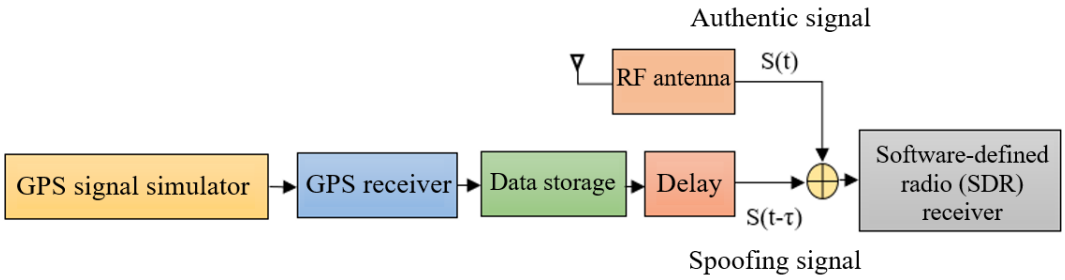


Fig. 3 Block diagram of delayed spoofing signal generation.

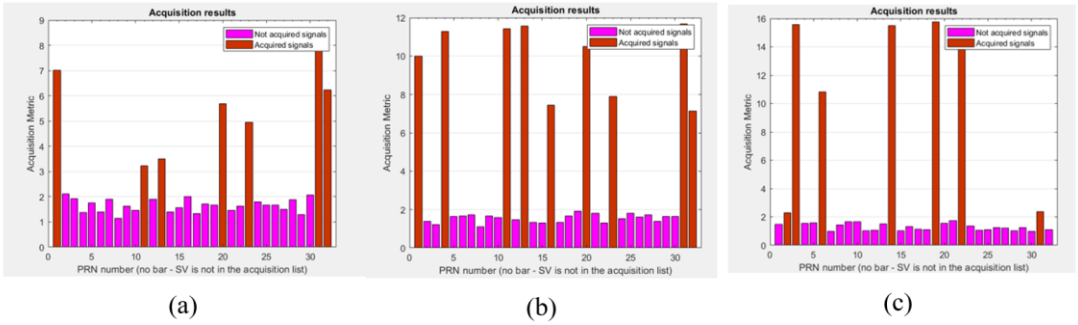
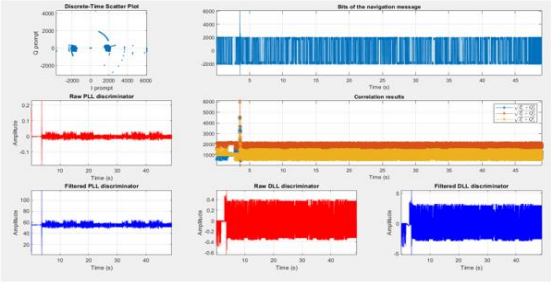


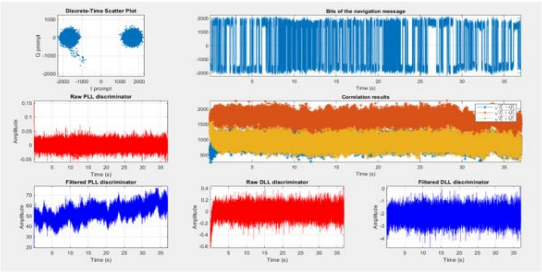
Fig. 4 Acquisition level of the signal: (a) authentic, (b) spoofed 6sec, and (c) spoofed 4sec.

In contrast, the 6-second signal shows two distinct clusters in the scatter plot corresponding to the 0 and 1 bits, reflecting proper carrier PLL, while the navigation message bits are clearly demodulated and separable. Correlation results show large initial peaks for successful tracking. The raw and filtered PLL discriminators fluctuate near zero with smooth behavior, indicating stable PLL lock, and the raw and filtered DLL discriminators confirm accurate and stable code tracking.

The TEXBAT dataset includes eight different spoofing scenarios, six using a fixed antenna and two using a mobile antenna, as well as two clean reference scenarios. A small constant offset between the code range rates and carrier rates is observed in scenarios one through four, as shown in Table 1. This discrepancy may result from a uniform shift applied to all replayed carrier frequencies, as was done to generate scenarios 1–4. In this study, we used Scenario 4 and 7 from this dataset.



(a)



(b)

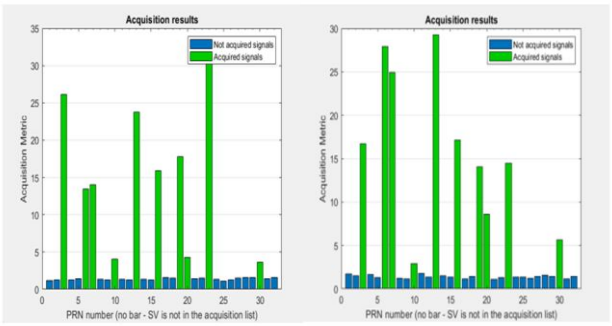
Fig. 5 Tracking results for the spoofed signal: (a) 4 seconds, and (b) 6 seconds.

Table 1 Sample and time offsets in static scenarios of TEXBAT [36].

Scenario	Sample offset from clean data	Time offset (seconds)
1	62,561,438	2.50245752
2	74,922,938	2.99691752
3	55,083,021	2.20332084
4	69,344,725	2.77378900
7	0	0
8	0	0

Scenario 4 is similar to Scenario 3, with the difference that the power advantage of the spoofer has further decreased (from 1.2 dB to 0.4 dB) and the spoofing affects the receiver's position rather than time. The fluctuations observed in the band power indicate that the spoofing signals are in phase, such that their constructive and destructive interactions with the authentic signals tend to occur simultaneously. The spoofing appears from approximately the 100th second onward. The results of the acquisition process in terms of the number of acquired satellites for the TEXBAT ds4 data are shown in Fig. 6.

The software-defined spoofer was improved from 1-bit to 2-bit output, reducing quantization noise, a Doppler frequency error was corrected, and a high-quality digital combination method was developed to merge authentic GNSS recordings with spoofer outputs. These enhancements enabled more realistic spoofing scenarios in datasets such as ds7.bin.



(a) (b)

Fig. 6 Acquisition level of the TEXBAT data signal: (a) authentic, and (b) spoofed.

