

Research

GNSS jamming detection using attention-based mutual information feature selection

Ali Reda¹ · Tamer Mekkawy¹

Received: 7 November 2023 / Accepted: 5 March 2024

Published online: 23 March 2024

© The Author(s) 2024 [OPEN](#)

Abstract

Global navigation satellite systems (GNSS) are extensively utilized for military and civilian applications. Unfortunately, because of the signal weakness, GNSS is susceptible to interference, fading, and jamming, which reduces the position accuracy. Therefore, it would be beneficial to have a simple and highly accurate model for detecting the jamming signals to improve the GNSS receiver accuracy. In this paper, we propose a hybrid deep learning (DL) model for predicting jamming signals. Initially, we utilize a feature selection algorithm that combines mutual information (MI) with the minimal redundancy maximum relevance (mRMR) technique to identify the most crucial features. Subsequently, the model undergoes training using a soft attention-double-layer bidirectional long short-term memory (A-DBiLSTM) model. This particular model has shown outstanding performance in comparison to other DL models when applied to datasets collected from both kinematic and static jamming scenarios. To assess the effectiveness and efficiency of the proposed MI feature selection algorithm, we evaluate its performance through the calculation of confusion matrices and conducting numerical simulations. The simulation results of the A-DBiLSTM model demonstrate higher accuracy, precision, recall, and $F1_{\text{Score}}$ of 98.82%, 98.4%, 98.68%, and 98.36%, respectively. By employing the MI feature selection algorithm, dimensionality reduction is achieved. Moreover, the MI feature selection algorithm reduces 19% of the learning time with almost the same accuracy.

Article Highlights

- A soft attention-double-layer bidirectional LSTM is superior for dynamic and static environments.
- Overlapping errors indicate a similar decision boundary for the mutual information, despite its slightly lower reported accuracy.
- Mutual information achieves impressive dimensionality reduction by eliminating unnecessary features and choosing the most crucial features from a broader group of original features.

Keywords Bidirectional long-short term memory (Bi-LSTM) · Attention mechanism (AM) · Deep learning (DL) · Feature selection · Jamming detection · Mutual information (MI)

✉ Ali Reda, ali.aboelyazeed@ieee.org | ¹Department of Avionics, Military Technical College, 11331 Cairo, Egypt.



Discover Applied Sciences

(2024) 6:163

| <https://doi.org/10.1007/s42452-024-05792-7>

1 Introduction

Global navigation satellite systems (GNSS) play an essential role in several applications such as aviation, maritime navigation, and military operations. However, GNSS signals are susceptible to radio frequency interference (RFI) and sometimes on-purpose jamming signals affect the receiver's accuracy. RFI may cause a complete or partial loss of GNSS signal reception, degrade signal quality, and high errors in estimating the position and time [1]. The classification of GNSS jamming signals is detailed in [2], the chirp jammers exhibit remarkably high jamming effectiveness despite their simple design and exceptionally low manufacturing costs. Thus, jamming detection for GNSS is a crucial research direction that aims to enhance robustness and reliability. Some traditional methods for detecting the jamming signals are signal power measurements and signal-to-noise ratio (SNR) analysis [3]. The importance of using multi-constellation GNSS signals to resist jamming signals for high-dynamic environments was discussed in [4]. Numerous studies discussing the behavior of GNSS receivers in the presence of jamming scenarios have been presented in [5]. However, these methods are not effective in all scenarios especially when the number of satellites in view is minimal [6].

1.1 Technical literature review

The detection of GNSS interference was suggested using several traditional methods [3, 7–9]. The Time-frequency-analysis-based GNSS interference mitigation approaches have recently received more attention [3]. Subsequently, Wigner Ville distribution (WVD) and spectrogram are glowing as two common techniques for identifying GNSS interference [7]. However, the spectrogram suffers from poor Time-frequency localization characteristics [8]. Although WVD faces some challenges with cross-term interference, WVD effectively solves the Time-frequency resolution problem [9]. Finally, the authors in [10, 11] proposed the characteristics of eigen-space and eigen-spectrum of symmetric and Toeplitz covariance matrices. This approach aims to address the challenge of detecting and estimating sinusoidal signals with extremely low SNR.

Recently, deep learning (DL) and other machine learning (ML) models have been widely used in wireless communication, e.g., channel estimation, adaptive modulations, and beamforming [12]. Furthermore, DL has gained recognition as an effective technique for RFI detection in wireless networks, because of their high prediction accuracy in various wireless network applications [13]. It has been used to identify GNSS jamming signals, because of their ability to automatically extract the features and learn the complicated patterns [14]. The detection and classification of jamming attacks against orthogonal frequency division multiplexing (OFDM) were proposed using an ML technique [15]. Subsequently, a deep reinforcement learning (DRL) model for anti-jamming scenarios was shown in static and dynamic environments [16]. Moreover, convolutional neural networks (CNNs) with long short-term memory (LSTM) were applied to identify transient RFI, which results in effectively detecting the sources of transient RFI signals [17]. For identifying signal interference, bidirectional long-short-term memory (Bi-LSTM) was implemented [18]. Consequently, the author in [12] proposed the attention mechanism (AM) with recurrent neural networks (RNNs) to predict the throughput for long-term evolution (LTE).

On the other hand, estimating the parameters from the statistical data is very important before training the model. Some techniques such as recursive [19] and hierarchical [20] were used for classifying the time series models. Numerous methods have been proposed to address the challenge of accurately classifying time series data [21]. One of the well-established and commonly used approaches in time series classification is the integration of a nearest neighbor classifier with a distance function [22]. Notably, the dynamic time warping distance, when combined with the nearest neighbor classifier, has been recognized as a robust baseline for this task [21].

Although using all features in DL assures higher accuracy, the required training time is higher if it is compared with some selected features. Thus, feature selection methods minimize the learning time of the model, which can be categorized into three primary approaches [14]; filter-based approaches, wrapper approaches, and embedding approaches. The semi-supervised feature selection approach with manifold regularization [23] was adopted to maximize the classification margin between distinct classes. In [24], the algorithm's complexity is computed by assessing the decomposability order, and the benefit of incorporating feature dependencies exhibits considerable reducing returns. Additionally, mutual information (MI) was utilized to get a set of Gabor features that are informative and

non-redundant [25], these chosen features are then refined using kernel methods to enhance their effectiveness in recognition. The efficiency of MI feature selection for face recognition was discussed in [26]. Then, the Pearson correlation coefficient (PCC) was applied to further reduce the number of features [27].

1.2 Motivation

In the classification of different outliers, DL approaches have become increasingly popular as a substitute for conventional methods, like statistical hypothesis testing [28]. However, statistical hypothesis testing has its limitations when dealing with intricate problems, as it necessitates considerable effort to estimate probability density functions for all hypotheses, particularly when the number of classes expands. On the other hand, DL approaches tackle the challenges associated with establishing robust statistical hypothesis testing in complex problem domains [29].

From the above-mentioned works, conventional jamming detection methods often rely on specific assumptions and conditions or require manual feature engineering, which restricts their applicability to datasets with homogeneous characteristics [13]. In contrast, DL handles the diverse nature of the GNSS dataset, which is more versatile and suitable for real-world scenarios. Furthermore, DL techniques can overcome the limitations associated with conventional methods for detecting continuous wave (CW) and chirp jamming signals. Because DL is specifically trained to identify the complex and nonlinear characteristics exhibited by CW and chirp jamming signals. On the other hand, although feature selection in DL decreases the overall model accuracy, the utilization of MI feature selection brings forth important considerations.

