



OPEN An incremental adversarial training method enables timeliness and rapid new knowledge acquisition

Yuxin Ge, Yanhua Dong✉, Hongyu Sun✉, Yuetong Liu & Chengli Wang

Adversarial training is an effective defense method for deep models against adversarial attacks. However, current adversarial training methods require retraining the entire neural network, which consumes a significant amount of computational resources, thereby affecting the timeliness of deep models and further hindering the rapid learning process of new knowledge. In response to the above problems, this article proposes an incremental adversarial training method (IncAT) and applies it to the field of brain computer interfaces (BCI). Within this method, we first propose a deep model called Neural Hybrid Assembly Network (NHANet) and then train it. Then, based on the original samples and the trained deep model, calculate the Fisher information matrix to evaluate the importance of deep neural network parameters on the original samples. Finally, when calculating the loss of adversarial samples and real labels, an Elastic Weight Consolidation (EWC) loss is added to limit the variation of important weights and bias parameters in the Neural Hybrid Assembly Network (NHANet). The above incremental adversarial training method was applied to the publicly available epilepsy brain computer interface dataset at the University of Bonn. The experimental results showed that when facing three different attack algorithms, including fast gradient sign method (FGSM), projected gradient descent (PGD) and basic iterative method (BIM), the method proposed in this paper achieved robust accuracies of 95.33%, 94.67%, and 93.60%, respectively, without affecting the accuracy of clean samples, which is 5.06%, 4.67%, and 2.67% higher than traditional training methods respectively, thus fully verifying the generalization and effectiveness of the method.

In recent years, deep neural networks have achieved significant success in fields such as brain computer interfaces^{1,2}, object detection^{3–5}, texture recognition⁶, image classification^{7–11}, etc. However, Szegedy et al.¹² revealed the existence of adversarial samples in deep learning models, making them exceptionally vulnerable to adversarial attacks. Attackers only need to add small perturbations generated by specific algorithms to clean samples, and deep neural networks can output erroneous classification results with high confidence^{13,14}. For example, in the process of neural rehabilitation, if attackers add small perturbations to electroencephalogram (EEG) signals, deep neural networks may misunderstand the patient's intentions due to adversarial attacks, leading to treatment failure or adverse reactions. Therefore, the robustness and security issues of deep learning models have received widespread attention and research from both academia and industry.

To address the vulnerability of deep learning models to adversarial samples, researchers have developed various defense methods to enhance model robustness. Among them, adversarial training (AT) is considered one of the most effective. The core idea is to introduce carefully designed adversarial samples into the training set, so as to have stronger resistance to interference and disturbance. Madry et al.¹⁵ proposed an adversarial training method using projected gradient descent, which effectively improves the model's ability to resist adversarial attacks. However, its multi-step perturbation process requires high computational resources and time, which to some extent limits the practicality of this method. To address this problem, researchers have proposed an alternative method—fast adversarial training¹⁶, which only uses the one-step fast gradient sign method to generate training data. However, this rapid adversarial training method has a significant drawback, which is that it can easily lead to overtraining and overfitting of the model on the training data, resulting in poor performance when faced with new and unseen data. To alleviate this problem, Rice et al.¹⁷ proposed an early stopping version of projection gradient descent adversarial training. Unlike traditional adversarial training methods, this method introduces a stop criterion to avoid the degradation of model performance due to overtraining of adversarial samples. In addition, Zhang et al.¹⁸ proposed an adversarial training method called Trades, which aims to achieve an ideal balance between clean sample accuracy and robust accuracy by optimizing the loss function.

College of Mathematics and Computer, Jilin Normal University, Siping 136000, China. ✉email: computerdyp@jlnu.edu.cn; hongyu@jlnu.edu.cn

Although existing adversarial training methods improve model robustness and security to some extent, they suffer from notable limitations: (1) retraining the original and adversarial samples not only increases the complexity of model training, but also reduces the timeliness of the model, especially in application scenarios that require fast iteration and response. (2) When faced with large-scale datasets, these adversarial training methods significantly increase the demand for computing resources, which may become a bottleneck in resource constrained environments. To address the aforementioned problems, this paper proposes an incremental adversarial training method (IncAT). This method first uses clean samples and a pre trained Neural Hybrid Assembly Network (NHANet) to calculate the Fisher information matrix. The weights with higher Fisher information matrix values are considered more critical for clean samples. Therefore, in the learning process of adversarial samples, if these weights are significantly updated, they will receive greater punishment. Then, adversarial samples are generated during the training phase. Finally, by introducing a quadratic penalty term in the loss function, the significant changes in weights during the learning of adversarial sample features in deep learning models can be alleviated. This strategy not only enables deep learning models to learn features of adversarial samples, but also maintains memory of clean samples, thereby improving the robustness and generalization ability of the model. The main contributions of this paper on incremental adversarial training are as follows:

- (1) To address the issues of insufficient feature extraction and poor generalization ability of existing deep learning models for brain-computer interfaces in complex scenarios, this paper proposes a hybrid neural network, NHANet. This model integrates the advantages of multiple deep learning modules, aiming to more effectively process time-series data and capture long-term dependencies as well as complex spatial features. This innovation not only significantly improves the model's performance in complex environments but also provides new ideas and valuable practical experience for the application of deep learning in the field of brain-computer interfaces.
- (2) In response to the security risks of adversarial attacks faced by deep learning models in BCI application scenarios, this study conducts adversarial attacks on the trained NHANet model. The aim is to conduct a multi-dimensional performance evaluation to deeply analyze the impact of adversarial perturbations on the feature representation ability and classification decision stability of deep learning models in the BCI field, thereby revealing the importance and urgency of enhancing the robustness of deep learning models in BCI applications.
- (3) This paper introduces the incremental adversarial training method for the first time. This approach utilizes adversarial examples to continuously adjust the parameters of the baseline model, thereby enhancing the robustness and security of deep learning models and avoiding the problem that traditional adversarial training methods require retraining the entire network. In addition, to further verify the effectiveness of the proposed method, the robust accuracy is introduced as an evaluation index to reflect the ability of the deep learning model to resist adversarial attacks after adversarial training.
- (4) The proposed method was extensively tested on the publicly available epilepsy BCI dataset from the University of Bonn. The experiments demonstrated that the proposed method outperformed traditional adversarial training methods in terms of accuracy on clean samples and robustness accuracy.