In ds7.bin, the first 110 seconds contain only authentic signals. From 110–130 s, spoofing signals are gradually introduced with carrier phase alignment, achieving constant amplitude while smoothly shifting phase until they become twice as strong and antipodally aligned with authentic signals. Between 130–150 s, the receiver is fully under spoofer control, though its observables remain similar to authentic data except for a π phase offset, creating potential for undetectable data bit manipulation. From 150–400 s, the spoofer maintains frequency lock while gradually inducing a code phase drift at 1.2 m/s, leading to a clock offset. Finally, from 400–468 s, amplitude and phase remain fixed while the code phase drift continues, producing a total offset of 381.6 m (1.273 µs). Overall, the upgraded testbed and ds7.bin scenario demonstrate sophisticated, seamless spoofing attacks capable of covertly capturing a receiver's tracking loops and gradually biasing its timing [37]. A summary of the datasets used in this paper is presented in Table 2.

5 Method

CAF images are typically high-dimensional, which makes machine learning based spoofing detection computationally intensive and time-consuming. Beyond achieving high detection accuracy, minimizing the detection latency of spoofing signals by neural networks is critical. Therefore, a dimensionality reduction approach that preserves the essential features of CAF images is necessary to reduce processing time while maintaining accurate detection.

The SVD_VAR algorithm is designed to reduce image data into vectors using SVD and variance analysis. Its aim is to represent high-dimensional image data in a more compact form while preserving as much of the original data variance as possible. The reduction process leverages the features of SVD, a mathematical technique commonly employed in data reduction and dimensionality reduction tasks. In large datasets, such as those used in image processing, high-dimensional data can be

computationally expensive and challenging to handle. The goal of SVD_VAR is to reduce these dimensions while preserving a significant portion of the variance (or information) present in the original data. This transformation enables efficient storage, transfer, and analysis of the data, while minimizing the loss of critical information.

The steps of the proposed algorithm are as follows:

Step 1: Data input and loading

Assume X is the input image matrix loaded from the dataset file. The data structure is defined such that $X \in \mathbb{R}^{m\times n}$.

Step 2: Singular value decomposition

After loading the image matrix X, singular value decomposition is applied:

$$X = USV^T \tag{9}$$

where $U \in \mathbb{R}^{m\times m}$ and $V^T \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $S \in \mathbb{R}^{m\times n}$ is a diagonal matrix containing the singular values s_i, which quantify the significance of each corresponding dimension.

Step 3: Computing the explained variance for S

The singular values s_i represent the amount of variance explained by each dimension of the original data. The larger the singular value, the more information (or variance) that component accounts for. To capture this, the algorithm computes the proportion of variance explained by each singular value as follows:

$$\text{Explained Variance} = \frac{\{s_i^K\}^2}{\sum_{j=1}^{r}\{s_j^k\}^2}; r = min(m,n) \tag{10}$$

where $\{s_i^K\}^2$ denotes the squared singular values after retaining K components. Equation (10) normalizes the squared singular values and provides the fraction of the total variance preserved by each component. This ratio determines how much of the total data variance is retained by the selected singular components.

Step 4: Peak representative of each row

As noted, $X \in \mathbb{R}^{m\times n}$, and in accordance with Step 3 For each row i, the preserved variance is given as:

$$v = (v_1, v_2, \dots, v_m)^T \in \mathbb{R}^{m\times 1} \tag{11}$$

For each row i, the maximum element (peak) is selected:

$$p_i = \underset{1\le j\le n}{max} X_{ij}, i = 1,2,\dots,m. \tag{12}$$

which yields the peak representative vector $p = (p_1, p_2, \dots, p_m)^T \in \mathbb{R}^{m\times 1}$. The final output vector is obtained by element-wise multiplication of the two vectors:

$$y = p \odot v \tag{13}$$

where $\odot$ denotes the Hadamard (element-wise) product. Thus, the resulting output vector is $y \in \mathbb{R}^{m\times 1}$.

Step 5: Saving the results

For each image and .mat file, the algorithm applies the preserved variance computation using the singular values and saves the resulting data in a new .mat file. This produces a compact representation of the original image data. If the input data has dimensions m×n, the output data will be a 1×m vector, as illustrated in Fig. 7.

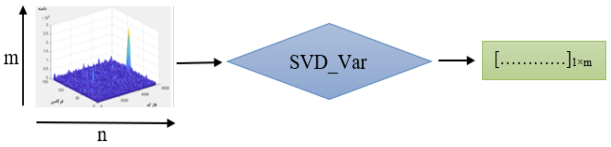


Fig. 7 Operation of the SVD_VAR method.

Table 2 Details of the datasets utilized in this article.

Dataset name	Quantity	PRNs	Dimensions
6Sec	1713 Authentic+ 1794 Spoof = 3507	1, 11, 13, 20, 23, 31, 32	141×5714
4Sec	470 Authentic + 1280 Spoof = 1750	3, 6, 14, 19, 22	141×4092
TEXBAT ds4	2100 Authentic + 2100 Spoof = 4200	3, 6, 7, 13, 16, 19, 23	41×25000
TEXBAT ds7	7536 Authentic + 7536 Spoof = 15072	3, 6, 7,10, 13, 16, 19, 23	41×25000

The diagram of this method is also shown in Fig. 8. The explained variance histogram plots for the 6sec, 4sec, and TEXBAT datasets are presented in figures 9, 10, and 11, respectively.

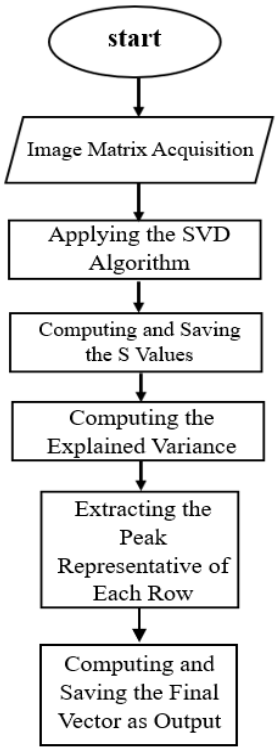


Fig. 8 Diagram of the SVD_VAR algorithm.