1.3 Our contributions

In this paper, we use a filter-based approach that uses MI [26] for selecting the highest mutual features. Subsequently, we present the nonlinear relationship between the input features and the target characteristic. The MI method in this study is employed to choose the most crucial features from a broader group of original features. To assess the effectiveness of our proposed MI feature selection approach, we adopt the soft Attention-based double-layer bidirectional long short-term memory (A-DBiLSTM) model for generating and comparing the confusion matrices. A minor reduction of 1% in accuracy can be considered acceptable, given the significant advantages offered by the MI model in terms of computing speed and model size. Furthermore, MI achieves impressive dimensionality reduction by eliminating unnecessary features while retaining the most crucial ones. This effective dimensionality reduction ensures valuable information is preserved. The substantial 19% decrease in training time enables real-time performance, making MI a suitable choice for low-power edge devices and ongoing online learning. The contributions of this study can be summed up as

- We propose an MI filter-based feature selection algorithm with the minimal redundancy maximum relevance (mRMR) technique to explore the correlation and interdependence between different features in jamming GNSS datasets, by selecting the most relevant features.
- The prediction accuracy is improved by using AM between double layers of the Bi-LSTM model and validating the model in both dynamic and static environments.
- Selecting the crucial vectors using the MI feature selection algorithm reduces the learning time by 19% and reduces the dimensionality of the optimization problem.

In the rest of the paper, GNSS signal representation and problem characterization are discussed in Sect. 2. Section 3 presents the prediction model, the collected datasets, and the pre-processing phase. The proposed MI feature selection algorithm is shown in Sect. 4. In Sect. 5, the DL model used in the proposed algorithm is introduced. Section 6 analyzes the computational complexity of our model. The experimental methods followed by discussions are shown in Sect. 7. Finally, Sect. 8 concludes and summarizes the results.

2 GNSS signal representation

In an environment with interference, the received signal at the GNSS receiver can be affected by both natural interference, $w(t)$ and intentional jamming signals, $J(t)$. GNSS jammers are frequently employed to compromise electronic systems. These jammers are typically installed within moving vehicles or other locations on the Earth's surface. There are various types of GNSS jamming signals [2], e.g., CW interference, pulse jammers, FM modulated signal, and chirp jamming signal. Consequently, the total received signal of the GNSS receiver can be represented as [9]:

$$S_{RF}(t) = \sum_{i=1}^{L_s} y_i(t) + J(t) + w(t) \quad (1)$$

where L_s denotes the number of satellites in view and $y_i(t)$ is the RF received signal from the i^{th} GNSS satellite that can be written as [11]:

$$y_i(t) = A_i e_i(t - \tau_i) m_i(t - \tau_i) \cos [2\pi(f_{RF} + f_{d,i}) + \theta_i(t)] \quad (2)$$

where A_i is the amplitude of the signal, $e_i(t - \tau_i)$ is the pseudo-random noise (PRN) of the periodic code sequence, τ_i is the code phase delay caused by the transmitted channel delay, $m_i(t)$ represents the navigation message, f_{RF} is the carrier frequency of the GNSS signal, $f_{d,i}$ is the Doppler frequency, and $\theta_i(t)$ denotes the initial carrier phase offset. The block diagram of the GNSS receiver is shown in Fig. 1. The GNSS received signal, $S_{RF}(t)$, is demodulated to an intermediate frequency (IF) stage using a local oscillator in the GNSS receiver [1]. The signal is then converted from analog to digital and passed through acquisition and tracking stages. The received signal at IF can be represented as [9]:

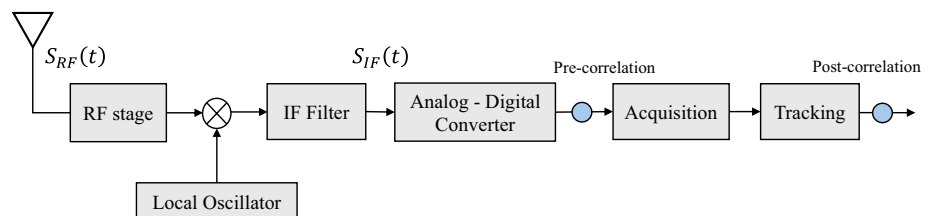
$$S_{IF}(t) = \sum_{i=1}^{L_s} A_i \tilde{e}_i(t - \tau_i) m_i(t - \tau_i) \cos [2\pi(f_{IF} + f_{d,i}) + \theta_i(t)] + J(t) + w(t) \quad (3)$$

where f_{IF} is the IF of the GNSS receiver. The spreading waveform is filtered as $\tilde{e}_i(t - \tau_i)$ after the received signal in the GNSS receiver front-end; however, for simplicity, we ignore the filter's effect, i.e., $\tilde{e}_i(t) \approx e_i(t)$. In this paper, our challenge is to accurately identify the interfering term by predicting the value of the next-time-step signal with minimal error. Since jammers employ a brute force method with a straightforward signal to disrupt the receiver, the distortion of the jammer signal due to factors such as time delay spreading or frequency spreading is not considered significant [2].

The effectiveness of the jamming signal necessitates a relatively high power level to disrupt the operation of the GNSS receiver. This is essential because the signal experiences attenuation due to propagation near the Earth's surface and obstruction by various obstacles. Additionally, the complexity and cost of the jammers are very critical issues. Therefore, the chirp and CW jammers exhibit remarkably high jamming effectiveness despite their simple design and exceptionally low manufacturing costs [5].

In this paper, we address linear chirp jamming and CW jamming signals. Firstly, for linear chirp jamming, the jamming signal, denoted as $J(t)$, undergoes frequency modulation with nearly constant amplitude, resulting in a

Fig. 1 Block diagram of GNSS receiver



sweep jamming scenario. Sweep jamming degrades the accuracy of GNSS receivers in calculating location by scanning available frequency bands to lock onto the operational carrier frequency. Subsequently, the jammer transmits a high-power signal on this locked carrier frequency. The linear chirp jamming signal, represented as $J_{ch}(t)$, can be expressed as

$$J_{ch}(t) = A \cos \left[2\pi \left(f_0 t + \frac{B}{2T} t^2 \right) + \varphi_0 \right] \quad (4)$$

where φ_0 is the carrier phase of the jamming signal at $t = 0$, which has a uniform distribution with $[-\pi, +\pi)$, and A is the amplitude of the instantaneous jamming signal. f_0 is the starting frequency, B is the bandwidth of the chirp signal, T is the duration of the chirp.