Methods

The incremental adversarial training method proposed in this article is designed as shown in Fig. 1. Firstly, we train the NHANet model to help it better understand the underlying patterns in the data. Then, we carry out adversarial attacks on all the original data to generate adversarial samples. Next, the EWC loss term is introduced when calculating the adversarial sample loss function, and the total loss function is constructed based on this. Finally, utilizing the backpropagation mechanism, the model parameters are adjusted based on the total loss function to enhance the deep model's ability to resist adversarial attacks.

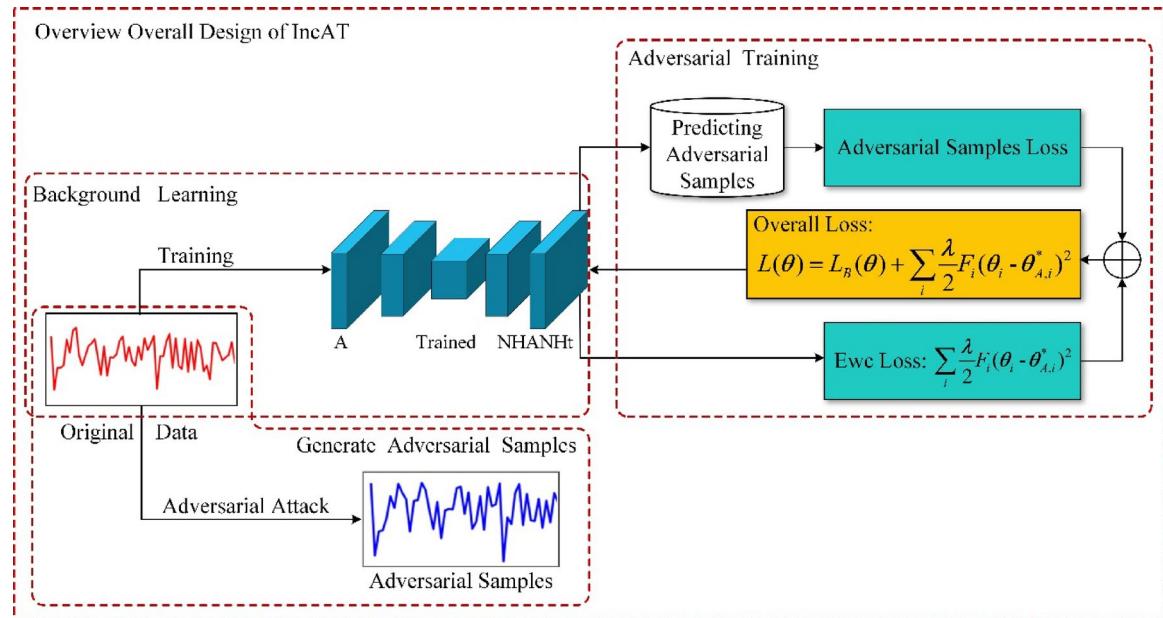
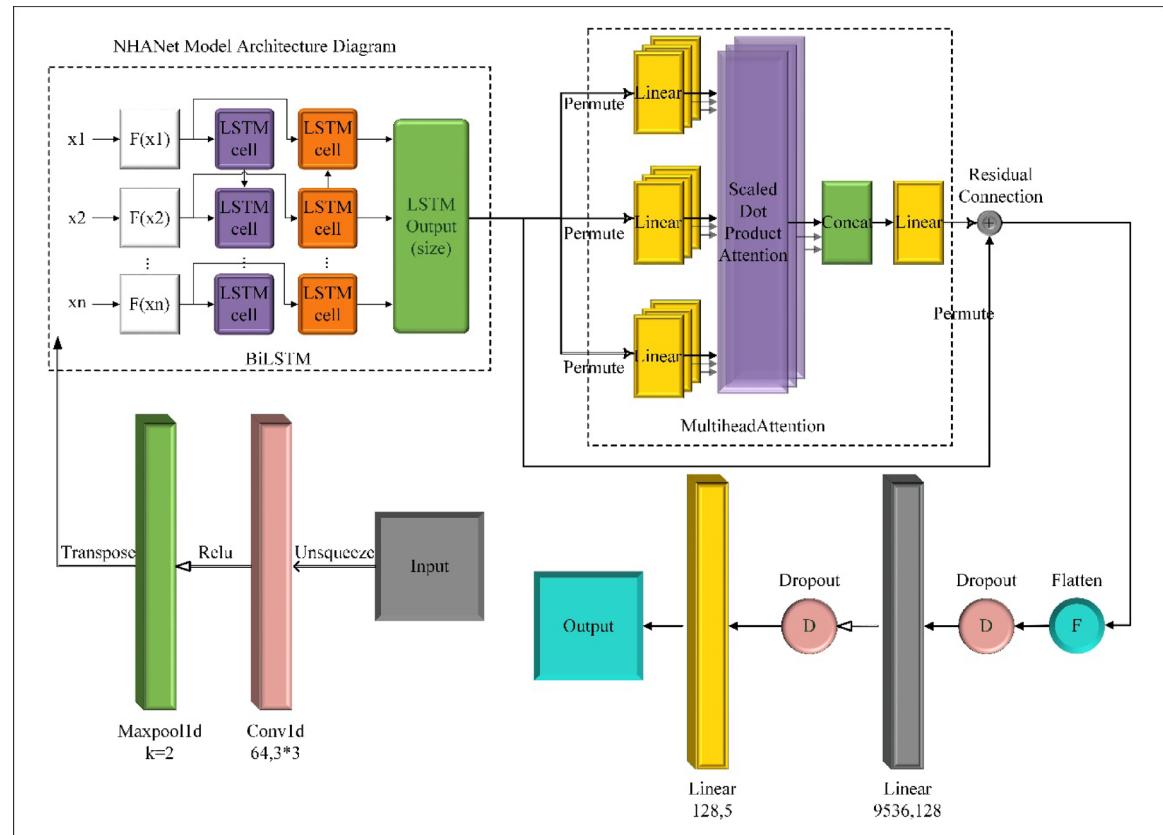
Neural hybrid assembly network architecture design