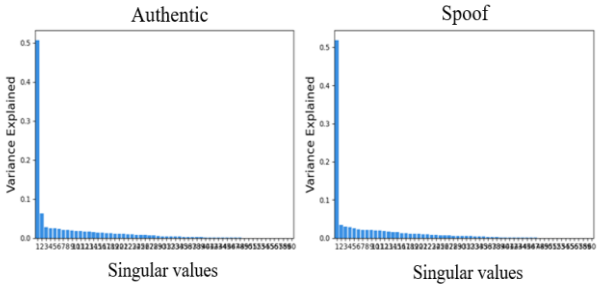


Fig. 9 Explained variance for the 6sec dataset.

The explained variance plot in Fig. 9 clearly shows that approximately 53% of the information is captured by the first vector, and 7% of the information is captured by the second vector along with their corresponding singular values.

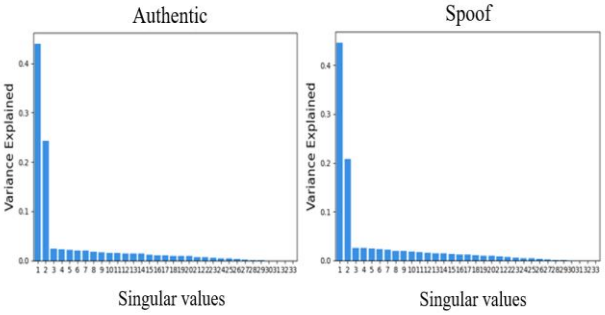


Fig. 10 Explained variance for the 4sec dataset.

Similarly, the explained variance plot in Fig. 10 clearly shows that approximately 48% of the information is captured by the first vector, and 25% of the information is captured by the second vector along with their corresponding singular values.

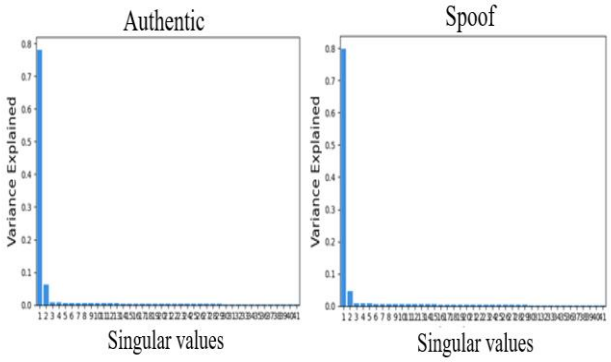


Fig. 11 Explained variance for the TEXBAT ds4 dataset.

Finally, the explained variance plot in Fig. 11 clearly shows that approximately 80% of the information is captured by the first vector, and 6% of the information is captured by the second vector along with their corresponding singular values.

Therefore, for further processing, we reconstruct the images using the top singular vectors and generate the corresponding data from this algorithm in two cases: once using the top singular vectors and once using all non-zero singular vectors, as summarized in Table 3.

Table 3 Dimensions after applying the algorithm

Dataset	Dimensions with top singular vectors ≥ 2%	Dimensions with non-zero singular vectors	Dimensions with top singular vectors ≥ 5%
6Sec	1×8	1×60	1×2
4Sec	1×6	1×33	1×2
TEXBAT	1×2	1×41	1×2

6 Machine Learning Structure

Considering the suggested dimensionality reduction algorithm, since the output is in the form of a vector, there is no need for complex neural networks for spoofing detection. Therefore, a simple MLP neural network and the kNN algorithm are employed to classify authentic and spoofed signals.

kNN method is a supervised learning technique used for classification and regression. It predicts a data point's outcome based on the majority class or average value of its k nearest neighbors, determined by a distance metric such as Euclidean distance. kNN relies on nearby samples rather than global class boundaries, making it particularly suitable for datasets with overlapping class regions. The steps of the kNN algorithm are as follows [13,38]:

- **Distance calculation**: Compute the distance between the test data and each training data point. Here, the Euclidean distance is used, which is given by Eq. (14):

$$d(I_1, I_2) = \sqrt{\sum_p (I_1{}^p - I_2{}^p)^2} \quad (14)$$

Where I_1 and I_2 represent the training data matrix and the test data matrix, respectively, and p indicates the corresponding positions in the matrices.

- **Sorting:** Arrange the data in ascending order based on their distances.
- **Selecting the nearest points:** Choose the kk points with the smallest distances and determine the frequency of each class among these k points.
- **Classification:** The class with the highest frequency among the k nearest points is assigned as the predicted class for the test data.

7 Results

For simulating the proposed method, an SDR software receiver in the MATLAB environment was used. The IF signal sampling frequency for the 6 sec and 4 sec datasets was set to 5.7143 MHz, and the IF signal frequency to 1.4054 MHz. For the TEXBAT dataset, these values were set to 0 MHz and 25 MHz, respectively. The hardware employed consisted of a laptop with a Core i7-12650H 2.3 GHz processor, RTX 3070 GPU with 8 GB RAM, and 32 GB of data memory. For spoofing detection, an MLP neural network with a single hidden layer of 64 neurons and the kNN algorithm were employed. Fully connected layers also employed Tanh and sigmoid activations along with dropout layers with a probability of 0.5. For extracting CAF images, the Doppler frequency shift search step was set to 500 Hz and 100 Hz, and the code phase search step to 0.5 chips. The number of images used is provided in Table 2. Among the total data, 70% were used for training and 30% for testing and evaluation of the neural networks. For better results the Nadam optimizer was applied to determine the network weights and thresholds. The goal of the proposed method is to accelerate processing speed while maintaining a reasonable detection accuracy (above 70%).

Before comparing the performance of the proposed methods, the results are first simulated using the original data dimensions. The results for the data with original dimensions on the 6 sec dataset (141×5714) are presented in Fig. 12 and Table 4.