On the other hand, the malicious jamming signal, $J(t)$ in (1), can be presented as CW jamming signals. In this form of interference, the jammers generate single or multi-tone frequency signals within the GNSS frequency bands, disrupting the reception of satellite signals by GNSS receivers in the nearby area. This CW signal can be represented as [23]:

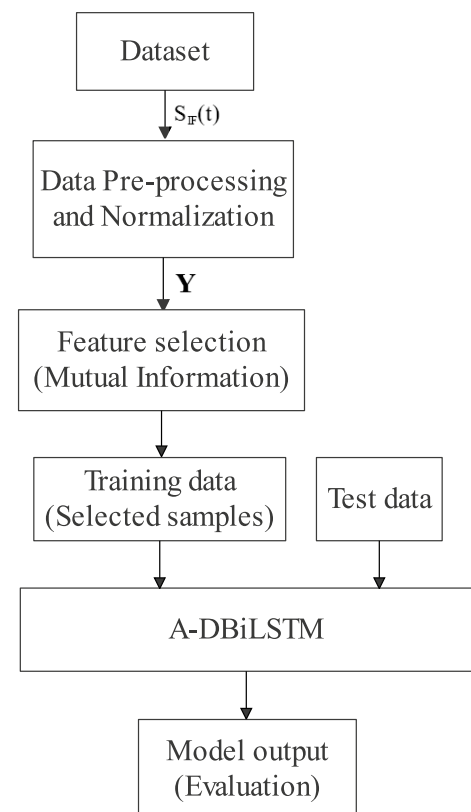
$$J_{CW}(t) = A_{CW}(t) \cos [2\pi f_{CW}(t)t + \varphi_0] \quad (5)$$

where the carrier frequency, denoted as $f_{CW}(t)$, the amplitude factor represented by $A_{CW}(t)$, and the initial carrier phase denoted as φ_0 .

3 Materials and methods

This section describes the prediction model to identify the jammed GNSS signals. Additionally, it provides an overview of the available dataset and highlights the importance of data pre-processing.

Fig. 2 Flow chart of the prediction model



3.1 Prediction model

The proposed prediction model is depicted in Fig. 2. First, the received IF GNSS signal, $S_{IF}(t)$, comprises both pure and jamming signals, which need a certain level of pre-processing, including filtering and down-conversion. The low-quality and outlier data is removed from the data during this stage. Consequently, the remaining clean GNSS data is analyzed to extract the relevant feature vectors for the prediction model. As $\mathbf{Y} \in \mathbb{C}^{N \times T}$ is a matrix with N features at time-steps T , the extracted feature vectors are represented in \mathbf{Y} as

$$\mathbf{Y} = \begin{bmatrix} f_1^1 & \dots & f_T^1 \\ \vdots & \ddots & \vdots \\ f_1^N & \dots & f_T^N \end{bmatrix}_{N \times T} = [\mathbf{x}_1 \dots \mathbf{x}_T] \quad (6)$$

where \mathbf{x}_j is the j^{th} time-sample with N features, and f represents the feature variable. The input datasets are divided into two sets; one for training and the other for test. The training set is specified as $\mathbf{R} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K)$, and the test set reflects the other input data as $\mathbf{Z} = (\mathbf{x}_{K+1}, \mathbf{x}_{K+2}, \dots, \mathbf{x}_T)$, where $K \in T$. MI feature selection algorithm is utilized to select the most important features with the most highly relevant and least redundant data. The proposed MI feature selection algorithm for the GNSS jamming detection is evaluated using the A-DBiLSTM model.

In this study, the training dataset \mathbf{R} comprises K feature vectors, each representing a specific time step and containing relevant variables related to GNSS datasets. The training process is carried out separately for the GNSS dataset. Following the MI feature selection algorithm, the resulting input feature vector \mathbf{y} is utilized in each training phase.

3.2 Dataset

In this paper, the datasets were sequentially collected, capturing information at different time steps (32,832 sample/time-step). This dataset was specifically compiled for the Ultrahack Galileo innovation challenge [30], it can be processed using the Georinex Python package because it is stored in Receiver INdependent EXchange (RINEX) format [31].

The RINEX format incorporates different observation codes that are used to differentiate between various tracking modes and the generation of measurements. These codes encompass specific signal types, such as D (data) and P (pilot), components represented by I (in-phase) and Q (quadrature), and duration indicated by S (short) and L (long). Additionally, codes like B and C are employed to indicate data-only and pilot-only tracking, respectively.

The dataset is divided into two sets: GPS and Galileo, each dataset is affected by two types of jamming signals (i.e., chirp jamming, and CW). In addition, ambient natural interference was introduced during the data collection process. The data was captured using three distinct receivers: a professional receiver, a low-cost receiver, and an Android device receiver. In our simulation environment, the jammer-to-signal ratio ranges from 3 to 40 dB for the datasets used. The jamming GPS and Galileo datasets are parsed based on these three types of data collection for the static scenario, kinematic scenario, and natural scenario. Each dataset consists of 18 variables, as specified in Table 1. The GNSS data comprises three frequency bands for GPS (L1: 1575.42 MHz, L2: 1227.60 MHz, and L5: 1176.45 MHz), as well as three bands for Galileo (E1: 1575.42 MHz, E5a: 1176.45 MHz, and E5b: 1191.795 MHz).

3.3 Data pre-processing

The proposed model is implemented in the post-correlation stage of the GNSS receiver block as shown in Fig. 1. After the GNSS data acquisition, the data is prepared for extracting the input features during the data pre-processing stage, as described in Fig. 2. It is crucial to check the dataset for any missing, erroneously written, or overly large values. Because Python cannot employ heterogeneous data types, any non-numeric items should be converted to numeric values [32]. After extracting the features and labels into TensorFlow data structures, the data is normalized. Then, the dataset is split into 25% for validation and 75% for training.

Table 1 List of basic features for GNSS RINEX dataset

| GNSS system | Carrier frequency (MHz) | Channel/code | Pseudo range | Carrier phase | Doppler shift | Signal strength |
|-------------|-------------------------|--------------|--------------|---------------|---------------|-----------------|
| GPS | L1/ 1575.42 | C/A | C1C | L1C | D1C | S1C |
| | L2/ 1227.60 | Z-tracking | C2W | L2W | D2W | S2W |
| | L5/ 1176.45 | Q | C5Q | L5Q | D5Q | S5Q |
| Galileo | E1/ 1575.42 | C | C1C | L1C | D1C | S1C |
| | E5a/ 1176.45 | Q | C5Q | L5Q | D5Q | S5Q |
| | E5b/ 1207.14 | I | C7I | L7I | D7I | S7I |

To process our raw observation data, we divide it into equal-sized windows. We specifically choose a window size of 10 s, as this aligns with the minimum interval for the presence of jammers in our utilized datasets. This approach allows us to analyze the data as a time series, considering multiple time steps for each input. Each time step represents a specific moment and encompasses observations from multiple satellites.

4 MI feature selection algorithm

Certainly, using the full set of input features improves the detection accuracy. However, with a large number of highly correlated features, the accuracy is decreased and the processing time is increased. Therefore, finding some methods for dimensionality reduction of the feature space is very important. Several researchers have utilized feature selection techniques on specific datasets to detect various attacks [33].