This article proposes a hybrid neural network model called NHANet, which integrates various cutting-edge deep learning techniques such as convolutional neural networks, bidirectional long short-term memory networks, multi head attention mechanisms, residual connections, and fully connected layers. The goal is to fully utilize the strengths of different neural networks in processing specific data, thereby enhancing the performance of deep models in complex EEG signal recognition tasks. The specific network framework is shown in Fig. 2.

Firstly, add a channel dimension to the preprocessed data to ensure it meets the input requirements of the one-dimensional deep convolution module. Then, in the one-dimensional deep convolution module, 64 convolution kernels of size 3 are used to convolve along the time axis of the original signal, and ReLU activation function is used to increase nonlinearity. Finally, a maximum pooling layer with a size of 2 is used for downsampling to reduce data redundancy. Although the deep convolution module can obtain local features of EEG signal data, it cannot capture long-term dependencies of time series data. Therefore, a bidirectional LSTM layer was introduced to compensate for this deficiency.

In the NHANet model, by introducing a bidirectional LSTM layer, its bidirectional recursive structure is utilized to conduct bidirectional feature extraction of time series data, thereby effectively capturing the long-term dependencies in the signal sequence and further improving the representation ability of the deep learning model for the dynamic features of electroencephalogram signals.

Specifically, there are 32 hidden units in each direction of the bidirectional LSTM layer, which work together to capture complex features in the input sequence. Although the BiLSTM layer can effectively process time series data, it still has limitations in capturing the global dependencies of the entire sequence. To further improve the

**Fig. 1.** Overview overall design of IncAT.**Fig. 2.** NHANet model architecture diagram.

performance of deep learning models, we added a multi head attention mechanism module after the BiLSTM layer.

By introducing this mechanism, deep learning models can focus on different parts of input data in parallel, significantly improving their ability to capture key information in sequences. Inside the multi head attention

module, 16 attention heads work in parallel, each with 64 embedding dimensions, capable of independently focusing on different aspects of information. $X = [X_1, X_2, \dots, X_n]$ is the matrix output by the BiLSTM module, which is mapped to three vector spaces Q (query), K (key), and V (value) through linear transformation. The specific formula is shown as follows:

$$Q = XW^Q \quad (1)$$

$$K = XW^K \quad (2)$$

$$V = XW^V \quad (3)$$

where W^Q, W^K, W^V is the weight matrix and the three are trainable parameter matrices. In the multi head attention mechanism, each head independently calculates attention weights and generates multiple attention outputs in parallel. The formula is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

and finally the outputs of all attention are concatenated using the concat function.

However, as the number of neural network layers increases, training may encounter problems such as vanishing or exploding gradients. To address this challenge, we incorporated a residual connection mechanism into the deep learning model, directly connecting the input and output of the multi-head attention mechanism. This cross-layer connection design helps to enhance the flow of gradients within the network, thereby improving the training stability and performance of the model.

Finally, first use a fully connected layer to map the input vector to a hidden space with a dimension of 128, and increase nonlinearity through the ReLU activation function. Then, use another fully connected layer as the output layer to map the representation in this hidden space to the number of corresponding final output categories, thereby completing the classification task.

Adversarial attack based on neural hybrid assembly network

This section focuses on the impact of adversarial attacks on the performance and robustness of NHANet models. By implementing adversarial attacks on deep neural networks, we can gain a deeper understanding of the vulnerability of deep learning models and better design security defense mechanisms to resist the negative impact of adversarial attacks on deep models.

The research on the adversarial attack is divided into the following three parts: first, based on the trained NHANet model with same weight parameters, three algorithms such as fast gradient sign method (FGSM)¹⁹, basic iterative method (BIM)²⁰, and projected gradient descent (PGD)¹⁵ are respectively used to generate the adversarial sample by conducting adversarial attack on all the original sample. Then, the trained deep neural network is used to predict the generated adversarial samples, and the impact on the classification performance of the model is observed by adjusting the epsilons. In addition, by visualizing the raw data and adding perturbed data, we can observe whether the generated adversarial samples have concealment. Finally, in order to further reveal the impact of adversarial attacks on deep models, we also conducted adversarial attacks on common deep learning models. That is, adversarial attacks not only affect the classification effect of NHANet model, but also affect the performance of other deep learning models.

