TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/31 pp68–86
DOI: 10.26599/TST.2024.9010004
Volume 30, Number 1, February 2025

Non-Line-of-Sight Multipath Classification Method for BDS Using Convolutional Sparse Autoencoder with LSTM

Yahang Qin, Zhenni Li*, Shengli Xie, Bo Li, Ming Liu, and Victor Kuzin

Abstract: Multipath signal recognition is crucial to the ability to provide high-precision absolute-position services by the BeiDou Navigation Satellite System (BDS). However, most existing approaches to this issue involve supervised machine learning (ML) methods, and it is difficult to move to unsupervised multipath signal recognition because of the limitations in signal labeling. Inspired by an autoencoder with powerful unsupervised feature extraction, we propose a new deep learning (DL) model for BDS signal recognition that places a long short-term memory (LSTM) module in series with a convolutional sparse autoencoder to create a new autoencoder structure. First, we propose to capture the temporal correlations in long-duration BeiDou satellite time-series signals by using the LSTM module to mine the temporal change patterns in the time series. Second, we develop a convolutional sparse autoencoder method that learns a compressed representation of the input data, which then enables downscaled and unsupervised feature extraction from long-duration BeiDou satellite series signals. Finally, we add an $l_{1/2}$ regularizer to the objective function of our DL model to remove redundant neurons from the neural network while ensuring recognition accuracy. We tested our proposed approach on a real urban canyon dataset, and the results demonstrated that our algorithm could achieve better classification performance than two ML-based methods (e.g., 11% better than a support vector machine) and two existing DL-based methods (e.g., 7.26% better than convolutional neural networks).

Key words: convolutional sparse autoencoder; BeiDou Navigation Satellite System (BDS); long short-term memory (LSTM); multipath classification

1 Introduction

Nowadays, the BeiDou Navigation Satellite System (BDS) is an important element of the Global Navigation Satellite System (GNSS), which provides all-weather, all-day, high-precision location services

• Yahang Qin and Zhenni Li are with School of Automation, Guangdong University of Technology, and also with Guangdong-Hong Kong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangzhou 510006, China. E-mail: yahang.qin@gmail.com; lizhenni2012@gmail.com.

• Bo Li is with School of Automation, Guangdong University of Technology, and also with Guangdong Key Laboratory of IoT Information Technology, Guangzhou 510006, China. E-mail: libo999@gdut.edu.cn.

• Shengli Xie is with 111 Center for Intelligent Batch Manufacturing Based on IoT Technology, and also with Key Laboratory of Intelligent Information Processing and System Integration of IoT, Ministry of Education, Guangzhou 510006, China. E-mail: shlxie@gdut.edu.cn.

• Ming Liu is with Department of Electronic and Computer Engineering, Hong Kong University of Science and Technolog, Hong Kong 999077, China. E-mail: eelium@ust.hk.

• Victor Kuzin is with Academician of Russian Engineering Academy, Moscow Moscow 125009, Russia. E-mail: vfkuzin@mail.ru.

∗ To whom correspondence should be addressed.

Manuscript received: 2023-10-25; revised: 2023-12-08; accepted: 2023-12-26

for autonomous driving, intelligent transportation, robot navigation, and other related fields[1−3]. Usually, BDS can reach centimeter-level positioning accuracy in open areas[2]. However, multipath interference dramatically reduces the positioning accuracy of BDS in the so-called “urban canyon”[4]. The multipath interference depends on the satellite elevation angle (ELE), the environment around the receiving antenna, the distance from the reflecting surface to the receiver, the reflection coefficient of the reflecting source, and the antenna characteristics. This leads to multipath interference becoming the main error source in the field of BDS and GNSS high-accuracy positioning[5−8].

In the urban canyon environment, satellite signals develop multiple propagation paths by specular reflection and bypassing. These extra signals interfere with the direct signals from the satellite at the receiver antenna, generating the multipath effect[9]. For the BDS, there are two types of signals causing multipath errors: line-of-sight (LOS) multipath signals and non-line-of-sight (NLOS) signals. NLOS-based errors tend to be worse than LOS-based errors. Examples of LOS and NLOS signals are shown in Fig. 1. The LOS contains only the one direct signal emitted by the satellite, and the NLOS contains only the one signal reflected by the environment near the receiver[10]. LOS multipath can easily reduce the GNSS accuracy to tens of meters. NLOS positioning errors can be several hundred meters above[11]. Therefore, some current studies[11, 12] are focusing on the issue of multipath interference with the aim of mitigating its influence on satellite positioning accuracy.

Exploring how to cope with the satellite positioning offset problem caused by multipath interference has received wide attention. Proposals have mainly been in the areas of hardware upgrading and software processing. Hardware devices such as anti-multipath antennas[13] are used to filter multipath interference, which not only increases the size of the receiving device, but also increases its cost. Meanwhile, sidereal filtering (SF)[14], multipath hemispherical maps (MHM)[15], 3D city models[16], and other software methods have been proposed to mitigate multipath interference. However, the SF method requires precalculation of the satellite operational period. Because the operational period of each satellite varies slowly over time, there is a consequent decrease in the multipath mitigation effect. The MHM method requires a very large matrix of multipath models, which increases the computational load and does not establish a fully accurate multipath model. In addition, real-time updating of 3D city models is difficult.

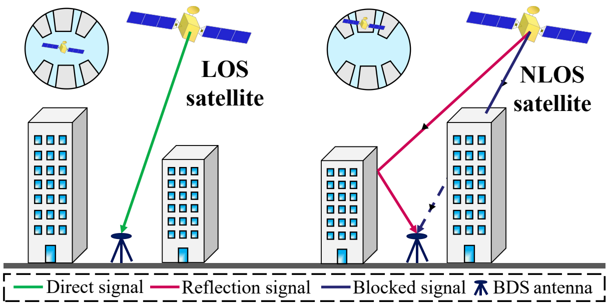


Fig. 1 Two types of BDS signal: LOS and NLOS multipath.

Artificial intelligence (AI) is a rapidly developing technology that can learn multipath signals from sufficient data without the need to build complex mathematical models and shows great potential for identifying multipath in urban canyon environments. These multipath signal-recognition methods can be divided into supervised and unsupervised cases. Supervised algorithms for recognizing multipath signals include decision tree (DT)[17], support vector machine (SVM)[18], k-nearest neighbor (KNN)[19], and neural network (NN)[20]. They have own advantages, and which algorithm to choose for a specific task should be considered according to the data characteristics. For example, SVM has strong generalization ability for small samples, but it is relatively slow in processing large data sets. Overall, supervised learning has two main limitations: the construction of databases containing a large number of labels is difficult, and the recognition accuracy of the models needs to be improved. However, some recent works have attempted to explore the use of unsupervised multipath-signal-recognition methods. For example, the generative adversarial network considers unsupervised domain adaptation (UDA)-based models[21] to reduce the discrepancy between real and simulated data. In Ref. [22], a k-means clustering approach was used to achieve unlabeled multipath signal recognition. Unfortunately, this process includes the problems of high-dimensional data redundancy and low classification accuracy.

In this paper, we propose an unsupervised convolutional sparse autoencoder containing a long short-term memory (LSTM) module to improve the accuracy of NLOS multipath recognition. First, to tackle the twin issues of the high dimensionality of the training samples and the need for a large number of labels, we use a convolutional autoencoder to realize descending and unsupervised signal feature extraction. Then, to learn more fully the information related to the time series of BDS signals, we employ multiple LSTM modules to capture the time series variation patterns of BDS signal features such as carrier-to-noise ratio (C/N_0), ELE, and the pseudorange residual (PR). This enables the model to learn the underlying features of the signal. Furthermore, to deal with the high-dimensional data redundancy in convolutional autoencoder models, we propose to add an $l_{1/2}$ regularizer to the objective function of the unsupervised deep learning (DL) model of a convolutional autoencoder with an LSTM module, which can remove redundant neurons from the NN while obtaining high classification accuracy. Finally, the performance of our model has been evaluated using a real urban canyon dataset, demonstrating superior performance to that of existing models.

The main contributions of this paper can be summarized as follows:

(1) We develop a convolutional autoencoder method that learns a compressed representation of the input data for downscaling and unsupervised feature extraction of long-duration time-series signals containing multipath.

(2) We employ an LSTM module to capture the temporal correlation of the long-duration time-series signals from BeiDou satellites and thereby mine a temporal change law for the time series.

(3) We add an $l_{1/2}$ regularizer to the objective function of an unsupervised DL model of a convolutional autoencoder with an LSTM module to achieve the removal of redundant neurons from the NN while obtaining higher sparsity and recognition accuracy.

(4) We develop a novel convolutional sparse autoencoder and LSTM network model for NLOS multipath classification.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 illustrates the BDS features with different signal types. Section 4 describes the proposed DL networks and is explained in detail. In section 5, we present the environment where the data set is collected, followed by signal type labeling method using the combination of sky mask and code pseudorange double difference observable. Then, the experimental setup and classification results are provided and analyzed in section 6. Furthermore, we present a discussion in this section. Finally, conclusions are drawn in Section 7.

2 Related Work

Numerous works have been studied in the field of multipath identification. The research includes three main kinds of methods: 3D, machine learning (ML), and DL-based methods. In the following, we review the closely related works and highlight the novelty of our contributions.

2.1 3D-based methods

The 3D city models have been widely used in the field of multipath environmental positioning. Typically, a combination of shadow matching and 3D mapping aided (3DMA) was presented in Ref. [23], and the method can identify satellite signal types by building boundaries. The sky and building contours are estimated by fitting data curves to a smooth spline model, and building boundaries are calculated using an adaptive weighting scheme. Reference [24] introduced a shadow-matching classifier and a confidence check to improve the NLOS classifier accuracy. Moreover, exploring how to build applications over the NLOS also achieved wide attention. For example, applying a 3D model to correct positions so that the NLOS contributes to the absolute positioning[25]. Some works use 3D maps and ray tracing to correct the localization[26]. However, 3DMA-based methods rely on the timely updating of 3D models. This poses a great challenge in data-intensive computations. To avoid the limitations of this 3D city model, ML was adopted.