The mutual dependence of the variables demonstrates the amount of information that can be learned about the variables. When two discrete random variables, A and B , are presented in a system, such that; A is the input variable with probability distribution of $(p_A(a_1), p_A(a_2), \dots, p_A(a_n))$, the information entropy of random A can be calculated as follows:

$$H(A) = - \sum_{i=1}^n p_A(a_i) \log_2 p_A(a_i) \quad (7)$$

Similarly, the information entropy of random variable B . The degree of uncertainty is the metric of MI that can be connected to entropy. Entropy and conditional entropy can be used to indicate the MI as follows:

$$\begin{aligned} I(A, B) &= H(B) - H(B|A) \\ &= \sum_A \sum_B p(a, b) \log \frac{p(a, b)}{p(a)p(b)} \end{aligned} \quad (8)$$

where $H(B|A)$ denotes the conditional entropy. Which is non-negative and equals 0 if A and B are independent [34]. On the other hand, the mRMR algorithm is a highly effective method for selecting the features, which utilizes MI to assess the relevance and redundancy of important variables. Choosing the most effective subset of features, and the mRMR technique maximizes the relevant features and reduces the redundancy. The mRMR's ranking criterion is [35]:

$$J_m(f_k) = \max_{f_i \in S, f_k \in F-S} \left[I(f_k; L) - \frac{1}{|S|} \sum I(f_k; f_i) \right] \quad (9)$$

where f_i can be any feature in S , $I(A, B)$ is given in (8), f_k is a feature candidate, F is the entire feature set, S is the feature set that has already been chosen, and L are the class labels. The second term of (9) considers the redundancy between a feature candidate and previously selected features in terms of paired variables. However, it only considers the relevance and conditional redundancy of up to two variables, failing to fully incorporate the combined relevance and conditional redundancy for more variables. The MI feature selection algorithm results in 9 selected features for each input vector, which are denoted by \mathbf{y} . We summarize the steps of the feature selection based MI with mRMR as shown in Algorithm 1.

Algorithm 1 Feature selection based MI with mRMR.

```

Input :  $N$  the number of features in the dataset,  $M$  the
        selected features,  $L$  is class.
Output: The selected features set  $S$ .
1 initialization  $S = \emptyset$ ,  $\mathbf{x} = \{f_1, f_2, \dots, f_N\}$ ;
2 for  $i = 1 \rightarrow n$  do
3   | compute  $I(f_k; L)$ 
4 end
5 Extract the feature  $f_{new}$  from  $\mathbf{x}$  by maximizing the MI, and
  then incorporate it into  $S$ .
6  $f_{new} \leftarrow f_k$ .
7  $count \leftarrow 1$ .
8 while  $count < M$  do
9   |  $l = N - count$ ;
10  | for  $m = 1 \rightarrow l$  do
11    | for  $k = 1 \rightarrow count$  do
12      | compute  $I(f_k; f_i)$ .
13    | end
14    | compute (9).
15  | end
16  | Extract the feature  $f_m$  from  $\mathbf{x}$  by satisfying (9), and
    | then incorporate it into  $S$ .
17  |  $count = count + 1$ 
18 end

```

5 Jamming detection based DL models

The RNN is a memory-based model specifically designed to predict time series data. It incorporates feedback from previous layers and propagates information through hidden layers utilizing an activation function. However, as mentioned earlier, the challenge arises when applying RNN to lengthy sequences of GNSS data due to the gradient vanishing problem. In response to this challenge, an LSTM model is employed [36]. LSTM demonstrates the capability to handle extended data sequences effectively. Particularly in the context of long-term series data, LSTM surpasses RNN in prediction tasks by adeptly capturing long-term dependencies.

5.1 LSTM model

LSTM is comprised of three distinct stages; the first stage involves the forget gate, which filters the previous state layer C_{t-1} through the first unit, which is based on the calculated values of the forget gate at time-step t . The second stage is the input gate's value i_t at the time t , which determines the values of updates. The state value of the candidate vector \hat{C}_t is determined by the \tanh unit, which is calculated based on the current input and the previous hidden state.

The LSTM model takes time series feature vectors, with one vector per time-step, as input and transforms them into probability vectors at the output layer for detection purposes. Processing the output from the preceding layer, the LSTM layer, which is equipped with n hidden units, operates on the input. Following this, the output from the LSTM layer is then fed into a fully connected layer, succeeded by the softmax activation layer. This particular configuration is deliberately crafted for the explicit purpose of detecting jamming.

5.2 Bidirectional LSTM model

Bi-LSTM has emerged as a robust solution for context-sensitive natural language processing (NLP) prediction challenges. Regarding time series prediction, the Bi-LSTM outperforms the LSTM model. Because, Bi-LSTM contains two sequence-related LSTM hidden layers, one of which uses information from the past at time $t - 1$, while the other does it from the future at time $t + 1$ to be used at time t . Therefore, it can make use of both past and future data. The forward and the backward hidden-layers, $\overrightarrow{h}_t, \overleftarrow{h}_t$ respectively, are described as:

$$\vec{h}_t = \tanh(\mathbf{W}_{y\vec{h}} \mathbf{y}_t + \mathbf{W}_{h\vec{h}} \vec{h}_t + \mathbf{b}_{\vec{h}}) \tag{10a}$$

$$\overleftarrow{h}_t = \tanh(\mathbf{W}_{y\overleftarrow{h}} \mathbf{y}_t + \mathbf{W}_{h\overleftarrow{h}} \overleftarrow{h}_t + \mathbf{b}_{\overleftarrow{h}}) \tag{10b}$$

$$\mathbf{O}_t = \mathbf{W}_{\vec{h}O} \vec{h}_t + \mathbf{W}_{\overleftarrow{h}O} \overleftarrow{h}_t + \mathbf{b}_O \tag{10c}$$

where $\mathbf{W}_{y\vec{h}}$ and $\mathbf{W}_{h\vec{h}}$ denote the weight matrices in forward and backward LSTM cell, respectively, the both forward and reverse LSTM cells' bias vectors are $\mathbf{b}_{\vec{h}}$ and $\mathbf{b}_{\overleftarrow{h}}$, respectively. \mathbf{O}_t is the output sequences of Bi-LSTM model. The weight matrices that connect the output layer with the forward and backward hidden layers as well as the output bias vector are designated as $\mathbf{W}_{\vec{h}O}$ and $\mathbf{W}_{\overleftarrow{h}O}$, and \mathbf{b}_O , respectively.

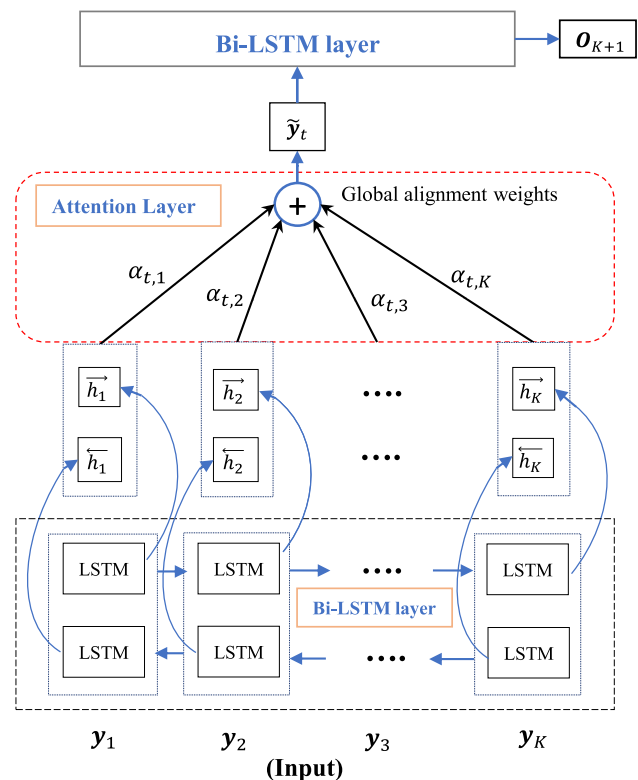
5.3 A-DBiLSTM

Attention mechanism is a computational method that selects the most important information, by assigning each component of the incoming data a different priority level. AM is a technique for the output \mathbf{O} to give particular attention to various components of the input \mathbf{y} , where each component's contribution or weight is stated. AM is applied to the encoder-decoder with LSTM layers to focus on the variables that have a major impact on the output to increase the detection accuracy [37]. The structural layout of the AM is shown in Fig. 3.