Neural hybrid assembly network incremental adversarial training

In response to the problems of lack of timeliness and consumption of computational resources of traditional adversarial training methods, this paper employs an incremental learning algorithm to continuously learn the generated adversarial samples. Because existing research shows that in the case of limited storage space and computing resources, adopting incremental learning method can not only effectively cope with the challenge of new tasks or data, but also maintain the performance of old tasks.

The framework of the method is shown in Fig. 3. Among them, θ_i ($i \in 1, 2, 3, \dots, N$) is the neural network parameters, N is the number of neural network parameters, F is the Fisher information matrix, λ is a hyper parameter to measure the importance of the original sample relative to the adversarial sample, $L_B(\theta)$ is the loss function of the adversarial sample dataset, and $\theta_{A,i}^*$ is the original model parameters.

First, the NHANet deep learning model is trained to enhance its predictive performance for EEG signals. After the training is completed, the model weights are saved. Then, based on the original dataset samples and the parameters of the original NHANet deep learning model, the first derivatives of the NHANet deep learning model output about the neural network parameters are calculated, and the Fisher information matrix is constructed. The importance of the neural network parameters on the original samples can be reflected by the Fisher information matrix. Among them, the larger FIM value represents the higher importance of the parameters in the original dataset. Finally, during the adversarial training process, all original data is attacked to generate adversarial samples. When calculating the loss between the predicted results of adversarial samples and the true labels, an additional EWC loss term is added to limit the changes in important weights and bias parameters in the NHANet hybrid model. The specific total loss value is shown as follows:

$$L(\theta) = L_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta_{A,i}^*)^2 \quad (5)$$

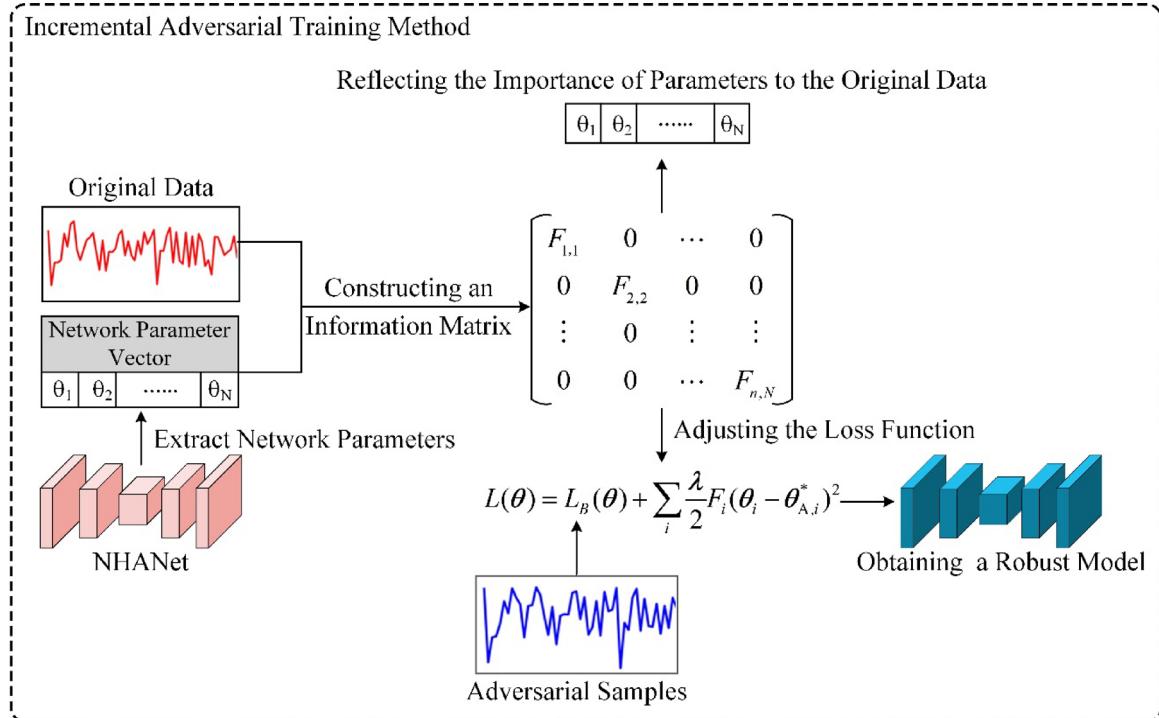


Fig. 3. Flow chart of incremental adversarial training method.

For parameters that are more important in the original dataset, a greater penalty value will be assigned during the update process to ensure that they are less prone to significant changes. Therefore, when using adversarial training methods based on incremental learning to improve the robustness of deep learning models, it not only firmly grasps existing knowledge but also flexibly responds to new challenges, maintaining its robustness and adaptability in a dynamic environment.

Experiments

To verify the effectiveness of the proposed method and the classification performance of the deep learning model, we conducted a systematic experiment on the epilepsy dataset. Firstly, we constructed the neural hybrid assembly network NHANet, which achieved efficient feature extraction and high-precision classification in complex scenarios through a multi-module collaborative mechanism. Secondly, three typical adversarial attack algorithms, FGSM, BIM, and PGD, were used to conduct adversarial attacks on the trained NHANet, aiming to illustrate the impact of adversarial attacks on deep learning models. Finally, we introduced the incremental adversarial training method to enhance the model's defense performance and compared it with existing adversarial training methods to verify the effectiveness and generalization of this method.