2.2 Machine learning-based methods

Due to the complex and variable architectural structure of the urban environment in which the receiver is located. Making the establishment of accurate physical models is difficult. Nevertheless, ML can mine the underlying features of large amounts of data. The gradient boosted the decision tree (GBDT)[17] was used to classify LOS, multipath, and NLOS with three features: C/N_0, PR, and ELE. Reference [19] compared various ML algorithms such as KNN, NN, SVM, and DT. LOS/NLOS classifiers based on the signal-to-noise ratio (SNR), number of received satellites (NRS), ELE, PR, pseudorange residual percentage (PRP), and normalized pseudorange residual (NRP) are established. In particular, the SVM[18, 20] has played a vital role in multipath detection, where they are used to learn the features of national marine electronics association (NMEA), receiver independent exchange (RINEX), and correlator levels. However, traditional ML methods can only process information within one epoch, and it is easy to lose the data features in the time domain. Thus, some researchers have considered applying 3D techniques to identify multipath.

2.3 Deep learning-based methods

Recently, with the continuous development of AI techniques, the popular DL techniques have been widely adopted in various fields and have achieve outstanding performances. Meanwhile, many works are using the DL techniques to identify NLOS. For example, the fully connected neural networks (FCNNs)[27] and LSTM[28] were applied to the classification of LOS and NLOS. In Ref. [29], a deep convolutional neural network (CNN) was used to detect multipath and further improve the final localisation accuracy. Considering the successful application of CNN in the image field. The mapping of the correlator output signal to a two-dimensional input image is proposed for classification. A deep CNN was used to detect the correlator output multipath[30], and this deep CNN method outperformed the SVM method. DL can be applied not only for signal classification but also for positioning correction. Deep Neural Networks (DNN) is used to learn a set of PR and the satellite LOS vector as the position correction, which outperforms the weighted least square (WLS) baseline on real data[31]. In contrast to traditional model-based methods, DL has shown great potential in solving complex multipath interference problems. However, the training database for multipath studies based on DL is difficult to construct and requires a large amount of labeled data. In addition, the problems of high dimensional data redundancy and low classification accuracy exist in the process of multipath signal recognition.

To tackle the above challenge, in this paper, we proposed an unsupervised convolutional sparse autoencoder with LSTM structures to recognize the NLOS multipath. To address the difficulty of constructing a large number of labeled multipath databases, we proposed an unsupervised deep feature extraction method with a convolutional autoencoder. Also considering the problem of low classification accuracy, we used the LSTM method to extract the relevant information from the time series. To solve the problem of high dimensional data redundancy, we use $l_{1/2}$ regularizer to maintain the classification performance while removing redundant neurons and increasing the model sparsity. Finally, the NLOS signal was classified by the constructed convolutional sparse autoencoder with the LSTM model.

3 Features Selection and Analysis of Collected Data

In this section, we analyze the collected data. By comparing the features of BDS signals, C/N_0, ELE, and PR[14] are selected as the input data. At the same time, we provide a data normalization method.

3.1 Carrier to noise ratio (C/N_0)

The C/N_0 is the ratio of the received carrier signal power to the noise signal and is an important indicator of satellite signal quality. Usually, the receiver in an unobstructed environment receives mostly LOS signals with a high C/N_0. In the complex urban environment, the signal is blocked by high-rise buildings as well as reflections, making the signal change propagation direction and go through multiple paths to reach the receiver, so that the C/N_0 of the NLOS is usually lower.

3.2 Elevation angle (ELE)

The ELE is the magnitude of the angle between the receiver's horizontal plane and the satellite in the sky and is an important indicator of satellite visibility. Generally speaking, in the same environment, the satellite signal interference is usually greater at low angles, so the low angle is more likely to be NLOS. Thus, satellites with high ELE are not easily blocked by high-rise buildings and are usually LOS. The ELE θ can be calculated as

$$\theta^{(i)} = -\arcsin\left(u_D^i/\hat{r}\right) \tag{1}$$

where u_D^i is the component of the distance between the satellite and the receiver in the local Cartesian coordinates coordinate system in the "Up" direction. $\hat{r}$ is the estimated value of the distance between the receiver and the satellite.

3.3 Pseudorange residual (PR)

The PR is the part of the observation after subtracting the useful information, and it is a relatively objective standard for evaluating the quality of the signal. Normally, the PR of LOS is around zero with less jitter. This is due to the fact that LOS is the signal from the satellite that arrives at the receiver via a direct path, without reflections from redundant paths. Conversely, the NLOS signal quality is poor and the PR is generally larger. The position of the receiver is related to the PR measurements and the satellite position.

Satellite positions can be obtained directly from the broadcast ephemeris. The state of the receiver can be calculated using a PR equation estimated by the least squares method, which can be expressed as

$$\gamma = (G^{\mathrm{T}}G)^{-1}G^{\mathrm{T}}\rho \tag{2}$$

where γ is the receiver state. ρ denotes pseudorange. G denotes the matrix consisting of the unit LOS vector between the satellite and receiver, which can be expressed as

$$G = \begin{bmatrix} u_N^{(1)} & u_E^{(1)} & u_D^{(1)} & -1 \\ u_N^{(2)} & u_E^{(2)} & u_D^{(2)} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ u_N^{(i)} & u_E^{(i)} & u_D^{(i)} & -1 \end{bmatrix} \tag{3}$$

Once the pseudorange and satellite states are obtained, the PR can be expressed as

$$\eta = \rho - G\cdot\gamma \tag{4}$$

In addition to the above features, the raw data consists of the horizontal dilution of precision (HDOP), vertical dilution of precision (VDOP), azimuth angle (AZ), consistency between delta pseudorange and pseudorange rate (ζ) and Number of visible satellites (NS), geometric dilution of precision and the doppler shift frequency in the RINEX format. We explore which features contribute more to deep learning based classifier. When more features are used, the accuracy decreases greatly for most models. This may be due to the fact that these features are not strongly associated with the type of signal, e.g., sky occlusion varies in different environments, and it is not possible to determine the type of signal from a high or low azimuth angle.

3.4 Normalization of features

Due to we adopted a Convolutional AutoEncoder (CAE) to extract features, the inputs and outputs of this structure should have the same range of values. Meanwhile, the range of the selected features varies widely. Normalization can eliminate the influence of the different ranges of data to ensure the stable convergence of the weights and deviations. In addition, data without normalization will slow down the training speed of the network. Therefore, the input data features C/N_0, ELE and PR need to be normalized so that they remain between [0−1]. We adopt the min-max normalization method, which can be expressed as:

$$x_i^* = \frac{x_i - \min\limits_{0\leqslant j\leqslant n}\left\{x_j\right\}}{\max\limits_{0\leqslant k\leqslant n}\{x_k\} - \min\limits_{0\leqslant j\leqslant n}\left\{x_j\right\}} \tag{5}$$

where x_i is any value in the input sample, $\max\limits_{0\leqslant k\leqslant n}\{x_k\}$ and $\min\limits_{0\leqslant j\leqslant n}\left\{x_j\right\}$ denote the maximum and minimum values of the sample, respectively.

4 Non-Line-of-Sight Multipath Classification Method for BDS Using Convolutional Sparse Autoencoder with LSTM

In this section, we first introduce the convolutional sparse autoencoder network model with an LSTM module for identifying BDS signals. Next, we describe the objective function of the convolutional sparse autoencoder. Furthermore, We also present the details of this training process of the proposed algorithm.

4.1 Convolutional sparse autoencoder with LSTM neural network architecture

In this part, we first present the overall process of the proposed method. Meanwhile, each process of our method from the original signal to the BDS signal classification is described in detail. Then, the basic principles of convolutional sparse autoencoder with LSTM module are presented.

4.1.1 Proposed method

The identification of single-moment BDS signal type makes it easy to lose time correlation information, and the extraction of BDS signal time series features is of great significance for signal identification. DL usually requires a large amount of labeled data, which is a difficult task. In contrast, unsupervised learning networks require less labeled data. Autoencoder networks, as an important unsupervised learning network can achieve downscaling and feature extraction of signals using only unlabeled data. However, the training process of autoencoder networks is prone to overfitting and there is a possibility that the output layer simply copies the input layer.

To address the above problems, a convolutional sparse autoencoder with an LSTM module is proposed for the recognition of BDS signals. The proposed method structure is shown in Fig. 2. First, a convolutional sparse autoencoder is used to learn the compressed representation of the input data. Second, The output feature data of the encoder is fed to the LSTM layer to learn the sequential dependencies between the feature data. Then, an $l_{1/2}$ regularizer sparsity constraint is added to the objective function of the method to further reduce the redundancy of the neural network. Unsupervised high-level feature extraction of BDS signals can be achieved with this network. Finally, to further improve the classification accuracy of the proposed method, we replace the softmax layer of the last layer of the autoencoder module with a decision tree to form a hierarchical network, which in turn outputs the NLOS/LOS class of BDS signals. In Fig. 2, the encoder part of the one-dimensional convolutional sparse autoencoder network used in this paper consists of three convolutional layers and three maximum pooling layers. Each convolutional layer uses Rectified Linear Unit (ReLU) as the activation function. The first convolutional layer comprises 16 filters, the subsequent convolutional layer comprises 32 filters, and the final convolutional layer comprises 64 filters. The kernel size of all convolutional layers is set to 1 × 2.

The decoder part of the convolutional sparse autoencoder replaces the original pooling layers with inverse pooling layers and consists of three one-dimensional convolutional layers and three upsampling layers. The number of filters is the opposite of the encoder part, 64, 32, and 16, respectively. The kernels of all convolutional layers are also set to 1 × 2. The 1D convolutional layers are mainly responsible for extracting data features from the time series. Maximum pooling is a down-sampling operation that retains the maximum value in each channel of the output of the previous layer, which can effectively avoid over-fitting. The maximum pooling means that only the strongest of these features are retained while discarding other weak features. The network parameters can be effectively reduced to prevent over-fitting of the model. This pyramidal architecture can eliminate redundant features while learning a compressed representation of the BeiDou satellite series signal.