A-DBiLSTM is an encoder-decoder algorithm that incorporates an AM. The encoder of A-DBiLSTM consists of a Bi-LSTM layer with forward and backward hidden states, $(\vec{h}_i, \overleftarrow{h}_i)$, which allows it to extract hidden information from each unit-sequence of the input. The Bi-LSTM layer learns the mapping from \mathbf{y}_t to h_t for the input sequence $(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)$ of length k :

$$h_t = f_1(h_{t-1}, \mathbf{y}_t) \tag{11}$$

Fig. 3 Attention block with DBiLSTM



where at time-step t the hidden layer represents h_t and f_1 and f_1 corresponds to an LSTM unit. Subsequently, the attention weights assigned to various hidden layers h_i of the encoder, considering the hidden layer h_i of the decoder, can be computed through the following equations:

$$\text{score}(s_t, h_i) = \mathbf{V}_a^T \tanh(\mathbf{W}_a [s_t; h_i]) \quad (12)$$

$$\alpha_{t,i} = \frac{\exp(\text{score}(s_t, h_i))}{\sum_{j=1}^n \exp(\text{score}(s_t, h_j))} \quad (13)$$

Subsequently, c_t is the context vector (summation of input sequence hidden states) such that:

$$c_t = \sum_{i=1}^n \alpha_{t,i} h_t \quad (14)$$

where \mathbf{V}_a and \mathbf{W}_a are the weight matrices that required to be learned, $\alpha_{t,i}$ is the attention score of the i^{th} feature at a time-step, which denotes the correlation of an input feature x_t^i in $\mathbf{y}^i = (x_1^i, x_2^i, x_3^i, \dots, x_K^i)$. To guarantee that the attention weight values sum up to one, the soft-max function in (13) is employed. The computed attention values are then utilized to allocate weights to the input feature vector \mathbf{y}_t at time-step t , $\tilde{\mathbf{y}}_t = (\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \dots, \alpha_t^N x_t^N)$.

The output feature vector $\tilde{\mathbf{y}}_t$ is created at the same time-step as the feature attention layer's adaptive selection of multiple features at time-step t . In contrast to \mathbf{y}_t , $\tilde{\mathbf{y}}_t$ prioritizes various features rather than giving each one the same amount of weight. As a result, $\tilde{\mathbf{y}}_t$ rather than the output h_t from the first layer is used as the input value for the Bi-LSTM network unit's final layer.

The output feature vector of the attention layer, denoted as $\tilde{\mathbf{y}}_t = (\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \dots, \tilde{\mathbf{y}}_K)$, is passed to the final layer of the A-DBiLSTM model. This final layer, which is also a Bi-LSTM, extracts relevant information from the data sequence. The purpose of this layer is to anticipate \mathbf{O}_{K+1} by learning from the sequential data.

6 Complexity analysis

Let's assume that M represents the number of features intended for selection, K denotes the total number of instances in the dataset, and N indicates the overall number of features present. The time complexity of MI is $\mathcal{O}(K)$ since all instances require examination for probability estimation. In the mRMD method, each iteration involves computing the information terms for all features, resulting in a time complexity of $\mathcal{O}(KN)$. Considering that the total number of iterations is M , the overall time complexity of mRMD amounts to $\mathcal{O}(MKN)$.

Moreover, the computational complexity of a single layer in the LSTM model is $\mathcal{O}(Kd^2)$, where d represents the dimension of the model and K is the input length. When the LSTM model consists of n layers, the time complexity for multiple LSTM layers becomes $\mathcal{O}(Kd^2n)$. This results in the respective complexities of $\mathcal{O}(2Kd^2n)$ for the Bi-LSTM model and $\mathcal{O}(K^2d)$ for the AM model. Hence, the total computational complexity of the A-DBiLSTM model can be characterized as $\mathcal{O}(2Kd^2N + K^2d)$.

7 Results and discussion

In this section, we assess the performance of four different DL models namely, LSTM, Bi-LSTM, A-BiLSTM, and A-DBiLSTM from various perspectives. Our evaluation process involves initially examining each DL model's performance using the complete feature set on the datasets independently to identify the model with the most robust performance. Following this, we further evaluate the models' performance using the MI feature selection algorithm. This assessment aims to investigate the influence of the feature selection technique on classifier accuracy.

7.1 Evaluation metrics

In this paper, we show the effectiveness of the prediction model which is proposed for GNSS jamming detection. The pure GNSS signal is devoid of interference, whereas the jamming signal is affected by interference. First, we use several

optimizer techniques and batch sizes to determine the prediction model's accuracy. The confusion matrices are then computed using the batch size and most effective optimizer approach.

We define four parameters for the confusion matrix computation: 1) A jamming signal is considered to be true positive (TP) if the model accurately identifies it as a jamming signal. 2) True negative (TN) indicates that the pure GNSS signal has been accurately recognized as such. 3) False positive (FP) signals indicate that the pure GNSS signal has been mistaken for a jamming signal. 4) False negative (FN) is a term used to describe when a jamming signal is mistaken for pure signals. We use the formula $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$ to calculate the percentage of the right prediction accuracy based on the parameters listed above. Additionally, Precision is defined as: $\text{Precision} = \frac{TP}{TP+FP}$ to show how effective the model is at predicting a particular class. Recall = $\frac{TP}{TP+FN}$ is also calculated to show how frequently the model can identify a specific category. Finally, the error's form is taken into account in addition to the number of prediction errors when computing an alternative evaluation measure known as the $F1_{\text{Score}}$. In other terms, it is stated as the harmonic mean of the recall and precision:

$$F1_{\text{Score}} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (15)$$

7.2 Experimental implementation

In this study, we utilize Keras, which is a Python library that relies on TensorFlow, Scikit-Learn, and Georinex. The experiments are conducted on a Windows 11 64-bit operating system and carried out on an ASUS laptop with an 11-th Intel Core i7-11800 H processor, 16GB of RAM, and a GeForce RTX 3060 graphics card with 4GB of GDDR4 memory.

In the training phase: we use *tanh* as an activation function, mean square error (MSE) as an objective function, and the learning rate (Lr) is adjusted to 0.0001. The accuracy for various batch sizes (32, 64, 128, 256, and 512) and several optimizer algorithms (SGD, Adam, Adagrad, and Adamax) were compared using the A-DBiLSTM model. We observe that the model underfits with 32 and 64 batch sizes, while it overfits with 256 and 512 batch sizes. Moreover, Adam optimizer offers the best level of accuracy. Therefore, to compute the confusion matrices, we use the Adam optimizer with a 128 batch size.

7.3 Performance evaluation of MI feature selection algorithm

For our jamming detection problem, we employ a correlation analysis to select the most relevant features from the full feature space. The performance of the proposed MI feature selection algorithm is compared among four DL models: LSTM, Bi-LSTM, A-BiLSTM, and A-DBiLSTM, using the datasets presented in Tables 2 and 3.