Experimental design

Dataset

This article selects the epilepsy dataset publicly available from the University of Bonn²¹ to verify the effectiveness of incremental adversarial training method. It consists of five categories, each containing 100 channel sequences with a duration of 23.6 seconds and 4097 signal sampling points. To further improve model performance and accelerate convergence, we performed a series of preprocessing operations on the dataset. To address the issue where high data dimensionality might increase computational burden, we implemented dimensionality reduction on the data to enhance efficiency while retaining key information. Considering that the limited size of the original dataset could easily lead to overfitting due to insufficient training samples, we expanded the dataset by synthesizing new samples. Additionally, we standardized and normalized the data, and converted non-numerical labels into numerical encodings to meet the requirements of deep learning models.

Experiment details

The experiment is implemented based on the PyTorch deep learning framework, and the dataset is split into a training set and a test set in a 7:3 ratio. The optimizer uses adaptive momentum estimation, and the Dropout value is set to 0.5, the batch-size is set to 32, and the learning rate is set to 0.0003, and the number of heads in the multi-head attention mechanism is set to 16. When conducting incremental adversarial training, λ is set to 1e-5.

Evaluation metrics

Model evaluation metrics

In order to illustrate the effectiveness and stability of the deep learning model, accuracy, precision, recall, and F1-score are introduced as evaluation metrics.

Attack evaluation metrics

This study uses four indicators, namely adversarial accuracy, attack success rate, average L_1 distance, and average L_2 distance, to evaluate the impact of adversarial attacks on deep learning models.

Adversarial accuracy refers to the accuracy of a classification model on adversarial samples. It is measured by calculating the proportion of adversarial samples where the predicted labels match the true labels. The higher the adversarial accuracy, the stronger the ability of the deep learning model to resist adversarial attacks. Conversely, the lower the adversarial accuracy, the weaker the ability of the model to resist adversarial attacks.

The attack success rate is used to measure the attack effect of adversarial samples on the target model. This metric reflects the effectiveness of adversarial attacks. The closer the ASR value is to 1, the stronger the attack capability is²². The specific formula is shown as follows:

$$I_{ASR} = \frac{\sum_{i=1}^N (F(x_i) = real_i) \wedge (F(x_i + \delta_i) \neq real_i)}{\sum_{i=1}^N (F(x_i) = real_i)} \quad (6)$$

where, $F(*)$ is the sample label predicted by the depth model, and $real_i$ is the true label of the i th sample.

The average L_2 distance is used to measure the degree of difference between adversarial samples and raw samples. The smaller the average L_2 distance, the smaller the perturbation amplitude added to the original sample and the closer it is to the original sample.

The average L_1 distance refers to first calculating the sum of the absolute differences between each generated adversarial sample and the elements of the original sample, then adding the L_1 distances of all samples, and finally dividing by the number of samples. The larger the average L_1 distance, the greater the difference between the generated adversarial samples and the original samples.

Defense evaluation metrics

In order to evaluate the performance of the adversarial training method proposed in this paper, the accuracy, precision, recall and F1-score are used as evaluation metrics in the original data set. On the generated adversarial sample data set, the robust accuracy is used as the evaluation metric. Among them, the robust accuracy refers to the accuracy of the deep learning model in the face of adversarial samples, which reflects the ability of the deep learning model to resist adversarial attacks after adversarial training. The specific formula is shown as follows:

$$Robust\text{-}accuracy = \frac{N_{corr}}{N_{total}} \quad (7)$$

where N_{corr} is the number of correctly classified adversarial samples, and N_{total} is the total number of adversarial samples.

Network model analysis

The division of the dataset

In the development of deep learning models, the proportion of dataset division is a crucial step, and its rationality directly affects the training effect of the model, parameter optimization, and generalization ability. Given the small size of the dataset, this experiment only divided it into the training set and the test set. In this experiment, to explore the impact of different division ratios on the model performance, we set two typical schemes with training set-test set ratios of 8:2 and 7:3. The specific results are presented in Tables 1 and 2.

Through the data analysis of Tables 1 and 2, it can be seen that when the dataset partition ratio is 8:2, regardless of the batch size, the performance of the model is superior to that of the model with a dataset partition ratio of 7:3. However, due to the limited total number of samples in the dataset, using a 7:3 split ratio allows for the creation of a relatively large test set. This enables us to more accurately evaluate the generalization ability of the model, especially when dealing with limited sample data. A larger test set can provide a more stable performance assessment and reduce the evaluation errors caused by insufficient sample quantities. Therefore, considering the evaluation accuracy and actual application requirements, this experiment finally selects 7:3 as the dataset partition ratio.

Batch-size	Accuracy	Precision	Recall	F1-score
8	0.9980	0.9976	0.9983	0.9979
16	0.9940	0.9945	0.9933	0.9938
32	0.9860	0.9855	0.9862	0.9858
64	0.9860	0.9867	0.9860	0.9863
128	0.9740	0.9733	0.9748	0.9738

Table 1. Model performance for different Batch-sizes when the dataset is divided in a ratio of 8:2.

	Accuracy	Precision	Recall	F1-score
8	0.9920	0.9916	0.9921	0.9918
16	0.9893	0.9889	0.9898	0.9892
32	0.9827	0.9824	0.9831	0.9825
64	0.9747	0.9739	0.9756	0.9745
128	0.9587	0.9576	0.9597	0.9582

Table 2. Model performance for different Batch-sizes when the dataset is divided in a ratio of 7:3.

Batch-size	Model execution time	Best accuracy	Epoch
8	490.7729 s	0.9947	88
16	261.2940 s	0.9907	94
32	144.8274 s	0.9853	99
64	90.3085 s	0.9760	93
128	63.6864 s	0.9640	98

Table 3. Optimal accuracy parameters for different batches.