4.1.2 Convolutional autoencoder

The autoencoder is a typical representation learning

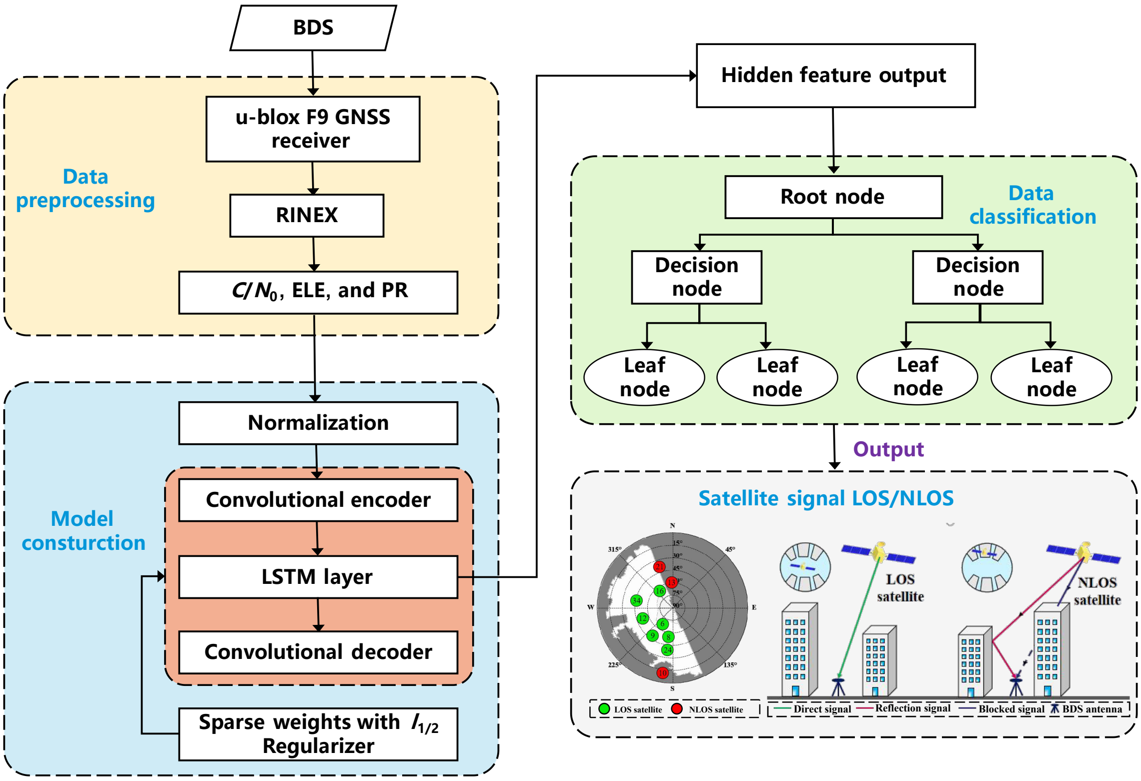


Fig. 2 Convolutional sparse autoencoder with LSTM module based BDS series NLOS recognition.

algorithm that can achieve unsupervised feature extraction from data. An autoencoder consists of an input layer, a hidden layer, and an output layer forming an encoder and a decoder. The network is trained by a backpropagation algorithm to make the output equal to the input[32]. In the encoder part of a recognition network, the hidden layer is responsible for learning the compressed representation of the input layer data. In the decoder part of a generative network, the output layer is responsible for reconstructing the input data from the hidden layer. The structure of the autoencoder is shown in Fig. 3.

The process of reconstructing the input data by the autoencoder can be briefly described as follows. First, for the input data $x \in \mathbf{R}^n$, the hidden layer feature $h \in \mathbf{R}^k$ $(k < n)$ can be obtained by the encoder. Then the output layer in the decoder reconstructs the output $\hat{x} \in R^n$ from the hidden layer. However, autoencoder often leads to the degradation of feature learning performance when dealing with high-dimensional data due to the excessive amount of parameters. CNN has the features of local connectivity and weight sharing, which can reduce the network parameters and speed up the computation of the network. And the unsupervised learning feature of the autoencoder can prevent the network from overfitting. Moreover, there are local correlations in continuous data. CNN can perceive richer local features of the signal using convolutional kernels. Convolutional sparse autoencoder combined with operations such as convolution and pooling in CNN can achieve unsupervised training. One-dimensional convolution is processed on BeiDou satellite time series data to achieve feature extraction to generate new feature maps. The convolutional sparse autoencoder network is trained many times so that the Beidou satellite signal data output from the decoder can approximately replicate the input data from the encoder.

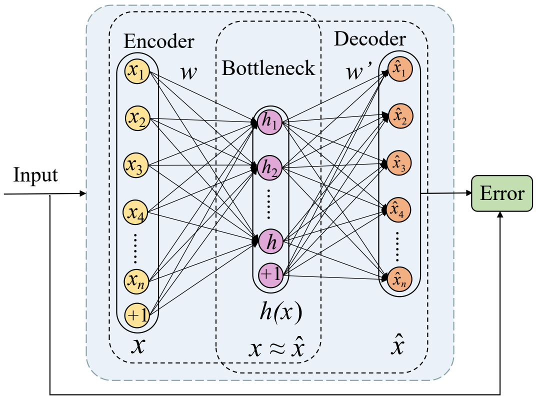


Fig. 3 Network structure of autoencoder.

4.1.3 LSTM module

LSTM can effectively solve the problems of gradient explosion, gradient disappearance, and the inability to preserve historical information for a long time during the training process[33]. Therefore, we adopt LSTM to solve the problem of long distance dependence of time series data. As shown in Fig. 4, each module contains the previous module's hidden state h and cell state C. Among them, the hidden state h can achieve short-term memory and the cell state C can achieve long-term memory. Satellite features have a high degree of temporal correlation, and LSTM can effectively extract the temporal information of the features, thus improving the recognition capability. The internal structure of the LSTM neural network is shown in Fig. 4. The retention and forgetting of historical information is achieved by three gating units. The forget gate determines which information needs to be deleted from the cell state. The input gate determines which new information is added to the cell state. The output gate determines which state features of the cell are output. The computational process of LSTM can be expressed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{6}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{7}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{8}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{9}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{10}$$

$$h_t = o_t * \tanh(C_t) \tag{11}$$

In the method proposed in this paper, the LSTM memory module in the forgetting gate f_t obtains the BDS signals at time t. The information that can reflect

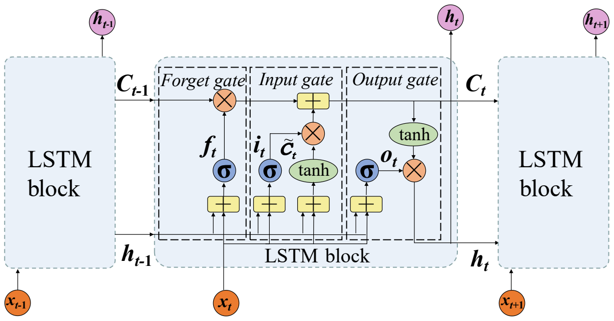


Fig. 4 Structure of the LSTM neural network.

the characteristics of the satellite signal type is memorized using the sigmoid function, and the information that cannot reflect the characteristics of the satellite signal type is forgotten. The sigmoid in the input gate $i * \tilde{C}_t$ will determine which important information in the output value of tanh that can reflect the type of BDS signal is to be retained. Cell state C_t allows the new BDS signal information found by the neural network to be updated into the cell state. The output gate o_t is used to determine the BDS signal information that the hidden state h_t should carry. The hidden state is used as the output of the current cell. At the same time, the new cell state and the new hidden state are passed to the next time step.

4.2 Objective function with the $l_{1/2}$ regularizer

The objective function of the sparse neural network is constructed by imposing a sparse constraint, which can be expressed as

$$J(\omega) = \min_{\omega} \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i;\omega)) + \lambda\Omega(\omega) \tag{12}$$

where $y = f(x_i;\omega)$ is represented as a generic deep neural network with a vector $x \in \mathbf{R}^d$ as input. After propagation through the neural network weight matrix ω formed by the hidden layers, the output vector $y \in \mathbf{R}^0$ is obtained. $L(\cdot,\cdot)$ is denoted as the loss function, $\Omega(\cdot)$ denotes the canonical constraint imposed by the network, and $\lambda \in (0,1)$ denotes the parameter used to balance the loss and canonical terms.

The objective function of the $l_{1/2}$ regularizer method in our study is

$$J(\omega) = \min_{\omega} \frac{1}{N}\sum_{i=1}^{N} (y-x)^2 + \lambda\|\omega\|_{\frac{1}{2}}^{\frac{1}{2}} \tag{13}$$

where (and henceforth) $\|\cdot\|$ denotes the Euclidean norm, ω is the weight matrix, x is the input signal, $x = \{x_1, x_2, ..., x_N\}^{\mathrm{T}} \in \mathbf{R}^N$, y is the output signal, $\lambda > 0$ is a regularization parameter, and $\lambda\|\omega\|_{\frac{1}{2}}$ is a regularization term.

To solve the nonconvex and nonsmooth $l_{1/2}$ norm regularized problem, we use the proximal gradient descent method[34] to solve Eq. (13), find its gradient, and set it to zero to get Eq. (14):

$$\omega_i - a_i + \frac{\lambda}{2}\nabla\left(|\omega_i|^{\frac{1}{2}}\right) = 0 \tag{14}$$

where $a_i = \omega_i - \nabla_{\omega_i}(y-x)^2$. Because $\nabla\left(|\omega_i|^{\frac{1}{2}}\right) = \frac{\mathrm{sign}(\omega_i)}{2\sqrt{|\omega_i|}}$, we have

$$\omega_i - a_i + \frac{\lambda\mathrm{sign}(\omega_i)}{4\sqrt{|\omega_i|}} = 0 \tag{15}$$

for any fixed i, the solution ω_i^* must be satisfied $\omega_i^* a_i > 0$. Next, we simply consider the case where $\omega_i a_i > 0$ for any i.

Case 1: $\omega_i > 0$: At this point, we can make $\sqrt{|\omega_i|} = z$. Thus we can obtain $\omega_i = z^2$. Thus Eq. (15) can be written in the form of a cubic algebraic of Eq. (16):

$$z^3 - a_i z + \frac{\lambda}{4} = 0 \tag{16}$$

For the cubic algebraic equation of Eq. (16), the solution can be found using the Cardano formula[35]. With the different signs a_i, the solution of Eq. (14) is different. This can be written as $r = \sqrt{\frac{|a_i|}{3}}, P = -\frac{a_i}{3}$, and $q = \frac{\lambda}{8}$.