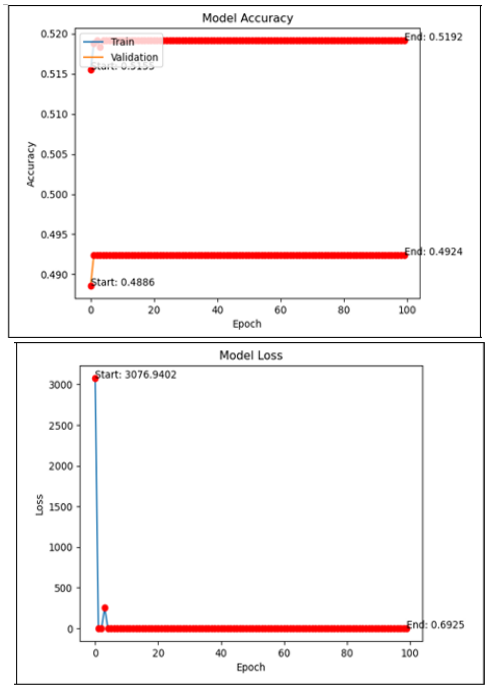


Fig. 12 Training and validation accuracy and loss for the data with original dimensions without any preprocessing.

Table 4 Measured parameter using the original data dimensions without any preprocessing.

Network	**MLP**
Number of layers	5
Time (seconds)	201
Training accuracy (%)	51.92
Validation accuracy (%)	49.24
Test accuracy (%)	49.52
Training parameters	1,272,513
Training loss	0.6919
Validation loss	0.6925

As evident from the results above, training on data with the original dimensions is very time-consuming and requires high-performance hardware for processing, while achieving high accuracy can also be challenging. Considering the hardware limitations for processing high-dimensional data, the results in Table 4 are reported for the maximum processable dimensions [pixels], which is approximately 20,000 pixels. The actual image dimensions are 805,674 pixels for 6sec dataset, which would make the execution time in Table 4 several times slower. This processing time would be much slower for the TEXBAT dataset, which contains 1,025,000 pixels.

The results for the 6Sec dataset after applying the proposed method, considering the top 5% singular vectors with dimensions 1×2, are presented in figures 13 and 14. A comparison of the measured parameters for MLP neural network and kNN on this dataset is provided in Table 5.

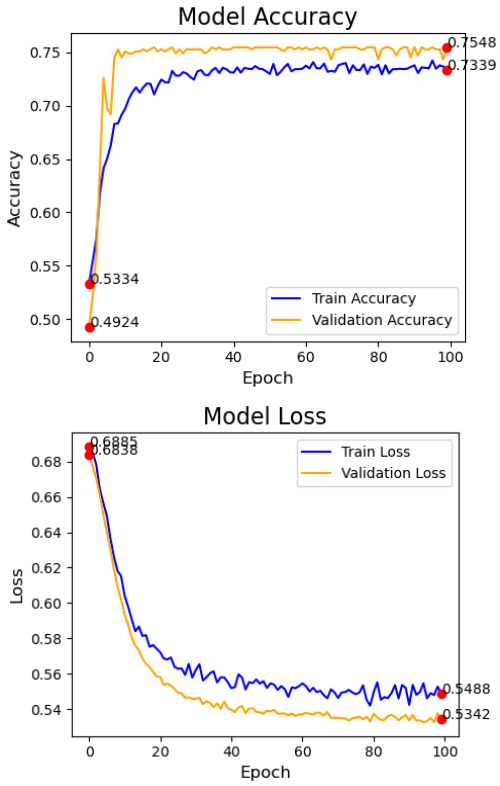


Fig. 13 Training and validation accuracy and loss for the 6Sec dataset (dimensions 1×2) generated using the proposed method on the MLP neural network.

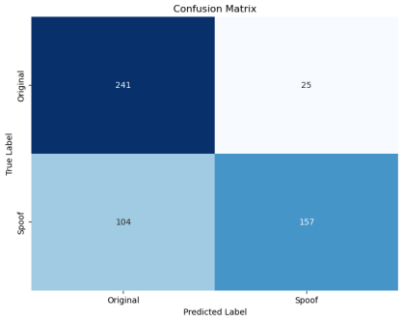


Fig. 14 Confusion matrix of the MLP neural network in the proposed method on 6Sec dataset.

8 Conclusion

In this study, a detection method based on singular values and variance from CAF data, using MLP neural network and kNN machine learning algorithms, was proposed. The presented method was evaluated on the

TEXBAT dataset and simulated data. The results demonstrate high detection accuracy despite requiring shorter processing time. Additionally, by implementing a dropout mechanism, optimizing parameter selection, and choosing appropriate activation functions, we aimed to further enhance the performance and accuracy of the results. The proposed method achieved high accuracy and robustness in distinguishing between genuine and spoofed GPS signals. Evaluation metrics included confusion matrices, accuracy, speed, and complexity. The results demonstrated the effectiveness of combining CAF with dimensionality reduction algorithms and machine learning for GPS spoofing detection. However, the CAF-based method may not be applicable to all scenarios and conditions. Furthermore, the current model was trained on specific datasets, while real-world GPS spoofing scenarios may involve varying environmental conditions, hardware configurations, and diverse spoofing techniques.

Table 5 Measured parameter results for the proposed algorithm on the 6Sec dataset.

Parameter / Network	kNN	kNN	kNN	MLP	MLP	MLP
Dimension	1×60	1×8	1×2	1×60	1×8	1×2
Time (s)	5.1	4.9	4.2	29.85	28.28	12
Training Accuracy (%)	-	-	-	86.96	78.04	73.39
Validation Accuracy (%)	-	-	-	87.64	80.8	75.48
Test Accuracy (%)	86.34	79.13	75.71	85.2	78	75.52
Training Parameters	-	-	-	9153	641	257
Training Loss	-	-	-	0.2917	0.4714	0.5488
Validation Loss	-	-	-	0.2746	0.4606	0.5342
Number of Samples	3507	3507	3507	3507	3507	3507

The corresponding results of the proposed method on the 4Sec dataset are also presented in Table 7. The corresponding results of the proposed method on the TEXBAT ds4 dataset are presented in Figures 15 and 16 and Table 6.

Table 6 Measured parameter results for the proposed algorithm on the TEXBAT ds4 dataset.

Measured Parameters /Algorithm	kNN	MLP
Dimensions	1×2	1×2
Time (s)	< 1	16.6
Training Accuracy (%)	-	99.22
Validation Accuracy (%)	-	100
Test Accuracy (%)	100	100
Training Parameters	-	65
Training Loss	-	0.0420
Validation Loss	-	0.0040
Number of Samples	4100	4100

The results for the TEXBAT ds7 dataset with dimensions 1×2 and 1×41 for different spoofing time intervals is presented in Tables 8 and 9.