Table 2 displays the outcomes for LSTM, Bi-LSTM, A-BiLSTM, and A-DBiLSTM based on the proposed MI feature selection algorithm. On the other hand, Table 3 shows the outcomes for the DL models when utilizing the full set of features. To assess the impact of the feature selection algorithm on classification accuracy and GNSS jamming detection, we analyze the confusion matrices for the DL models in Figs. 4 and 5 for the GPS and Galileo datasets, respectively. Notably, we observe that the LSTM model exhibits the highest FP and FN rates, whereas the A-DBiLSTM model demonstrates the lowest FP and FN rates.

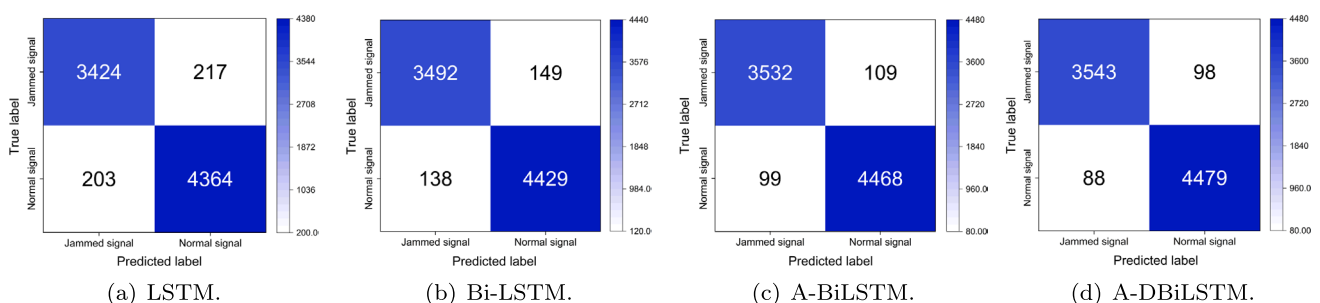


Fig. 4 Confusion matrices for different DL models using GPS dataset and MI algorithm

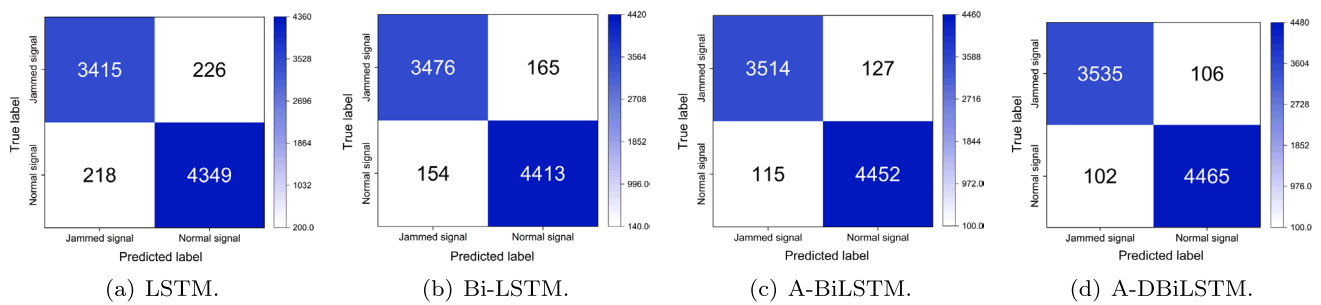


Fig. 5 Confusion matrices for different DL models using Galileo dataset and MI algorithm

Table 2 MI algorithm evaluation parameters

| Dataset | Model | Accuracy (%) | Precision (%) | Recall (%) | F1 _{Score} (%) | Training time (Sec) |
|---------|-----------|--------------|---------------|------------|-------------------------|---------------------|
| GPS | LSTM | 94.88 | 94.05 | 94.40 | 94.22 | 637 |
| | Bi-LSTM | 96.5 | 95.93 | 96.17 | 96.05 | 1014 |
| | A-BiLSTM | 97.47 | 97.01 | 97.27 | 97.14 | 1263 |
| | A-DBiLSTM | 97.73 | 97.31 | 97.55 | 97.44 | 1393 |
| Galileo | LSTM | 94.59 | 93.79 | 93.84 | 93.90 | 630 |
| | Bi-LSTM | 96.11 | 95.47 | 95.54 | 95.61 | 1012 |
| | A-BiLSTM | 97.05 | 96.51 | 96.89 | 96.67 | 1258 |
| | A-DBiLSTM | 97.47 | 97.09 | 97.10 | 97.14 | 1384 |

Table 3 Full-feature vectors evaluation parameters

| Dataset | Model | Accuracy (%) | Precision (%) | Recall (%) | F1 _{Score} (%) | Training time (Sec) |
|---------|-----------|--------------|---------------|------------|-------------------------|---------------------|
| GPS | LSTM | 95.94 | 95.23 | 95.6 | 95.41 | 786 |
| | Bi-LSTM | 97.6 | 97.12 | 97.38 | 97.25 | 1252 |
| | A-BiLSTM | 98.55 | 98.05 | 98.49 | 98.27 | 1559 |
| | A-DBiLSTM | 98.82 | 98.4 | 98.68 | 98.36 | 1720 |
| Galileo | LSTM | 95.6 | 94.95 | 95.13 | 95.04 | 778 |
| | Bi-LSTM | 97.14 | 96.59 | 96.95 | 96.77 | 1244 |
| | A-BiLSTM | 98.14 | 97.68 | 98.1 | 97.89 | 1553 |
| | A-DBiLSTM | 98.58 | 98.24 | 98.4 | 98.32 | 1709 |

The classification performance of each model relies on the reduced feature subset obtained from the MI feature selection algorithm as indicated in Table 2 and Fig. 6. The A-DBiLSTM model performed the best accuracy is 97.73%, and F1_{Score} is 97.44% for the GPS dataset. Additionally, the A-DBiLSTM model has the best accuracy by 97.47%, and 97.14% F1_{Score} for the Galileo dataset.

The receiver operating characteristic (ROC) curve in Fig. 7 shows the comparison between DL models, which reflects that the A-DBiLSTM model outperforms the other models. We demonstrate the effectiveness of the MI feature selection algorithm by presenting the inference training time and the accuracy for the four DL models using the full feature set. To show the impact on accuracy when decreasing the features by the MI algorithm, we compute the accuracy obtained by full feature vectors, as illustrated in Table 3.