When $\frac{q^2}{4} + \frac{p^3}{27} < 0$, $a > \frac{3}{4}\lambda^{\frac{2}{3}}$, $\phi_\lambda = \arccos\left(\frac{q}{2r^3}\right)$, the solution to Eq. (16) can be expressed as $z_1 = -2r\cos\left(\frac{\phi_\lambda}{3}\right), z_2 = 2r\cos\left(\frac{\pi}{3} + \frac{\phi_\lambda}{3}\right)$, and $z_3 = 2r\cos\left(\frac{\pi}{3} - \frac{\phi_\lambda}{3}\right)$.

Since $\omega_i > 0$, so z_1 less than zero is not a solution of Eq. (16), and z_2, z_3 greater than zero is a solution of Eq. (16). Further, analyzing z_2, z_3, because $z_2 > z_3$, z_3 is the only solution to Eq. (16). Therefore, in this case, Eq. (16) has a unique solution $\omega_i^* \in \mathbf{R}^N$. This unique solution can be written as

$$\omega_i^* = \frac{2}{3}|a_i|\left(1 + \cos\left(\frac{2\pi}{3} - \frac{2\phi_\lambda(a_i)}{3}\right)\right) \tag{17}$$

where $\phi_\lambda(a_i) = \arccos\left(\frac{\lambda}{4}\left(\frac{|a_i|}{3}\right)^{-\frac{3}{2}}\right)$.

Case 2: $\omega_i < 0$: At this point, we can make $\sqrt{|\omega_i|} = z$. Thus we can obtain $\omega_i = -z^2$. Thus Eq. (13) can be written in the form of a cubic algebraic Eq. (18):

$$z^3 + a_i z + \frac{\lambda}{4} = 0 \tag{18}$$

Following the Cardano formula for solving cubic algebraic equations and performing an analysis similar to that of the first case for Eq. (18), we obtain the unique solution:

$$\omega_i^* = -\frac{2}{3}|a_i|\left(1 + \cos\left(\frac{2\pi}{3} - \frac{2\phi_\lambda(a_i)}{3}\right)\right) \tag{19}$$

When $\omega_i = 0 \Leftrightarrow |a_i| \leqslant \frac{3}{4}\lambda_i^{\frac{2}{3}}$.

By analyzing the above two cases, we can obtain the solution of Eq. (14) as

$$\omega_i^* = \begin{cases} \dfrac{2}{3}|a_i|\left(1+\cos\left(\dfrac{2\pi}{3}-\dfrac{2\phi_\lambda(a_i)}{3}\right)\right), & a_i > \dfrac{3}{4}\lambda^{\frac{2}{3}}; \\ -\dfrac{2}{3}|a_i|\left(1+\cos\left(\dfrac{2\pi}{3}-\dfrac{2\phi_\lambda(a_i)}{3}\right)\right), & a_i < -\dfrac{3}{4}\lambda^{\frac{2}{3}} \end{cases} \tag{20}$$

Therefore, the weight parameter ω in the objective function will be updated according to Eq. (20).

4.3 Proposed convolutional sparse autoencoder with LSTM algorithm

We proposed a new method for BDS signal recognition that combines a convolutional sparse autoencoder and an LSTM network. During the training phase, the encoder can learn the compressed representation of the input data, and the decoder implements the reconstruction of the input data. The LSTM layer embedded at the end of the encoder learns the temporal features between the data. We design the mean loss update parameter with an $l_{1/2}$ regularizer thereby suppressing redundant information. We use DT to complete the classier. Herein, the details of this training process are summarized in Algorithm 1. First, a convolutional sparse autocodeer with LSTM is trained with the original input data x. This allows the encoder f_θ to output h_j^l and obtain the parameters θ_i. Next, the LSTM learns the temporal correlation of the h_j^l generated by the encoder and outputs the dimensionality-reduced data h_t. Furthermore, h_t obtains the reconstructed data y after going through the decoder f_ϕ function and updating the parameters ϕ_i. Then, we use the backpropagation algorithm to compute the gradient of the loss function J_i of the weights and deviation vectors and optimize the autoencoder with a stochastic gradient descent algorithm. Thereby each sample x can be mapped to a new feature space h_t. Finally, the DT model is used to classify and output the BDS signal type.

Algorithm 1 Training the convolutional sparse autoencoder with LSTM

```
Input: Training input data $x = \{x_1, x_2, ..., x_N\}^{\mathrm{T}} \in \mathbf{R}^N$, positive
       parameters $\lambda$.
Output: The type of the BDS signal.
$h_j^l = [h_1, h_2, ...h_M] \in M^l$, where $h_i$ is the number of hidden units
in layer $i$ and $l$ is the number of hidden layers.
$y = [y_1, y_2, ...y_z] \in Z^l$, where $y_i$ is the number of output units in
layer $i$ and $l$ is the number of output layers.
$\theta = [\theta_1, \theta_2, ...\theta_z]$, where $\theta_i = \left\{W_{ij}^l, b_j^l\right\}$
$\phi = [\phi_1, \phi_2, ...\phi_z]$, where $\phi_i = \left\{\hat{W}_{ij}^l, \hat{b}_j^l\right\}$
$(\theta, \phi) \longleftarrow$ initialize
while Stochastic gradient descent not converged do
  $\mathcal{M} \sim \mathcal{X}$ (sample minibatch of data)
      for $x_i \in \mathcal{M}$ do
          $h_j^l = f(\sum_{i \in M_j} x * W_{ij}^l + b_j^l)$
          $h_t \longleftarrow \mathrm{LSTM}\left(h_j^l = x_t\right)$
          $h_j^{l-1} = h_t$
          $y = f(\sum_{i \in N_j} h_j^{l-1} * \hat{W}_{ij}^l + \hat{b}_j^l)$
      Let the loss J be defined as (using Eq. (13))
      $J_i \longleftarrow \min_\omega \frac{1}{N} \sum_{i=1}^N (y-x)^2 + \lambda \|\omega\|_{\frac{1}{2}}^{\frac{1}{2}}$
      Compute stochastic gradient of the loss w.r.t each $\omega$
      using $l_{1/2}$ regularizer trick (Eq. (20)).
      $\nabla \bar{\mathcal{J}} = \nabla_{\theta,\phi}\left(\frac{1}{|\mathcal{M} \sum J_i|}\right)$
      Update network parameters by backpropagation
      $(\theta, \phi) \longleftarrow$ Stochastic gradient descent optimizer
      for $h_j^l = [h_1, h_2, \ldots, h_M] \in M^l$ do
          To classification on the data $h_j^l$ using DT
      return The type of the BDS signal, $\theta$, $\phi$
```

5 Experiment

In this section, we first describe the BDS signal collection scheme. Second, we illustrate the urban complex environment of BDS signal acquisition. Then, we depicted the BDS signal labeling method. Finally, the data preprocessing is given in Section 5.3.

5.1 Data collection

Figure 5 shows the BDS signal collected device, which consists of a fisheye camera, a tripod, a pair of GNSS antennas, two u-blox F9 GNSS modules, a laptop, two rechargeable batteries, and a small portable foldable table. Above the base, the fisheye camera can be installed on the tripod, and other devices can be placed on the table. Due to the double differential approach that can get more accurate positioning results, we use two GNSS antennas for capturing the BDS signals, one of which is used as a base station and the other as a mobile station. The u-blox F9 GNSS module is used for high-precision positioning, and it requires a rechargeable battery for continuous power supply. In addition, the laptop is used to store the raw BDS data.

The BDS signals were collected at four different

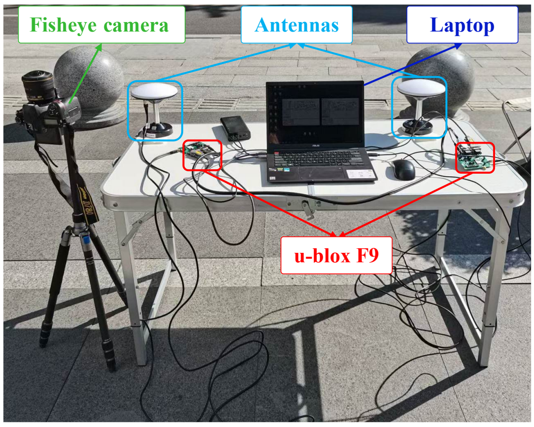


Fig. 5 Data collection equipment.

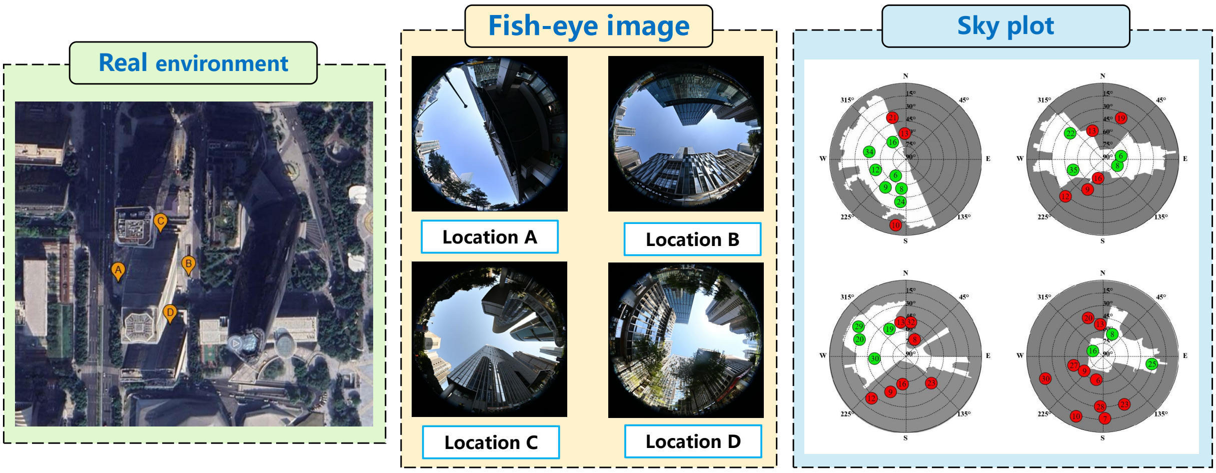
locations in Tianhe, Guangzhou. This environment is densely populated with high-rise buildings, and many of them are higher than 80 meters. Therefore, we can collect a large number of NLOS. From October 10 to 13 in 2022, we collected data from four Locations, A, B, C, and D, using the equipment in Fig. 5, and each location collected 4 hours of static data of BDS L1 signals in RINEX format with the sampling frequency set to 1 s. Thus, four BDS signal datasets (A, B, C, and D) were constructed.