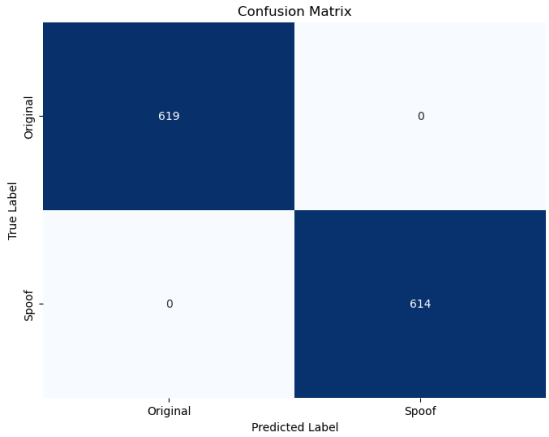


Fig 15: Confusion matrix of the MLP neural network in the proposed method on TEXBAT ds4 dataset.

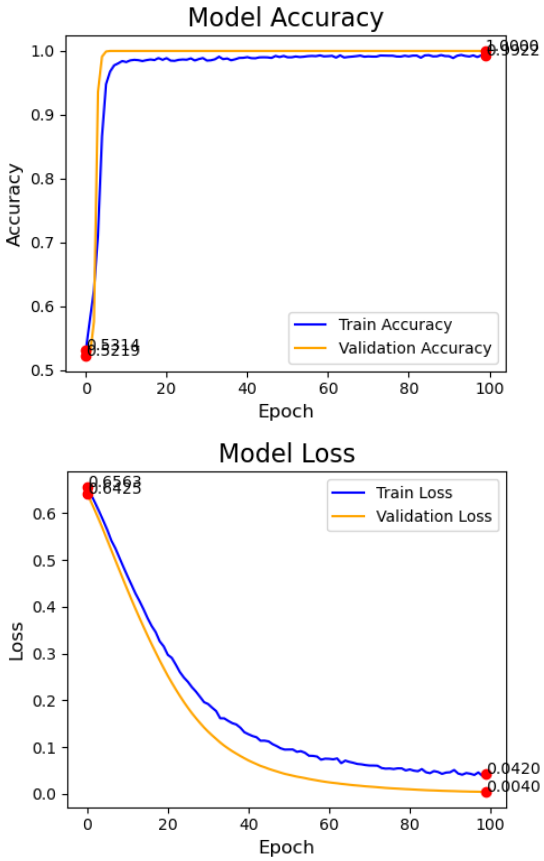


Fig. 16 Training and validation accuracy and loss for the TEXBAT ds4 dataset (dimensions 1×2) generated using the proposed method on the MLP neural network.

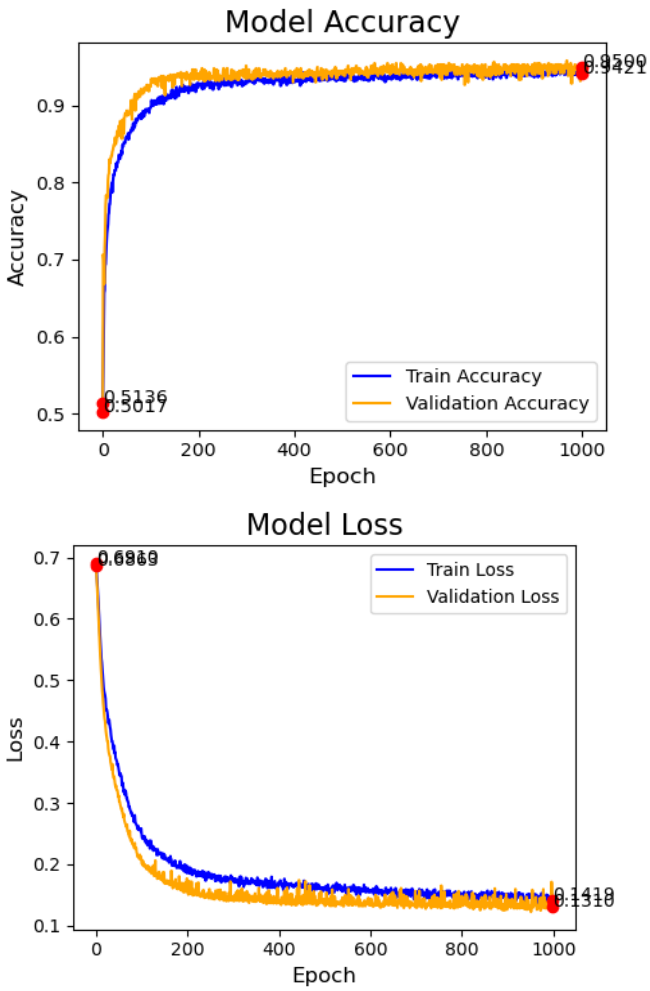


Fig. 17 Confusion matrix of the MLP neural network in the proposed method on TEXBAT ds7 dataset.

Table 7 Measured parameter results for the proposed algorithm on the 4Sec dataset.

Parameter / Network	**kNN**	**kNN**	**kNN**	**MLP**	**MLP**	**MLP**
Number of Layers	-	-	-	5	5	5
Dimension	1×2	1×6	1×33	1×2	1×6	1×33
Time (s)	1	1.2	1.2	9	9.53	16
Training Accuracy (%)	-	-	-	71.38	77.08	91.19
Validation Accuracy (%)	-	-	-	72.33	68.67	92.49
Test Accuracy (%)	97.64	98.43	99.61	74.4	78.35	93.7
Training Parameters	-	-	-	257	513	2113
Training Loss	-	-	-	0.5390	0.5714	0.2431
Validation Loss	-	-	-	0.5431	0.4941	0.2371
Number of Samples	-	-	-	1750	1750	1750

The accuracy versus k curve for the kNN algorithm is shown in Fig. 18, where the best performance is achieved at k = 5.