Batch-size	8	16	32	64	128
Accuracy	0.9933	0.9880	0.9800	0.9707	0.9560
Precision	0.9931	0.9876	0.9794	0.9702	0.9549
Recall	0.9935	0.9880	0.9805	0.9707	0.9568
F1-score	0.9933	0.9876	0.9798	0.9704	0.9553
Execution-time	481.4006 s	257.2427 s	145.0358 s	91.6405 s	65.3319 s

Table 4. NHANet model performance metrics at a learning rate of 0.0003.

Batch-size	8	16	32	64	128
Accuracy	0.9773	0.9693	0.9053	0.8160	0.7520
Precision	0.9772	0.9694	0.9032	0.8111	0.7494
Recall	0.9772	0.9703	0.9035	0.8134	0.7484
F1-score	0.9771	0.9694	0.9026	0.8106	0.7444
Execution-time	487.1841s	256.0790 s	143.1442 s	90.4628 s	64.0669 s

Table 5. NHANet model performance metrics at a learning rate of 0.0001.

Model performance analysis

The main purpose of this experiment is to study the effect of learning rate and batch-size on the performance and execution time of deep learning model. By testing the performance of the model under different batch-sizes and learning rates, the aim is to find an optimal batch-size and learning rate setting, so that the model can achieve high accuracy in a relatively short time. In the experiment, the batch-sizes were set to 8, 16, 32, 64, and 128 respectively, and the best accuracy, best epoch, and running time of the NHANet deep learning model under different batch-sizes were tested. The specific experimental results are shown in Table 3.

According to the experimental data in Table 3, When the batch-size is set to 8, the model achieves the highest accuracy of 0.9947 in the 88th epoch. However, this process is relatively time consuming, and the utilization rate of computing resources is low. In contrast, when batch-size is increased to 128, the running time of the model is significantly shortened, but the accuracy is relatively low. When batch-size is set to 32, when the model runs to the 99th round, it not only achieves a high accuracy of 0.9853, but also saves the consumption of computer resources and achieves a good balance between performance and efficiency. Therefore, we can see that the value of batch-size has an important impact on the running time and performance of the model. In practical application, we should strive to achieve high accuracy in a relatively short time to maximize the efficiency and practicality of the model.

By adjusting the learning rate, several groups of experiments were conducted to explore the impact of different learning rates on model performance and running time. When epoch is set to 100 and learning rate is set to 0.0003 and 0.0001 respectively, the performance of NHANet model is analyzed. The specific results are shown in Tables 4 and 5.

Num-heads	Accuracy	Precision	Recall	F1-score
16	0.9813	0.9809	0.9817	0.9812
8	0.9787	0.9783	0.9789	0.9786
4	0.9733	0.9728	0.9742	0.9732
2	0.9773	0.9769	0.9778	0.9773

Table 6. Performance of models with different numbers of heads.

	Accuracy	Precision	Recall	F1-score
DNN	0.9627	0.9628	0.9645	0.9632
BiLSTM	0.9640	0.9645	0.9654	0.9645
BiLSTM-MultiheadAttention	0.9587	0.9588	0.9613	0.9597
CNN-LSTM	0.9560	0.9553	0.9562	0.9557
CNN-LSTM-ATT	0.9307	0.9289	0.9307	0.9294
NHANet	0.9840	0.9842	0.9837	0.9839

Table 7. Performance metrics of different models.

According to the experimental data in Tables 4 and 5, when the batch-size is set to the same, different learning rates have a relatively small impact on the execution time of the deep learning model NHANet. However, it is worth noting that when the learning rate is adjusted from 0.0001 to 0.0003, the model performance is significantly improved. This may be because a lower learning rate slows down the update speed of model parameters, resulting in the need for more iterations for the model to achieve a better classification effect. Based on this, this study sets the learning rate to 0.0003, which enables the deep learning model NHANet to learn the patterns in the data more efficiently and thoroughly on the premise that the execution time does not increase significantly.

The "number of heads" in the multi-head attention mechanism serves as a core hyperparameter, and its value directly affects the feature representation ability and generalization performance of the deep learning model. To systematically explore the influence of the number of heads on the model's performance, in this experiment, the number of heads in the multi-head attention mechanism was set to 16, 8, 4, and 2. Comparative experiments were conducted based on the same training dataset and evaluation metrics. The specific experimental results are shown in Table 6.

As shown in the table above, the number of heads in the multi-head attention mechanism has a significant impact on the model performance. When the number of heads is set to 16, the model achieves the best results in terms of accuracy, precision, recall rate, and F1-score. This indicates that a larger number of heads enables the model to concurrently extract semantic information from different subspaces, capturing feature correlations more comprehensively and multi-dimensionally. Moreover, as the number of heads decreases, each indicator shows a stepwise decline trend, indicating that when the number of heads is insufficient, the model's perspective is limited, making it difficult to fully model complex feature relationships. It is worth noting that when the number of heads drops to 2, although the indicators slightly recover compared to when the number is 4, their overall performance is still far inferior to that of the 16-head model. This reflects that excessively reducing the number of heads will severely restrict the model's ability to capture rich feature patterns.