Figure 6b shows a sky image of the data collection location, where grey is the area shaded by buildings and white is the sky area in the figure. Our data include four different sky obscuration scenarios. Generally speaking, the more sky occlusion area, the easier the satellites are occluded by buildings, and there are fewer LOS signals in the dataset. One side, two sides, three sides and four sides of the sky occlusion cases are selected in the experiment, which includes the typical sky occlusion cases in urban canyons, and effectively avoids the problem of poor model generalisation ability brought by a single data distribution. Location A corresponds to a situation where one side is blocked by a building and the other three sides are open. Similarly, Location B corresponds to a situation where two sides are blocked. Location C corresponds to a situation where three sides are blocked by the building and the other sides are open. Location D represents a deep urban environment blocked by four sides. This location contains more NLOS due to building blockage.

5.2 Data labeling

Labeling the true signal type on BDS data collected for DL is necessary. In this study, a fisheye camera is used to assist in data labeling, and the labeling is achieved by the method of oprojections. The flow of data labeling is shown in Fig. 7. First, a sky image with building boundaries and sky regions is captured by a fisheye camera. Moreover, the sky image is segmented into sky area and non-sky area using an image segmentation algorithm, where the non-sky area corresponds to the part obscured by buildings. Finally, the ELE and AZ of the satellites acquired by the receiver are projected onto the fisheye camera sky image using an isotropic projection method, and the satellites falling in the non-sky regions are NLOS satellites and those falling in the sky region are LOS satellites.

An example of the method shown above is given in Figs. 6b and 6c, where the original sky image taken by the fisheye camera is shown in Fig. 6b, and the sky image after satellite projection is shown on Fig. 6c. The



(a) Real environment (b) Fish-eye image (c) Sky plot

Fig. 6 BDS data collection in an urban environments. (a) Real environment; (b) Fish-eye image; (c) Sky plot.

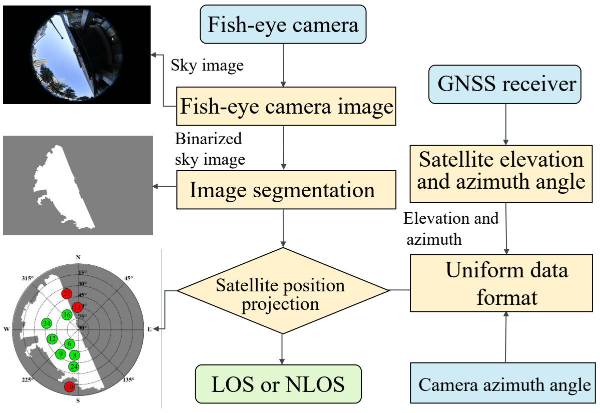


Fig. 7 Flowchart of fish-eye image-aided determination of LOS or NLOS satellites.

white area corresponds to the sky, and the black color indicates the buildings. Thus, the white areas are LOS satellites, marked with green dots. The black areas are NLOS satellites, marked with red dots.

5.3 Data preprocessing

We collected 4 hours of BeiDou satellite data at each location, which in turn created a signal classification dataset that can be used by the proposed unsupervised DL model. To meet the requirements of the input data format of our proposed algorithm, we use a sliding window to split the BDS signal data. Specifically, a window of length n and sliding step m is used to intercept the time series by sliding to the right one by one, with each window n corresponding to one sample. Figure 8 shows the time series data format we used. Green is 4 hours of BDS signal data collected at the same location. In purple is the length of a single sample, chosen to be 128 s. A single sample of data is slid along the yellow arrow to the right to intercept the time series, and the sliding step is chosen to be 10 s. The data is labeled at the last moment. Thus, the construction of the BeiDou satellite time series data set is completed.

Table 1 shows the specific form of the input data after pre-processing, where n is the time stamp. Our proposed NLOS signal classification method will identify the signal type of the last epoch based on the characteristics of the signal in each sample. However, the value of n is not as small as possible. Too small n will lead to insignificant sample differentiation in different types of data, and the model will have difficulty in achieving better recognition results. Here the input data length is taken as 128 s, and the sampling frequency of this paper is 1 Hz, thus a sample contains 128 epochs. In this paper, the sliding window is chosen to be 10, and the sliding window selection process is described in detail in the next section.

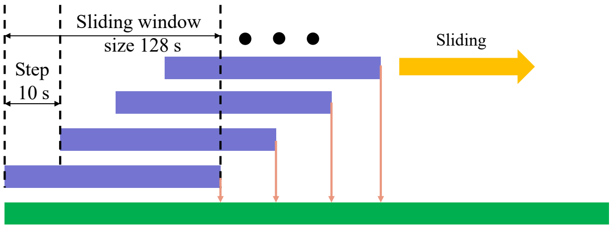


Fig. 8 BDS signal data format.

Table 1 Description of the time series data structure.

Number of epoches	Feature		
	C/N_0	ELE	PR
1	CNR_1	ELE_1	PR_1
2	CNR_2	ELE_2	PR_2
...	...	...	...
n−1	CNR_{n-1}	ELE_{n-1}	PR_{n-1}
n	CNR_n	ELE_n	PR_n

6 Experimental Result and Discussion

In this section, we mainly show the experimental results. First, we describe the effects of different input data lengths and sliding window sizes. Second, the $l_{1/2}$ regularizer is compared with several commonly used regularization, and introduces the parameter adjustment of the regularization. Next, several common evaluation methods for assessing the performance of classification methods are discussed. Then, we conduct comparative experiments using supervised learning methods such as DT, SVM, and CNN. Finally, sparsity analysis experiments are implemented for the $l_{1/2}$ regularizer.

6.1 Data parameter analysis

To optimize the performance of the the proposed method, adjustments to some parameters are necessary. The data format we use is the BDS signal time series. Therefore, we conduct experiments on the selection of the appropriate BDS signal time series length on the four Datasets A, B, C, and D. The results of the experiments are shown in Fig. 9.

We can find the classification accuracy of the datasets from Locations A, B, C, and D is comparable when the input data length is 16 s, and the accuracy is relatively low. When the input data length is increased to 32 s, the classification accuracy improves slightly, albeit not significantly. However, as the length of the

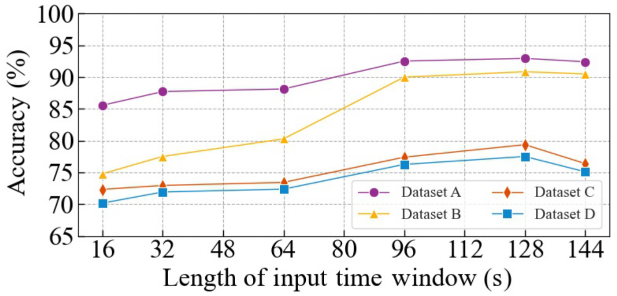


Fig. 9 Classification results of different window lengths.

BDS signal time series is further increased to 48 s, 64 s, 80 s, and 96 s, the classification performance of the proposed method progressively improves. An increase in the input BDS time series from 96 s to 128 s results in a smaller improvement in the classification accuracy. Subsequently, as the length of the time series data continues to increase to 144 s, the classification accuracy begins to decrease. Overall, the experimental results demonstrate that 128 s is the optimal length for the BDS signal time series. The input time series data length being too short results in an incomplete expression of the temporal correlation in the data. Conversely, the length of the input time series data is too long, and the data is easily affected by the environmental noise of the urban canyon. This leads to the proposed method of learning noisy information about the data features, which reduces the classification accuracy.

In addition to the time series length, choosing an appropriate sliding window is equally important. We determined this parameter by the experiences. Figure 10 reveals that the accuracy of the method proposed in this paper is poor on all four datasets when the sliding step size is set to 5 seconds. However, the accuracy of the method on all four datasets shows a significant improvement when the sliding step is increased to 8 seconds. Furthermore, the proposed method attains the highest classification accuracy on all datasets when the sliding step is set to 10 seconds. Nevertheless, when the sliding step length of the BDS time series data is increased to 12 seconds, the classification accuracy decreases slightly. Moreover, when the sliding step is further increased to 16 seconds, the classification accuracy decreases even further. Therefore, based on these experimental results, it can be concluded that 10 seconds is the most appropriate sliding step for the proposed method. It should be noted that the experimental outcomes reveal that if the sliding window of the time series data is too short, the data fluctuation is small, and the representation of data types with less distinctive features. On the other hand, if the sliding step of time series data is too long, it cannot extract more critical feature information of the BDS signal. Consequently, a balance must be struck to achieve optimal performance.

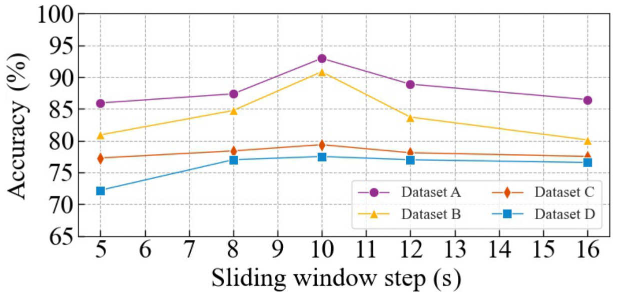


Fig. 10 Classification results of different sliding steps.

6.2 Regularizer parameter analysis

Our proposed model employs the $l_{1/2}$ regulariser, mainly because the $l_{1/2}$ regulariser is easier to solve than the classical l_0 regulariser. The $l_{1/2}$ regulariser produces sparser solutions than the now popular l_1 regulariser. Experiments have shown that the $l_{1/2}$ regulariser can replace the l_p $(0 < p < 1)$ regulariser, which has an important and wide applications.