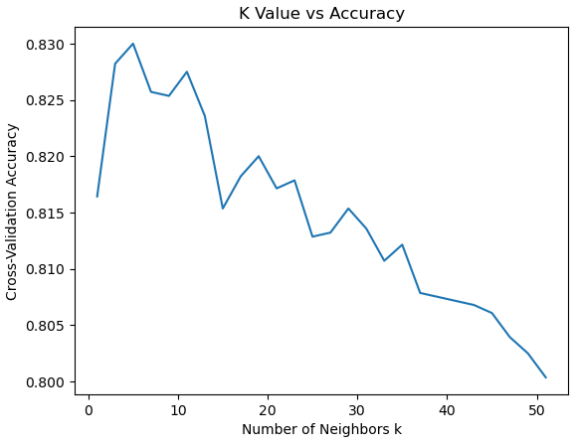


Fig. 18 accuracy versus k curve for the kNN algorithm.

By comparing the figures above, we observe the better performance of the kNN algorithm, which can be attributed to several reasons. The reduced data has a small number of dimensions, and in low-dimensional spaces, kNN usually performs better because it directly uses the distance between data points for classification. In contrast, MLP neural network generally performs better with high-dimensional data and complex patterns, where nonlinear transformations are required. kNN does not require model training; it simply stores the training data and classifies new data based on distances. MLP neural network, on the other hand, requires learning weights and biases during training, which may not be optimal for simple 1×2 data. Moreover, if the data is linearly separable or has simple clusters, kNN can naturally adapt to this structure by selecting appropriate neighbors. If the dataset is small, kNN often performs better since it uses all training data during classification.

In Figures 19 and 20, the detection performance with different activation functions is presented. It can be observed that the network performs better with the tanh and elu activation functions.

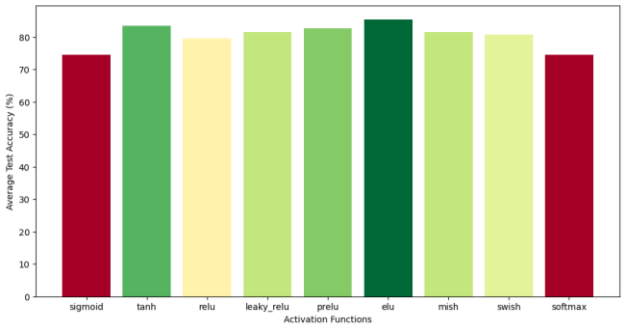


Fig. 19 Comparison of the network's detection performance with different activation functions.

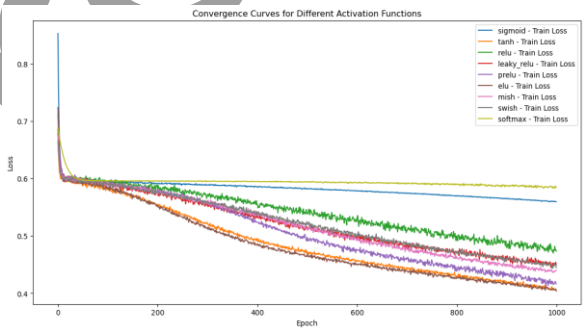


Fig. 20 Comparison of the network's loss with different activation functions.

Table 8 Measured parameter results for the proposed algorithm on the TEXBAT ds7 dataset 1×41.

Parameters / Time(s)	**130-150**	**150-370**	**110-370**	**130-150**	**150-370**	**110-370**
Algorithm	MLP	MLP	MLP	kNN	kNN	kNN
Dimension	1×41	1×41	1×41	1×41	1×41	1×41
Time (s)	22	25.6	44	2	2.4	3.1
Training Accuracy (%)	94.21	85.62	83.44	85	86	81
Validation Accuracy (%)	95	85.63	84.03	-	-	-
Test Accuracy (%)	94.83	88.63	82.86	-	-	-
Training Loss	0.1419	0.3475	0.412	-	-	-
Validation Loss	0.131	0.352	0.362	-	-	-
Number of Samples	4052	7004	15020	4052	7004	15020

Table 9 Measured parameter results for the proposed algorithm on the TEXBAT ds7 dataset 1×2.

Parameters / Time(s)	130-150	150-370	110-370	130-150	150-370	110-370
Algorithm	MLP	MLP	MLP	kNN	kNN	kNN
Dimension	1×2	1×2	1×2	1×2	1×2	1×2
Time (s)	7	10.3	18.7	< 1	< 1	< 1
Training Accuracy (%)	79	65	65.72	80	70	72
Validation Accuracy (%)	81.5	63	66	-	-	-
Test Accuracy (%)	76.83	65.34	67	-	-	-
Training Loss	0.47	0.6415	0.6171	-	-	-
Validation Loss	0.4387	0.6402	0.6011	-	-	-
Number of Samples	4052	7004	15020	4052	7004	15020

By comparing Tables 4 to 9, it can be observed that this method achieves between 99.69% and 99.99% dimensionality reduction [number of pixels]. Moreover, it provides an improvement in detection accuracy ranging from 47% to 84.3%, and in terms of time efficiency, we achieved an average improvement of about 93%. In the following, we adopt an approach aimed at further enhancing detection accuracy while maintaining an ideal runtime compared to the original dimensions. A comparison of different spoofing detection methods is presented in Table 10.

Table 10 Comparison of different spoofing detection methods.

Technique	Description	Spoofing case	Detection rate (%)
ZHOU [39] (2022)	Spoofing detection method based on weighted second-order central moment	TEXBAT Case 4 and Case 7	90
SUN [40] (2021)	Spoofing detection using Q-channel signal quality monitoring metric	TEXBAT Case 2 and Case 3	90
KHAN AND AHMAD [41] (2022)	Spoofing detection through measured autocorrelation function shape distortion	TEXBAT Case 2	83.9
		TEXBAT Case 3 and Case 4	100
		TEXBAT Case 5	97
		TEXBAT Case 6	96.8
GROSS [42] (2019)	Power-distortion monitoring for GNSS-signal authentication	TEXBAT Case 2–6	82.89
		TEXBAT Case 7	70
G. MARCHAND [33] (2023)	C/N0 and Q-channel signal quality monitoring metric	OAKBAT Case 4	82
BORHANI [12] (2020)	analysis of Cross-Ambiguity Function	-	80-95
PROPOSED METHOD	Spoofing detection based on variance and singular values analysis of Cross-Ambiguity Function	TEXBAT Case 4	100
		TEXBAT Case 7	82.86
		Simulated Data 6ec	85.2
		and 4Sec	93.7

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

M. J. Jahanta, S. Tohidi, and M. R. Mosavi: Conceptualization, Methodology, Software, Data curation, Visualization, Investigation, Original draft preparation, Data curation, Visualization, and Investigation.