In Table 4, we have conducted a comparative analysis to evaluate the effectiveness of our feature selection method, MI, in combination with the A-DBiLSTM DL model. We have compared our approach with two recent studies that utilized different feature selection techniques, principal component analysis (PCA) and PCC, along with ML models such as multi-layer perceptron (MLP) and k-nearest neighbors (kNN). As shown in Table 4, the integrated approach of the A-DBiLSTM

Fig. 6 Overall performance comparison after feature selection for GPS dataset

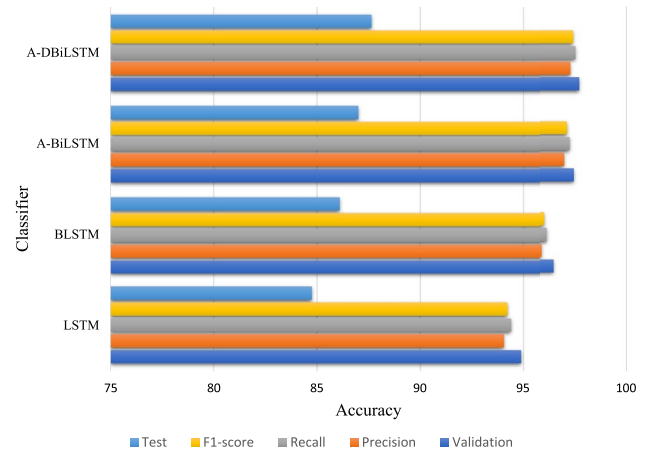


Fig. 7 ROC curve of different DL models with MI feature selection algorithm for GPS dataset

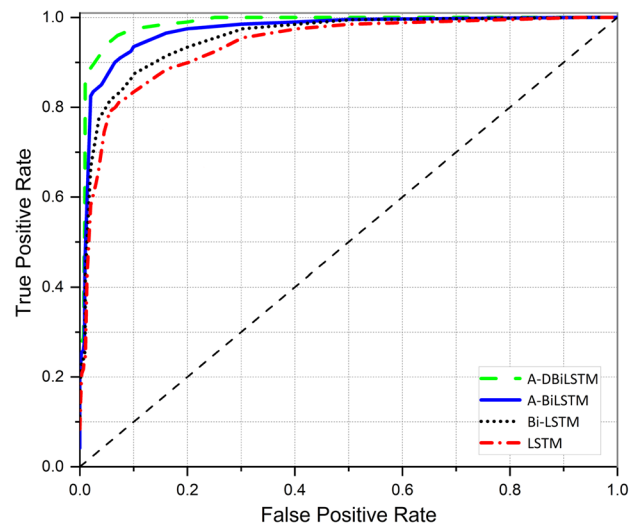


Table 4 Comparison of average accuracy MI using A-DBiLSTM with other existing approaches

| Features | Model | Accuracy (%) | Dataset |
|-----------------|-----------|--------------|--------------------|
| Proposed | A-DBiLSTM | 97.73 | GPS RINEX data |
| PCA [38] | MLP | 93.17 | PDF document |
| PCA | A-DBiLSTM | 95.47 | GPS RINEX data |
| PCC [29] | kNN | 92.6 | Spectrogram images |
| PCC | A-DBiLSTM | 95.67 | GPS RINEX data |

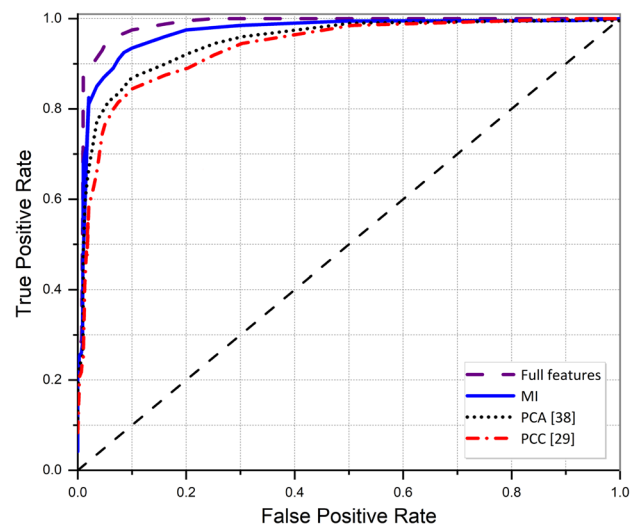
Bold indicates that the corresponding value represents the highest accuracy achieved among all the feature selection algorithms considered

model with MI achieved superior accuracy compared to the previous methods. Moreover, when using the same feature selection algorithm, the A-DBiLSTM DL model outperformed both MLP and kNN regarding detection performance.

In Fig. 7, we have presented a comparison of the ROC curves for the BiLSTM-A model with LSTM, Bi-LSTM, and A-BiLSTM models using different MI feature selection approaches for the GPS dataset. The ROC curves are generated using the Adam optimizer and a batch size of 128. Upon analyzing the ROC curves, we observe that the BiLSTM-A model achieves the highest accuracy by utilizing vectors with the complete set of dimensions. Furthermore, the results indicate that the BiLSTM-A model surpasses the LSTM, Bi-LSTM, and A-BiLSTM models by approximately 2.8%, 1.3%, and 0.4% in terms of accuracy, respectively. This improvement can be attributed to the incorporation of a double-layer Bi-LSTM between AM, enhancing the overall accuracy of the model.

In Fig. 8, we have presented a comparison of the ROC curve for the A-DBiLSTM model applied to the GPS dataset, considering different feature selection approaches. Specifically, we analyze the performance of the model using full features,

Fig. 8 Comparison between the ROC of full features, MI, PCC, and PCA algorithm using A-DBiLSTM model



the MI algorithm, and the PCA algorithm, along with the PCC feature selection algorithm. The results indicate that the full-feature model achieves the highest accuracy, as it utilizes the complete set of features. Additionally, we observe that the MI feature selection algorithm exhibits approximately a 3% higher accuracy compared to the PCA algorithm [38].

Finally, to demonstrate the efficiency of the MI algorithm, we evaluated its impact on the inference training time and accuracy of four DL models using both the MI and full feature approaches on GPS data. Subsequently, we compared the performance of the proposed MI algorithm with the full feature method to assess the extent of accuracy loss resulting from dimensionality reduction. As shown in Tables 2 and 3, applying the MI algorithm resulted in a significant 19% decrease in the inference training time for the four DL models. Furthermore, it effectively reduced the dimensionality of the input feature space, demonstrating its capability to handle high-dimensional data while maintaining performance.

In summary, the experimental research demonstrates that the performance of the A-BiLSTM model is significantly improved by incorporating a Bi-LSTM layer in the decoder phase. Additionally, by utilizing the MI feature selection algorithm before DL training, the inference training time was reduced while maintaining an acceptable level of accuracy.

8 Conclusions

GNSS signal jamming attacks are a common occurrence. In this study, we have developed a GNSS jamming detection system based on DL models, specifically LSTM, Bi-LSTM, A-BiLSTM, and A-DBiLSTM. To train our models, we have utilized two different datasets, GPS and Galileo, which included both pure and jamming signals. We have observed that introducing the AM to the double-layer Bi-LSTM resulted in improved performance compared to the other models. Additionally, we have employed the MI feature selection technique to identify the most crucial features from the datasets, enhancing the accuracy of GNSS jamming prediction. The accuracy and effectiveness of the proposed MI feature selection algorithm have been confirmed through numerical simulations and the use of a confusion matrix. Despite achieving similar training accuracy, MI facilitated dimensionality reduction and a 19% decrease in learning time. As part of our future work, we aim to explore combining multiple feature selection techniques to generate feature subsets that are more directly applicable to interference detection and mitigation.

Author Contributions All authors contributed to the conception of the research problem and discussed the results. AR was responsible for implementing the MI feature selection algorithm for GNSS datasets, conducting simulations for GNSS jamming detection using various DL models, and writing the initial draft of the paper. TM provided constructive feedback and made significant edits that contributed to improving the overall quality of the paper. The final manuscript was reviewed and approved by all authors.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Data Availability The datasets utilized in the analysis of this study were collected specifically for the Ultrahack Galileo innovation challenge [30].