We compare the proposed regularizer with some different regularizers, including the LOG regularizer[36, 37], and the MCP regularizer[38]. We chose MCP and LOG as the comparison methods. This is due to the fact that the l_1 and l_2 regularizers compress not only the smaller weights to 0, but also the larger ones, resulting in biased solutions. To compensate for this deficiency, the nonconvex regularisers MCP and LOG are proposed, which compress smaller weight values close to 0 while obtaining approximate unbiased estimates for larger weights. Thus, the non-convex regularisation methods are more robust and are able to obtain more accurate features. These regularization methods were used to sparsely constrain the objective function of the convolutional sparse autoencoder with the LSTM module. As shown in Fig. 11, we can see that the use of the $l_{1/2}$ regularizer yielded the highest classification accuracy across all datasets. The LOG regularizer was slightly less effective than the $l_{1/2}$ regularizer, while the MCP regularizer produced the lowest classification accuracy. Therefore, our results show that the $l_{1/2}$

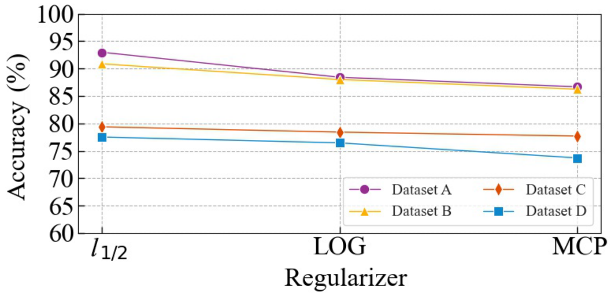


Fig. 11 Classification results of different regularizers.

regularizer is the most appropriate regularization method for this particular model.

Furthermore, the $l_{1/2}$ regularizer approach involves a regularization parameter λ, that must be adjusted for the optimal performance of the regularization technique. In this study, λ was set to 0.08, 0.1, 0.12, 0.14, and 0.16, and the resulting classification accuracy of the the proposed method was evaluated across four environmental datasets, as depicted in Fig. 12. The figure indicates that when λ is set to 0.08, the method's accuracy is marginally lower than when λ is set to 0.1. Notably, the classification accuracy of our method is highest when λ is set to 0.1 across all datasets. However, as λ increases beyond 0.1, the method's accuracy decreases, suggesting that λ of 0.1 is the most optimal parameter for the $l_{1/2}$ regularizer method.

6.3 Evaluation metrics

We employ accuracy, precision, recall, and F1-score to evaluate the proposed method's performance. Accuracy and recall were used to evaluate the classification methods concerning a particular aspect of actual or predicted values to measure their efficacy. Given that the sum of false positives (FP) and false negatives (FN) is a constant value, it is not achievable to enhance both precision and recall evaluation metrics. The F1-score, which integrates both precision and recall into a single metric, is an alternative evaluation metric that enables a more objective assessment of the classification method.

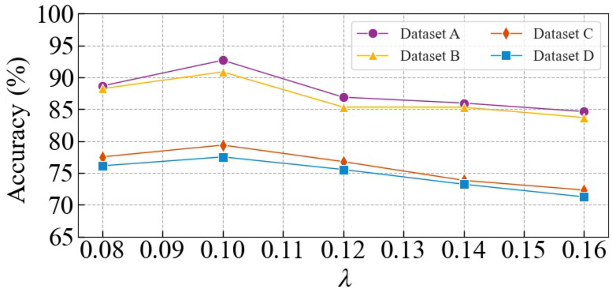


Fig. 12 Classification results of different λ.

Accuracy pertains to the fraction of accurately classified samples from the entire pool of samples. Precision represents the ratio of the accurately classified positive samples to the positive samples recognized by the classifier. The recall is the measure of the accurately classified positive samples relative to the actual positive samples. The F1-score is an average of the precision and recall rates. By utilizing the pertinent parameters, the formulas for accuracy, precision, recall, and F1-score can be derived as follows:

$$\text{Overall accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{21}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{22}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{23}$$

$$\text{F1-score} = 2\frac{\text{R}\cdot\text{P}}{\text{R}+\text{P}} \tag{24}$$

where TP, TF, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives, respectively. Furthermore, R and P represent recall and precision, respectively.

6.4 Performances analysis

To assess the effectiveness of the proposed method for recognizing BDS signals, we compare the proposed method with the existing methods, including DT[17], SVM[18], CNN[29], and Convolutional Sparse Autoencoder (CSAE). The first two comparison methods are based on traditional ML, and the latter two are based on DL architecture. To be specific, the DT method adopts a CART-type structure for classification. The CNN method uses 3 convolutional layers to obtain features and a softmax classifier for the output layer. The structure of the CSAE method is similar to the classifier part of this paper but does not include the LSTM module.

The evaluation results are listed in Tables 2−5. we can see that our proposed method surpasses other methods in terms of accuracy, precision, recall, and F1-score performance metrics across four different locations. Specifically, our proposed method achieved accuracy of 92.96%, 90.85%, 79.37%, and 77.51% on the datasets of the four locations, respectively. Compared to traditional ML algorithms and common CNN methods, our proposed algorithm yielded

Table 2 Performance comparison of Location A.

Method	Evaluation index			
	Acc (%)	F1-score (%)	Prec (%)	Recall (%)
DT[17]	81.40	67.56	67.91	67.24
SVM[18]	82.23	45.12	41.12	50
CNN[29]	85.70	74.64	75.61	73.80
CSAE	90.69	79.58	82.78	77.20
Proposed method	**92.96**	**84.72**	**87.29**	**82.62**

Table 3 Performance comparison of Location B.

Method	Evaluation index			
	Acc (%)	F1-score (%)	Prec (%)	Recall (%)
DT[17]	86.50	86.12	86.30	85.98
SVM[18]	73.04	72.24	72.40	72.13
CNN[29]	86.96	86.63	86.68	86.59
CSAE	88.89	88.54	88.85	88.31
Proposed method	**90.85**	**90.56**	**90.83**	**90.34**

Table 4 Performance comparison of Location C.

Method	Evaluation index			
	Acc (%)	F1-score (%)	Prec (%)	Recall (%)
DT[17]	75.63	68.22	68.45	68.02
SVM[18]	73.67	42.42	36.83	50
CNN[29]	77.56	69.90	70.04	69.78
CSAE	78.35	71.17	71.59	70.80
Proposed method	**79.37**	**72.40**	**72.78**	**72.06**

Table 5 Performance comparison of Location D.

Method	Evaluation index			
	Acc (%)	F1-score (%)	Prec (%)	Recall (%)
DT[17]	74.69	74.63	74.61	74.65
SVM[18]	64.52	61.36	67.78	63.10
CNN[29]	75.42	75.31	75.30	75.33
CSAE	76.34	76.21	76.20	76.20
Proposed method	**77.51**	**77.38**	**77.40**	**77.35**

significant improvements in accuracy, precision, recall, and F1-score. For instance, our method achieved a classification accuracy of 92.96% on location A, whereas the DT, SVM, CNN, and CSAE method achieved 81.4%, 82.23%, 85.70%, and 90.69%, respectively. This indicates that our proposed method is more accurate and stable than other methods, with a maximum difference in accuracy of 11.56% and a minimum difference of 1.86%. Overall, the proposed algorithm demonstrates significant improvement compared to traditional ML algorithms, with high accuracy, precision, recall, and F1-score reaching 92.96%, 84.72%, 87.29%, and 82.62%, respectively, on the dataset of Location A.

In addition, accuracy is crucial for NLOS recognition, as NLOS signals can have a detrimental impact on positioning. Thus, we give an in-depth analysis of accuracy, and the results are illustrated in Fig. 13. This analysis visually demonstrates that our proposed method outperforms other methods in terms of accuracy across four datasets. The convolutional sparse autoencoder with LSTM method suggested for the recognition of BDS signals holds great promise in its ability to effectively utilize a substantial quantity of unlabeled data and a comparatively small amount of labeled data, while simultaneously accounting for temporal continuity through the unsupervised training process. In comparison to ML techniques that merely identify features at a particular point in time, the NN method proposed by us can extract brief fluctuations in the time-series features of the BDS signals. The method is valuable for applying DL in the field of BDS signal recognition.

Finally, we analyze the mean evaluation metrics of four location datasets, and the results of the experiments are presented in Fig. 14. It is observed that SVM, DT, and CNN exhibited high accuracy, but low performance in other metrics, indicating a lack of method robustness. The method in this research achieved notable accuracy rates of 85.17%, 81.27%, 82.08%, and 80.59% in terms of accuracy, F1-score, precision, and recall, respectively, outperforming all other experimental methods. These results validate the effectiveness of the proposed method. The CSAE method also performed well in terms of accuracy, F1-score, precision, and recall, but its metrics were lower than those of the proposed method, suggesting that the

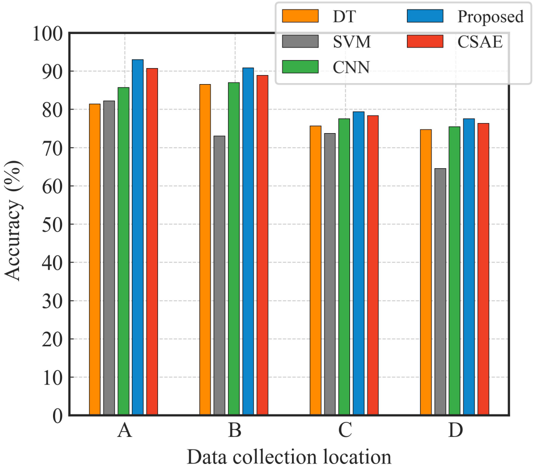


Fig. 13 Classification performance of different methods.

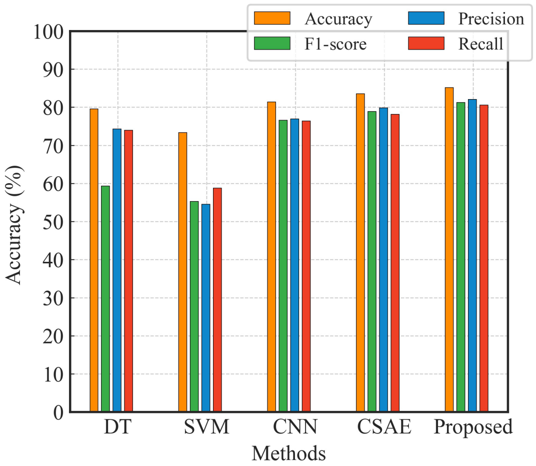


Fig. 14 Average classification performance of different methods.

addition of the LSTM module can further enhance the method's performance. It is worth noting that SVM approaches achieve such poor performance compared to other methods. The main reason may be that SVM can map low-dimensional data into high-dimensional space by kernel function, and then improve the classification accuracy. Therefore, SVM is more suitable for classification problems with low-dimensional small-sample data.