D. Martin: Investigation, and Writing – Review & Editing.

Funding

No funding was received for this work.

Informed Consent Statement

Not applicable.

References

[1] D. Dardari, E. Falletti, and M. Luise, "Satellite and Terrestrial Radio Positioning Techniques: A Signal Processing Perspective," *Academic Press*, 2011.

[2] M. G. Amin, P. Closas, A. Broumandan, and J. L. Volakis, "Vulnerabilities, Threats, and Authentication in Satellite-based Navigation Systems," *Proceedings of the IEEE*, Vol. 104, No. 6, pp. 1169–1173, 2016.

[3] D. Dardari, P. Closas, and P. M. Djurić, "Indoor Tracking: Theory, Methods, and Technologies," *IEEE Transactions on Vehicular Technology*, Vol. 64, No. 4, pp. 1263–1278, 2015.

[4] N. Williams, P. B. Darian, G. Wu, P. Closas, and M. Barth, "Impact of Positioning Uncertainty on Connected and Automated Vehicle Applications," *SAE International Journal of Connected and Automated Vehicles*, Vol. 6, No. 2, pp. 155-168, 2023.

[5] Z. M. Kassas, P. Closas, and J. Gross, "Navigation Systems Panel Report: Navigation Systems for Autonomous and Semi-Autonomous Vehicles—Current Trends and Future Challenges," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 34, No. 5, pp. 82-84, 2019.

[6] K. Yu, S.-H. Fang, A. Broumandan, P. Closas, G. Retscher, and A. Dempster, "IEEE Access Special Section: Positioning and Navigation in Challenging Environments," *IEEE Access*, Vol. 11, pp. 12636–12639, 2023.

[7] M. R. Mosavi, M. Moazedi, M. J. Rezaei, and A. Tabatabaei, "Interference Mitigation in GPS Receivers," *Iran University of Science and Technology*, 2015. (in Persian).

[8] M. R. Mosavi, Z. Nasrpooya, and M. Moazedi, "Advances Anti-Spoofing Methods in Tracking Loop," *Journal of Navigation*, Vol. 69, No. 4, pp. 883-904, 2016.

[9] M. J. Jahantab, S. Tohidi, M. R. Mosavi, A. Ayatollahi, "GPS Spoofing Detection using CAF Images and Neural Networks Based on the Proposed Peak Mapping Dimensionality Reduction Algorithm and TCNN Model," *Iranian Journal of Electrical and Electronic Engineering*, Vol. 20, No. 4, pp. 41-54, 2024.

[10] M. H. Jin, Y. H. Han, H. H. Choi, C. Park, M. B. Heo, and S. J. Lee, "GPS Spoofing Signal Detection and Compensation Method in DGPS Reference Station," *11th International Conference on Control, Automation and Systems*, pp. 1616-1619, Oct 2011.

[11] 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 Proceedings of the 21st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS '08)*, pp. 2314–2325, Savannah, Ga, USA, September 2008.

[12] P. B. Darian, H. Li, P. Wu, and P. Closas, "Deep Neural Network Approach to Detect GNSS Spoofing Attacks," *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pp. 3241-3252, 2020.

[13] J. Li, W. Li, S. He, Z. Dai, and Q. Fu, "Research on Detection of Spoofing Signal with Small Delay Based on KNN," in *2020 IEEE 3rd International Conference on Electronics Technology (ICET) IEEE*, pp. 625–629, 2020.

[14] J. Li, X. Zhu, M. Ouyang, W. Li, Z. Chen, and Q. Fu, "GNSS Spoofing Jamming Detection Based on Generative Adversarial Network," *IEEE Sensors Journal*, Vol. 21, No. 20, pp. 22823–22832, 2021.

[15] B. Yang, M. Tian, Y. Ji, J. Cheng, Z. Xie, and S. Shao, "Research on GNSS Spoofing Mitigation Technology Based on Spoofing Correlation Peak Cancellation," *IEEE Communications Letters*, Vol. 26, No. 12, pp. 3024–3028, 2022.

[16] M. L. Psiaki and T. E. Humphreys, "GNSS Spoofing and Detection," *Proceedings of the IEEE*, Vol. 104, No. 6, pp. 1258-1270, 2016.

[17] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, "GPS Vulnerability to Spoofing Threats and a Review of Anti-Spoofing Techniques," *International Journal of Navigation and Observation*, Vol. 2012, No. 1, pp. 1–16, Jul. 2012.

[18] 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*, Vol. 9, pp. 153960–153976, 2020.

[19] A. Rustamov, N. Gogoi, A. Minetto, and F. Dovis, "Assessment of the Vulnerability to Spoofing Attacks of GNSS Receivers Integrated in Consumer Devices," in *2020 International Conference on Localization and GNSS (ICL-GNSS) IEEE*, pp. 1–6, 2020.

[20] X. Zhu, Z. Lu, T. Hua, F. Yang, G. Tu, and X. Chen, "A Novel GPS Meaconing Spoofing Detection Technique Based on Improved Ratio Combined with Carrier-to-Noise Moving Variance. Electronics 2022, 11, 738," *Electronics*, Vol. 11, No. 5, pp. 738, 2022.

[21] S. Tohidi and M. R. Mosavi, "Effective Detection of GNSS Spoofing Attack using a Multi-Layer Perceptron Neural Network Classifier Trained by PSO," *2020 25th International Computer Conference, Computer Society of Iran (CSICC)*, Tehran, Iran, pp. 1-5, 2020.

[22] V. Truong, A. V. Picois, J. R. Hernan, and N. Samama, "Characterization of the Ability of Low-Cost GNSS Receiver to Detect Spoofing Using Clock Bias," *Sensors*, Vol. 23, No. 5, pp. 2735, 2023.

[23] Z. Zhang and X. Zhan, "Statistical Analysis of Spoofing Detection Based on TDOA," *IEEJ Transactions Electrical and Electronic Engineering*, Vol. 13, No. 6, pp. 840–850, 2018.