Comparative experiments

To verify the efficiency and accuracy of the NHANet deep learning model proposed in this paper, a comparative experiment was conducted by comparing it with several typical deep learning models. The experiment used the same dataset to evaluate the performance and generalization ability of models such as DNN, BiLSTM, BiLSTM-MultiheadAttention, CNN_LSTM, CNN_LSTM_ATT, and NHANet. The performance indicators of different deep learning models are shown in Table 7. Additionally, to more intuitively display the relationship between the model's prediction results and the true labels, we also constructed a confusion matrix. Figure 4 shows the confusion matrices of different deep learning models. Through these experimental results, we can clearly compare the performance of each model in the classification task and thereby verify the superiority of the NHANet model.

It can be seen from the experimental results in Table 7 that the NHANet deep learning model proposed in this paper is significantly superior to the comparison models in all evaluation indicators. It is worth noting that the traditional BiLSTM model also demonstrates a relatively high performance level. This is mainly attributed to its synchronous processing of sequence data through forward and backward LSTM units, effectively capturing the context information of the past and future in the signal. In contrast, the CNN_LSTM_self-attention mechanism model has poor classification performance on this dataset. Although the self-attention mechanism can enhance the model's ability to focus on key features, due to the limitations of the current task characteristics and data distribution characteristics, this model fails to give full play to its advantages. Instead, it leads to performance degradation of the model in feature integration and classification decision-making. In conclusion, the NHANet model proposed in this study has the most outstanding overall performance. This discovery provides new solution strategies and innovative ideas for the research in the field of brain-computer interfaces.

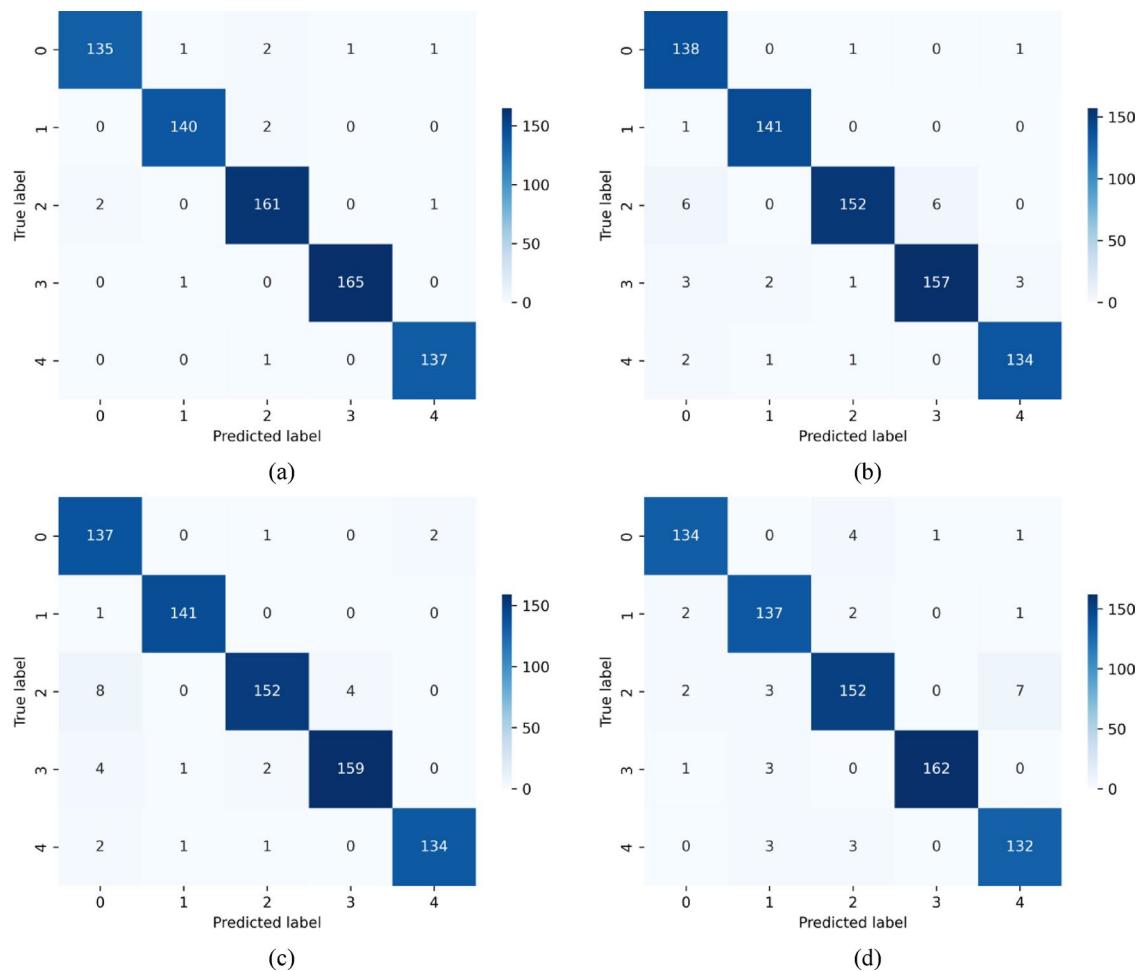


Fig. 4. Confusion matrix results for different models. (a) Confusion matrix for the NHANet model. (b) Confusion matrix for the DNN model. (c) Confusion matrix for the BiLSTM model. (d) Confusion matrix for the CNN-LSTM model. (e) Confusion matrix for the CNN-LSTM-ATT model. (f) Confusion matrix for the BiLSTM-MultiheadAttention model.

Exploring the impact of adversarial attacks on deep model performance

Experimental results and analysis

For the public epilepsy data set, FGSM, BIM and PGD attack algorithms are used to attack the trained NHANet model.