Declarations

Competing interests The authors declare that they have no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Dovic F. GNSS interference threats and countermeasures. London: Artech House; 2015.
2. Mitch RH, Dougherty RC, Psiaki ML, Powell SP, O'Hanlon BW, Bhatti JA, Humphreys TE. Signal characteristics of civil gps jammers. In: Proceedings of the 24th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2011); 2011. p. 1907–1919.
3. Sun K, Chen Y. A novel GNSS sweep interference detection and mitigation method based on Radon–Wigner transform. *IEEE Sens J*. 2023. <https://doi.org/10.1109/JSEN.2023.3240429>.
4. Elmezayen A, Karaim M, Elghamrawy H, Noureldin A. Enhanced GNSS reliability on high-dynamic platforms: a comparative study of multi-frequency, multi-constellation signals in jamming environments. *Sensors*. 2023;23(23):9552.
5. Hu Y, Bian S, Li B, Zhou L. A novel array-based spoofing and jamming suppression method for GNSS receiver. *IEEE Sens J*. 2018;18(7):2952–8.
6. Jang J, Seo S, Ahn W-G, Lee J, Park J. Performance analysis of an interference cancellation technique for radio navigation. *IET Radar Sonar Navigation*. 2018;12(4):426–32.
7. Borio D, Camoriano L, Savasta S, Presti LL. Time-frequency excision for GNSS applications. *IEEE Syst J*. 2008;2(1):27–37. <https://doi.org/10.1109/JSYST.2007.914914>.
8. Sun K, Jin T, Yang D. A new reassigned spectrogram method in interference detection for GNSS receivers. *Sensors*. 2015;15(9):22167–91. <https://doi.org/10.3390/s150922167>.
9. Sun K, Zhang M, Yang D. A new interference detection method based on joint hybrid time-frequency distribution for GNSS receivers. *IEEE Trans Vehicle Tech*. 2016;65(11):9057–71. <https://doi.org/10.1109/TVT.2016.2515718>.
10. Sharifi-Tehrani O, Sabahi MF. Eigen analysis of flipped Toeplitz covariance matrix for very low snr sinusoidal signals detection and estimation. *Digital Signal Process*. 2022;129:103677. <https://doi.org/10.1016/j.dsp.2022.103677>.
11. Sharifi-Tehrani O, Sabahi MF, Raees Danaee M. Efficient GNSS jamming mitigation using the Marcenko Pastur law and Karhunen–Loeve decomposition. *IEEE Trans Aerosp Electron Syst*. 2022;58(3):2291–303. <https://doi.org/10.1109/TAES.2021.3131400>.
12. Na H, Shin Y, Lee D, Lee J. LSTM-based throughput prediction for LTE networks. *ICT Express*; 2021.
13. Mao Q, Hu F, Hao Q. Deep learning for intelligent wireless networks: a comprehensive survey. *IEEE Commun Surv Tuts*. 2018;20(4):2595–621. <https://doi.org/10.1109/COMST.2018.2846401>.
14. Rauber TW, de Assis Boldt F, Varejão FM. Heterogeneous feature models and feature selection applied to bearing fault diagnosis. *IEEE Trans Ind Electron*. 2014;62(1):637–46.
15. Li Y, Pawlak J, Price J, Al Shamaileh K, Niyaz Q, Paheding S, Devabhaktuni V. Jamming detection and classification in OFDM-based UAVs via feature-and Spectrogram-Tailored machine learning. *IEEE Access*. 2022;10:16859–70.
16. Liu X, Xu Y, Jia L, Wu Q, Anpalagan A. Anti-jamming communications using spectrum waterfall: a deep reinforcement learning approach. *IEEE Commun Lett*. 2018;22(5):998–1001.
17. Czech D, Mishra A, Inngs M. A CNN and LSTM-based approach to classifying transient radio frequency interference. *Astron Comput*. 2018;25:52–7.
18. Xiao N, Song Z. Signal interference detection algorithm based on Bidirectional Long Short-Term Memory neural network. *Math Prob Eng*. 2022;22.
19. Zhou Y, Zhang X, Ding F. Hierarchical estimation approach for RBF-AR models with regression weights based on the increasing data length. *IEEE Trans Circuits Syst II Express Briefs*. 2021;68(12):3597–601.
20. Wang Y, Ding F. Novel data filtering based parameter identification for multiple-input multiple-output systems using the auxiliary model. *Automatica*. 2016;71:308–13.
21. Ruiz AP, Flynn M, Large J, Middlehurst M, Bagnall A. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min Knowl Disc*. 2021;35(2):401–49.
22. Lines J, Bagnall A. Time series classification with ensembles of elastic distance measures. *Data Min Knowl Disc*. 2015;29:565–92.
23. Xu Z, King I, Lyu MR-T, Jin R. Discriminative semi-supervised feature selection via manifold regularization. *IEEE Trans Neural Netw*. 2010;21(7):1033–47.
24. Vasconcelos M, Vasconcelos N. Natural image statistics and low-complexity feature selection. *IEEE Trans Pattern Anal Mach Intell*. 2008;31(2):228–44.
25. Shen L, Bai L. Information theory for Gabor feature selection for face recognition. *EURASIP J Adv Signal Process*. 2006;2006:1–11.
26. Xue B, Zhang M, Browne WN, Yao X. A survey on evolutionary computation approaches to feature selection. *IEEE Trans Evol Comput*. 2016;20(4):606–26. <https://doi.org/10.1109/TEVC.2015.2504420>.

27. Liu Y, Mu Y, Chen K, Li Y, Guo J. Daily activity feature selection in smart homes based on Pearson correlation coefficient. *Neural Process Lett.* 2020;51(2):1771–87.
28. Wesson KD, Gross JN, Humphreys TE, Evans BL. GNSS signal authentication via power and distortion monitoring. *IEEE Trans Aerosp Electron Syst.* 2018;54(2):739–54. <https://doi.org/10.1109/TAES.2017.2765258>.
29. Qin W, DAVIS F. Situational awareness of chirp jamming threats to GNSS based on supervised machine learning. *IEEE Trans Aerosp Electron Syst.* 2021;58(3):1707–20.
30. Ultrahack, European GNSS Agency (GSA), and Ublox: Galileo Innovation Challenge (2019). <https://ultrahack.org/galileoinnovationchallenge>. Accessed Nov 2020
31. Gurtner W, Estey L. Rinex-the receiver independent exchange format-version 3.00. Astronomical Institute, University of Bern and UNAVCO, Boulder, Colorado. 2007.
32. Emeç M, Özcanhan MH. A hybrid deep learning approach for intrusion detection in IoT networks. *Adv Electr Comput Eng.* 2022;22(1):3–12.
33. Kshirsagar D, Kumar S. An efficient feature reduction method for the detection of dos attack. *ICT Express.* 2021;7(3):371–5.
34. Bennasar M, Hicks Y, Setchi R. Feature selection using joint mutual information maximization. *Expert Syst Appl.* 2015;42(22):8520–32.
35. Wang Y, Cang S, Yu H. Mutual information inspired feature selection using kernel canonical correlation analysis. *Expert Syst Appl: X.* 2019;4:100014.
36. Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* 2019;31(7):1235–70.
37. Liu G, Guo J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing.* 2019;337:325–38.
38. Zhang J. Machine learning with feature selection using principal component analysis for malware detection: a case study; 2019. [arXiv: 1902.03639](https://arxiv.org/abs/1902.03639)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.