[24] Q. Yang and Y. Chen, "A GPS Spoofing Detection Method Based on Compressed Sensing," in *2022 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pp. 1–5, 2022.

[25] A. Siemuri, K. Selvan, H. Kuusniemi, P. Valisuo, and M. S. Elmusrati, "A Systematic Review of Machine Learning Techniques for GNSS Use Cases," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 58, No. 6, pp. 5043–5077, 2022.

[26] 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," *The Journal of Navigation*, Vol. 71, No. 1, pp. 169-188, 2018.

[27] N. Orouji and M. Mosavi, "A Multi-Layer Perceptron Neural Network to Mitigate the Interference of Time Synchronization Attacks in Stationary GPS Receivers," *GPS Solutions*, Vol. 25, No. 3, pp. 1-15, 2021.

[28] T. Talaei Khoei and N. Kaabouch, "A Comparative Analysis of Supervised and Unsupervised Models for Detecting Attacks on the Intrusion Detection Systems," *Information*, Vol. 14, No. 2, pp. 103, 2023.

[29] A. Shafique, A. Mehmood, and M. Elhadef, "Detecting Signal Spoofing Attack in UAVs using Machine Learning Models," *IEEE access*, Vol. 9, pp. 93803–93815, 2021.

[30] F. Gallardo and A. P. Yuste, "Scer Spoofing Attacks on the Galileo Open Service and Machine Learning Techniques for End-User Protection," *IEEE Access*, Vol. 8, pp. 85515–85532, 2020.

[31] S. Semanjski, A. Muls, I. Semanjski, and W. De Wilde, "Use and Validation of Supervised Machine Learning Approach for Detection of GNSS Signal Spoofing," in *2019 international conference on localization and GNSS (ICL-GNSS) IEEE*, pp. 1–6, 2019.

[32] P. Borhani-Darian, H. Li, P. Wu, and P. Closas, "Detecting GNSS Spoofing Using Deep Learning," *EURASIP J. Adv. Signal Process.*, Vol. 2024, No. 1, pp. 14, 2024.

[33] G. Marchand, A. Toumi, G. Seco-Granados, and J. A. López-Salcedo, "Machine Learning Assessment of Anti-Spoofing Techniques for GNSS Receivers," in *Work-in-Progress in Hardware and Software for Location Computation WIHAL*, 2023.

[34] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, "A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach," *Springer Science & Business Media*, 2007.

[35] A. Bazyar, S. M. Mousavi, A. Rahmati, and M. Moazedi, "A Novel and Low-cost Method for Generating GPS Spoofing Data to Protect Navigation Systems," *Darya va Fanavari (Journal of Sea and Technology), Imam Khomeini Maritime University, Noshahr*, vol. 1, no. 1, pp. 1–12, 2013. (in Persian)

[36] T. E. Humphreys, J. A. Bhatti, D. Shepard, and K. Wesson, "The Texas Spoofing Test Battery: Toward a Standard for Evaluating GPS Signal Authentication Techniques," *Proc. ION GNSS*, Nashville, TN, 2012.

[37] T. E. Humphreys, "*TEXBAT Data Sets 7 and 8*," *Radionavigation Laboratory, University of Texas at Austin*, March 16, 2016.

[38] A. Mahroof, I. Nabi, S. Z. Farooq, and N. A. Naqvi, "Machine Learning-Based Detection of Spoofing Attacks in GNSS: A Study Using TEXBAT Dataset," *in 14th International Conference on Electrical Engineering (ICEENG), IEEE*, pp. 90–95, 2024.

[39] W. Zhou, Z. Lv, X. Deng, and Y. Ke, "A New Induced GNSS Spoofing Detection Method based on Weighted Second-order Central Moment," *IEEE Sensors Journal*, vol. 22, pp. 12064–12078, 2022.

[40] C. Sun, J. W. Cheong, A. G. Dempster, H. Zhao, L. Bai, and W. Feng, "Robust Spoofing Detection for

GNSS instrumentation using Q-channel Signal Quality Monitoring Metric," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.

[41] A. M. Khan and A. Ahmad, "Global Navigation Satellite Systems Spoofing Detection through Measured Autocorrelation Function Shape Distortion," *International Journal of Satellite Communications and Networking*, vol. 40, pp. 148–156, 2022.

[42] J. N. Gross, C. Kilic, and T. E. Humphreys, "Maximum-Likelihood Power-Distortion Monitoring for GNSS-Signal Authentication," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, pp. 469–475, 2019.

M. J. Jahantab received his B.Sc. degree in Electrical Engineering from K. N. Toosi University of Technology (KNTU) in 2022. He is currently pursuing the master's degree in Digital Electronic Engineering at Iran University of Science and Technology (IUST). His research interests include artificial intelligence, image processing, global positioning system, signal acquisition, signal processing and GPS anti-spoofing technology.

S. Tohidi received her B.Sc. and M.Sc. degrees in Electronic Engineering from respectively Shahid Beheshti University and Malek Ashtar University of Technology, Tehran, Iran. She is currently a Ph.D. Student in the Department of Electrical Engineering at Iran University of Science and Technology. Her research interests include signal processing, artificial intelligence, and GPS applications.

M. R. Mosavi received his B.Sc., M.Sc., and Ph.D. degrees in Electronic Engineering from Iran University of Science and Technology (IUST), Tehran, Iran in 1997, 1998, and 2004, respectively. He is currently a faculty member (Full Professor) of the Department of Electrical Engineering of IUST. He is the author of more than 600 scientific publications in journals and international conferences in addition to 15 academic books. His research interests include circuits and systems design. He is also editor in-chief of "Iranian Journal of Marine Technology" and editorial board member of "Iranian Journal of Electrical and Electronic Engineering" and "GPS Solutions".

Diego Martín De Andrés received the B.Sc. degree in computer engineering and the M.Sc. degree in computer science from the Department of Informatics, Carlos III University of Madrid, Spain, where he received his Ph.D. degree in 2012. Now, he is a professor at the Computer Science Department Escuela de Ingeniería Informática de Segovia, Universidad de Valladolid. His main research subjects are internet of things, cyber-physical systems, physically unclonable functions, blockchain, knowledge management, information retrieval, and research methods.