6.5 Sparsity analysis of the weights with the $l_{1/2}$ regularizer layer

To analyze the sparsity of the $l_{1/2}$ regularizer, we use the sparsity evaluation metric proposed by Hoyer's sparsity[39] to calculate the sparsity of the feature representation learned by the proposed method. And this method uses the difference between the l_1 and l_2 parameters of the vector to evaluate the sparsity of the vector. The sparsity can be expressed as

$$\text{Hoyer's sparsity}(x) = \frac{\sqrt{n} - \left(\sum_{i=1}^{n} |x_i| \,/\, \sqrt{\sum_{i=1}^{n} x_i^2}\right)}{\sqrt{n} - 1} \tag{25}$$

where x is the sample vector and n is the dimensionality of x. The value of Hoyer's sparsity is in the range of $[0,1]$, and the more larger the Hoyer's sparsity, the greater the difference between the elements in the vector, indicating that x is more sparse. In Fig. 15, the correlation between the quantity of iterations and the scarcity of the proposed method is illustrated. The results of the experiment demonstrate that the $l_{1/2}$ regularizer progressively approaches a consistent level of scarcity as the number of iterations is raised. While preserving a high standard of classification, sparsity is ensured. This indicates that the algorithm is capable of acquiring more effective features of BDS signals.

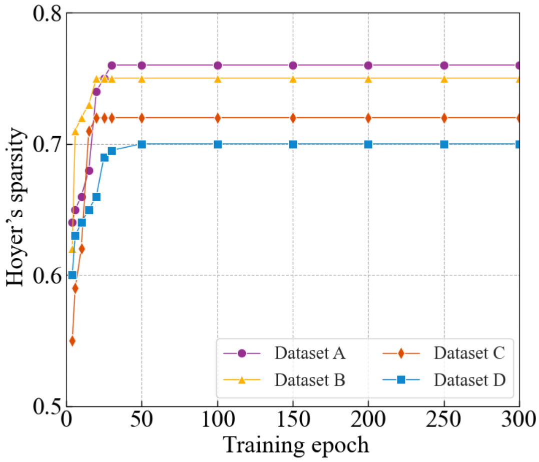


Fig. 15 Sparsity performance of $l_{1/2}$ regularizer.

6.6 Discussions

In our experiments, we investigate the performance of our proposed convolutional sparse autoencoder with the LSTM method in real environments with a variety of sky occlusion scenarios. In contrast to ML methods such as SVM and DT, we propose a convolution-based DL method. In addition, unlike CNN and convolutional sparse autocoder methods, our method incorporates LSTM into the method to extract important timing information. We also propose adding an $l_{1/2}$ regularizer to the objective function to induce strong sparsity among the parameters. Specific performance comparisons of our method against existing methods are listed as follows.

(1) From the detailed comparisons, as shown in Tables 2–5, it is clear that the proposed method is superior to ML methods such as SVM and DT for all metrics. Our proposed DL method improved the accuracy, precision, recall, and F1-scores by 11.56%, 17.16%, 19.38%, and 15.38%, respectively, compared with the DT method. The results suggest that the proposed method extracts rich features in the training process and can further improve performance across a variety of sky occlusion environments.

(2) Compared with existing DL methods such as the CNN and CSAE methods, as shown in Tables 2–5, the proposed method improved accuracy by 7.26% and 2.27%, respectively. The proposed method obtained better recognition accuracy in four environments (e.g., 92.96% for Dataset A, as shown in Fig. 13).

Furthermore, by checking the average values for accuracy, F1-score, precision, and recall, we note that all metrics for the method proposed in this paper exceed 80%, which is not achieved by any of the other methods, as shown in Fig. 14.

(3) In Fig. 11, the performances of various regularizers when using the corresponding optimal parameters are summarized. Here, the $l_{1/2}$ regularizer achieves the best classification performance across all datasets, which indicates that the $l_{1/2}$ regularizer is robust and can improve sparsity. A comparison of Hoyer's sparsity is shown in Fig. 15, where the $l_{1/2}$ regularizer pruned unnecessary weights from the NN (e.g., around 72% of the total across the four datasets). Furthermore, the $l_{1/2}$ regularizer could converge faster with fewer training epochs.

7 Conclusion

In this paper, we have presented an efficient convolutional sparse autoencoder with an LSTM training framework to address the NLOS signal classification problem. Specifically, to prevent high-dimensional data redundancy in DL methods, we implemented dimension reduction and feature extraction through a convolutional sparse autoencoder module structure. To capture the temporal correlations in long-duration time-series signals, we proposed an LSTM module that uses an input gate, a forget gate, and an output gate to obtain information about the evolution of the BDS signal over time. Then, to reduce unnecessary weights, we applied a regularizer to the weights of the hidden layers, which offered results beyond those for the LOG and MCP regularizers. Finally, the trained and simplified DL could provide accurate classification of BDS NLOS signals in real urban canyon environments.

Experiments with classification and pruning were conducted on four datasets with different sky occlusion scenarios. Compared with ML methods such as DT and SVM, our DL method, with its convolutional sparse autoencoder and LSTM module, demonstrated superior performance. Moreover, compared with existing DL-based methods such as CNNs and SCAEs, the proposed method achieved 7.26% and 2.27% improvement in accuracy, respectively. Compared with the LOG and MCP regularizers, the proposed $l_{1/2}$ regularizer used for NN pruning enabled our proposed method to induce strong sparsity in the parameters. The results demonstrate that, across all environments, through seeking deep sparse representations during propagation, the proposed method can improve classification performance by 1%–2%. Furthermore, the proposed $l_{1/2}$ regularizer can largely remove unnecessary weights while retaining the classification performance. In particular, it can reduce over 20% of the parameters while maintaining a significantly high classification accuracy for all urban canyon environments. The main reason may be that the method we proposed method that learns a compressed representation of the input data for downscaling and unsupervised feature extraction of long-duration time-series signals containing multipath. The cited methods also uses an unsupervised approach, but the cited methods uses a machine learning approach, and the recognition accuracy is significantly lower than the deep learning unsupervised feature extraction method used in this paper.

However, there are two main limitations of our algorithm: (1) The proposed algorithm only considers urban canyon; (2) The classification generalisation performance of the proposed algorithm needs to be improved. In order to address the above problem, in the future work, we will continue our studies of multipath signal-recognition-based unsupervised DL methods in more diverse and complex urban environments such as urban forests, overpasses, and viaducts. We note that our method loses information about the satellite environment, so we are considering learning the representation of indirect environmental information using a transformer. We are also working on inducing sparsity in DL algorithms that involve a CNN architecture, which should be more efficient for more complex DL environments.

Acknowledgment

This research was supported in part by the National Natural Science Foundations of China (Nos. 62273106, 62203122, 62320106008, 62373114, 62203123, and 62073086), in part by Guangdong Basic and Applied Basic Research Foundation (Nos. 2023A1515011480 and 2023A1515011159), and in part by China Postdoctoral Science Foundation funded project (No. 2022M720840).

References

[1] C. Ma, S. Pan, W. Gao, F. Ye, L. Liu, and H. Wang, Improving GNSS/INS tightly coupled positioning by using BDS-3 four-frequency observations in urban

environments, *Remote Sens.*, vol. 14, no. 3, p. 615, 2022.

[2] F. Li, R. Tu, J. Hong, S. Zhang, M. Liu, and X. Lu, Performance analysis of BDS–5G combined precise point positioning, *Remote Sens.*, vol. 14, no. 13, p. 3006, 2022.

[3] Y. Xu, Y. Yang, and J. Li, Performance evaluation of BDS-3 PPP-B2b precise point positioning service, *GPS Solut.*, vol. 25, no. 4, p. 142, 2021.

[4] H. Zhao, Z. Li, C. Chen, L. Wang, K. Xie, and S. Xie, Fusing vehicle trajectories and GNSS measurements to improve GNSS positioning correction based on actor-critic learning, in *Proc. 2023 Int. Technical Meeting of The Institute of Navigation*, Long Beach, CA, USA, 2023, pp. 82–94.

[5] W. Qiu, Q. Zeng, R. Xu, J. Liu, J. Shi, and Q. Meng, A multipath mitigation algorithm for GNSS signals based on the steepest descent approach, *Satell. Navig.*, vol. 3, no. 1, p. 14, 2022.

[6] P. R. R. Strode and P. D. Groves, GNSS multipath detection using three-frequency signal-to-noise measurements, *GPS Solut.*, vol. 20, no. 3, pp. 399–412, 2016.

[7] Groves, P. D, Jiang, Z, Rudi, M, and Strode, P, A portfolio approach to NLOS and multipath mitigation in dense urban areas, in *Proc. the 26th Int. Technical Meeting of the Satellite Division of The Institute of Navigation*, Nashville, TN, USA, 2013, pp. 3231–3247.

[8] L. T. Hsu, GNSS multipath detection using a machine learning approach, in *Proc. IEEE 20th Int. Conf. Intelligent Transportation Systems* (*ITSC*), Yokohama, Japan, 2017, pp. 1–6.

[9] Z. Li, K. Zeng, L. Wang, X. Kan, R. Yuan, and S. Xie, NLOS/LOS classification by constructing indirect environment interaction from GNSS receiver measurements using a transformer-based deep learning model, in *Proc. 2023 Int. Technical Meeting of The Institute of Navigation*, Long Beach, CA, USA, 2023, pp. 859–870.

[10] I. Smolyakov, M. Rezaee, and R. B. Langley, Resilient multipath prediction and detection architecture for low-cost navigation in challenging urban areas, *NAVIGATION*, vol. 67, no. 2, pp. 397–409, 2020.

[11] Z. Lyu and Y. Gao, An SVM based weight scheme for improving kinematic GNSS positioning accuracy with low-cost GNSS receiver in urban environments, *Sensors*, vol. 20, no. 24, p. 7265, 2020.

[12] 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, 2021.

[13] J. Chen, H. Shi, Z. Fang, C. Yuan and Y. Xu, Performance analysis of the GNSS instantaneous ambiguity resolution method using three collinear antennas, *IEEE Sens. J.*, pp. 11936–11945, 2023.

[14] G. Chang, C. Chen, Y. Yang, and T. Xu, Tikhonov regularization based modeling and sidereal filtering mitigation of GNSS multipath errors, *Remote. Sens.*, vol. 10, no. 11, p. 1801, 2018.

[15] R. Lu, W. Chen, D. Dong, Z. Wang, C. Zhang, Y. Peng, and C. Yu, Multipath mitigation in GNSS precise point positioning based on trend-surface analysis and multipath hemispherical map, *GPS Solut.*, vol. 25, no. 3, p. 119, 2021.

[16] A. El-Mowafy, B. Xu, and L. T. Hsu, Integrity monitoring using multi-GNSS pseudorange observations in the urban environment combining ARAIM and 3D city models, *J. Spatial Sci.*, vol. 67, no. 1, pp. 91–110, 2022.

[17] R. Sun, G. Wang, W. Zhang, L. T. Hsu, and W. Y. Ochieng, A gradient boosting decision tree based GPS signal reception classification algorithm, *Appl. Soft Comput.*, vol. 86, p. 105942, 2020.

[18] Y. Qin, Z. Li, S. Xie, R. Yuan, and J. Xie, BDS multipath signal classification using support vector machine, in *Proc. 4th Int. Conf. Industrial Artificial Intelligence* (*IAI*), Shenyang, China, 2022, pp. 1–6.

[19] G. Zhang, B. Xu, and L. T. Hsu, GNSS shadow matching based on intelligent LOS/NLOS classifier, in *Proc. 16th IAIN World Congr.*, Chiba, Japan, pp. 1–7, 2018.

[20] T. Suzuki and Y. Amano, NLOS multipath classification of GNSS signal correlation output using machine learning, *Sensors*, vol. 21, no. 7, p. 2503, 2021.

[21] R. Zawislak, M. Greiff, K. J. Kim, K. Berntorp, S. Di Cairano, M. Konishi, K. Parsons, P. V. Orlik, and Y. Sato, GNSS multipath detection aided by unsupervised domain adaptation, in *Proc. the 35th Int. Technical Meeting of the Satellite Division of The Institute of Navigation*, Denver, CO, USA, pp. 2127–2137, 2022.

[22] H. Wang, S. Pan, W. Gao, Y. Xia, and C. Ma, Multipath/NLOS detection based on K-means clustering for GNSS/INS tightly coupled system in urban areas, *Micromachines*, vol. 13, no. 7, p. 1128, 2022.

[23] H. Xu, L. T. Hsu, D. Lu, and B. Cai, Sky visibility estimation based on GNSS satellite visibility: An approach of GNSS-based context awareness, *GPS Solut.*, vol. 24, no. 2, p. 59, 2020.

[24] H. F. Ng, G. Zhang, and L. T. Hsu, Robust GNSS shadow matching for smartphones in urban canyons, *IEEE Sens. J.*, vol. 21, no. 16, pp. 18307–18317, 2021.

[25] S. Zhang, S. Lo, Y. H. Chen, T. Walter, and P. Enge, GNSS multipath detection in urban environment using 3D building model, in *Proc. IEEE/ION Position, Location and Navigation Symp.* (*PLANS*), Monterey, CA, USA, 2018, pp. 1053–1058.

[26] S. Miura, L. T. Hsu, F. Chen, and S. Kamijo, GPS error correction with pseudorange evaluation using three-dimensional maps, *IEEE Trans. Intell. Transport. Syst.*, vol. 16, no. 6, pp. 3104–3115, 2015.

[27] G. Zhang, P. Xu, H. Xu, and L. T. Hsu, Prediction on the urban GNSS measurement uncertainty based on deep learning networks with long short-term memory, *IEEE Sens. J.*, vol. 21, no. 18, pp. 20563–20577, 2021.

[28] H. Su, B. Wu, and X. Mao, Non-line-of-sight multipath detection method for BDS/GPS fusion system based on deep learning, *J. Shanghai Jiaotong Univ. Sci.*, vol. 27, no. 6, pp. 844–854, 2022.

[29] Y. Quan, L. Lau, G. W. Roberts, X. Meng, and C. Zhang, Convolutional neural network based multipath detection

method for static and kinematic GPS high precision positioning, *Remote. Sens.*, vol. 10, no. 12, p. 2052, 2018.

[30] A. Blais, N. Couellan, and E. Munin, A novel image representation of GNSS correlation for deep learning multipath detection, *Array*, vol. 14, p. 100167, 2022.

[31] A. V. Kanhere, S. Gupta, A. Shetty, and G. Gao, Improving gnss positioning using neural-network-based corrections, *NAVIGATION-US*, vol. 69, no. 4, p. 548, 2022.

[32] Z. He, H. Shao, Z. Ding, H. Jiang, and J. Cheng, Modified deep autoencoder driven by multisource parameters for fault transfer prognosis of aeroengine, *IEEE Trans. Ind. Electron.*, vol. 69, no. 1, pp. 845–855, 2022.

[33] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, LSTM: A search space odyssey, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, 2017.

[34] Z. Xu, X. Chang, F. Xu, and H. Zhang, $L_{1/2}$ regularization: A thresholding representation theory and a fast solver, *IEEE Trans. Neural Netw. Learning Syst.*, vol. 23, no. 7, pp. 1013–1027, 2012.

[35] Z. Li, S. Ding, W. Chen, Z. Yang, and S. Xie, Proximal alternating minimization for analysis dictionary learning and convergence analysis, *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 6, pp. 439–449, 2018.

[36] H. Zhao, J. Wu, Z. Li, W. Chen, and Z. Zheng, Double sparse deep reinforcement learning via multilayer sparse coding and nonconvex regularized pruning, *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 765–778, 2023.

[37] Z. Li, H. Zhao, Y. Guo, Z. Yang, and S. Xie, Accelerated *Log*-regularized convolutional transform learning and its convergence guarantee, *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10785–10799, 2022.

[38] Z. Li, Z. Yang, H. Zhao, and S. Xie, Direct-optimization-based DC dictionary learning with the MCP regularizer, *IEEE Trans. Neural Netw. Learning Syst.*, vol. 34, no. 7, pp. 3568–3579, 2023.

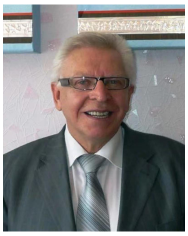
[39] P. O. Hoyer, Nonnegative matrix factorization with sparseness constraints, *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, 2004.

Yahang Qin received the MS degree from Guangxi University of Science and Technology, Liuzhou, China, in 2017. She is currently pursuing the PhD degree in control science and engineering with School of Automation, Guangdong University of Technology, and also with Guangdong-Hong Kong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangdong University of Technology. Her research interests include deep learning and BDS positioning in challenging environments.

Shengli Xie received the BS degree in mathematics from Jilin University, China in 1983, the MS degree in mathematics from Central China Normal University, China in 1995, and the PhD degree in control theory and applications from South China University of Technology, China in 1997. He is currently a full professor in School of Automation, Guangdong University of Technology, the head of the 111 Center for Intelligent Batch Manufacturing Based on IoT Technology (GDUT), and also with the Key Laboratory of Intelligent Information Processing and System Integration of IoT (GDUT), Ministry of Education. He has coauthored two books and over 150 research papers in refered journals and conference proceedings, and was awarded Highly Cited Researcher in 2020. His research interests include blind signal processing, machine learning, and Internet of Things. He was awarded the Second Prize of National Natural Science Award of China in 2009. He is an associate editor of *IEEE TSMC*.

Zhenni Li received the BSc degree from School of Physical Science and Electronics, Shanxi Datong University, in 2009, the MSc degree from School of Physics and Optoelectronic, Dalian University of Technology, in 2012, and received the PhD degree from School of Computer Science and Engineering, University of Aizu, Japan. She is currently an associate professor with School of Automation, Guangdong University of Technology, and also with Guangdong-Hong Kong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangdong University of Technology. Her research interests include deep learning, reinforcement learning, and high-precious position.

Bo Li received the MS degree in mathematics from Central China Normal University, Wuhan, China, in 1992, and the PhD degree in control theory and applications from South China University of Technology, Guangzhou, China, in 1997. He is currently the director of Laboratory for Intelligent Information Processing and a full professor with the School of Automation, Guangdong University of Technology, and also with Guangdong Key Laboratory of IoT Information Technology (GDUT). He has authored or coauthored two monographs, a dozen patents, and over 80 papers in journals and conference proceedings. His current research interests include automatic controls, signal processing, blind signal processing, and image processing.



Victor Kuzin is a leading expert in the field of physical and technical control of mining products, physical processes, and quality management. He has published over 300 articles. He is the author of 18 monographs and booklets, 3 of which were published abroad. He is a full member of Academician of Russian Engineering Academy, Academician of Russian Academy of Natural Sciences, International Academy of Mineral Resources Sciences, Academician of Academy of Technological Sciences of the Russian Federation, and Academician of International Academy of Mineral Resources. He is an honorary doctor and professor at several universities in Europe and China. In 1979, he became a recipient of the Lenin Komsomol Prize. He was awarded the Russian Government Prize for Science and Technology in 2011 and the Russian Prize for Science and Technology in 2013.

Ming Liu received the PhD degree in robotics from ETH Zurich. He is currently the director of Intelligent Autonomous Driving Center of HKUST. He published over 300 articles. His research interests include dynamic environment modeling, 3-D mapping, machine learning, and visual control. He received various IEEE awards, including the IEEE IROS Young Professional Award in 2018 and 15 best paper awards or finalist awards. He is particularly interested in the investigation of novel, real-time online approaches in solving autonomous driving, mobile robot mapping, and navigation. His team developed the first autonomous vehicle in Hong Kong in 2017. During COVID-19, his autonomous vehicles served several cities and were reported by more than 100 media, including IEEE Spectrum and Xinhua.