Furthermore, to explore the influence of different epsilons on deep learning models, the epsilons were set to 1/255, 2/255, 3/255, 4/255, and 5/255, and the evaluation indicators of adversarial accuracy, attack success rate, average L_1 distance, and average L_2 distance were used for verification. The research results show that, with the continuous increase of perturbation intensity, the adversarial accuracy shows a gradual downward trend, while the attack success rate continues to increase. Figure 5 shows the line charts of the model adversarial accuracy and attack success rate as the perturbation intensity increases. At the same time, the specific performance of NHANet model under different attack algorithms with different epsilon values are shown in Tables 8, 9 and 10 below.

Specifically, without adding perturbation, the adversarial accuracy of NHANet model reached 0.9840, indicating that the model has good performance on the original test set. However, as the perturbation value continues to increase, the adversarial accuracy gradually decreases. When the epsilon is set to 5/255, the adversarial accuracy rate of FGSM attack algorithm is 0.1307, that of PGD attack algorithm is 0.1413, and that of BIM attack algorithm is 0.0840. This indicates that the accuracy rate of model classification will decline sharply with the increase of perturbation. At the same time, the attack success rate increases with the increase of perturbation value, which means that the stability of NHANet model's classification prediction is poor when facing small changes in input data. In addition, the average L_1 distance and the average L_2 distance increase with the increase of the perturbation value, indicating that the difference between the adversarial sample and the original sample increases with the increase of the perturbation value.

At the same time, it can be seen that when the epsilons are set to 1/255, 2/255, 3/255, 4/255, the attack success rate of PGD and BIM algorithm is higher than that of FGSM algorithm. This phenomenon shows that PGD and BIM algorithms have stronger attack capability at lower perturbation levels. In addition, the average L_1 distance and average L_2 distance are important indicators for measuring the differences between adversarial samples and

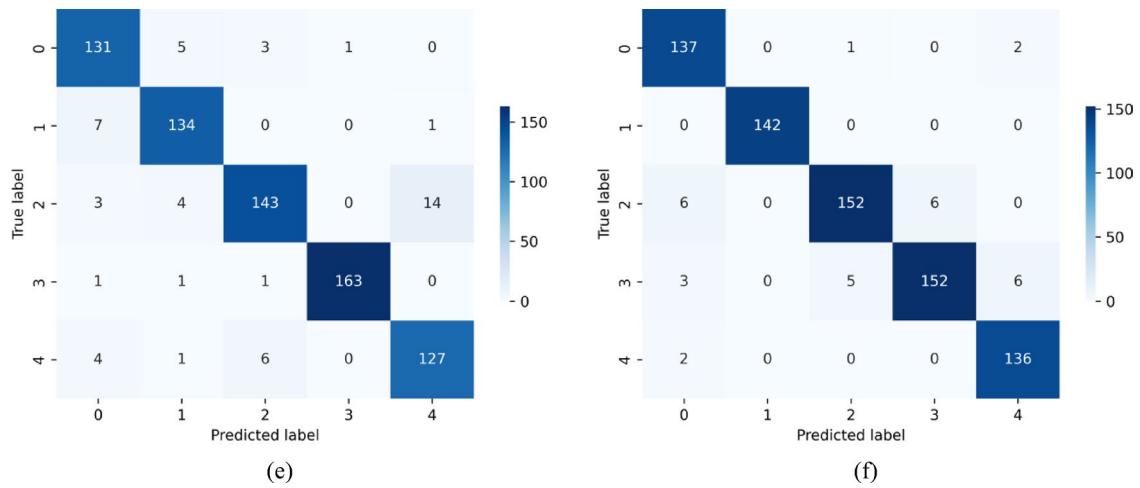


Fig. 4. (continued)

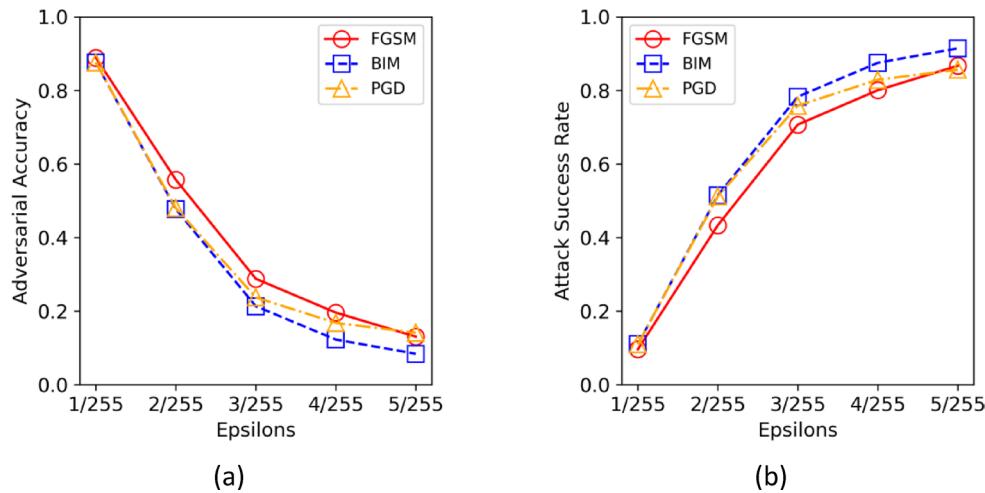


Fig. 5. Line charts showing the variation of adversarial accuracy and attack success rate of the NHANet model as the perturbation strength increases. **(a)** Line chart for the adversarial accuracy of the NHANet model as it changes with perturbation. **(b)** Line chart for the attack success rate of the NHANet model as it changes with perturbation.

	Original	$\epsilon = 1/255$	$\epsilon = 2/255$	$\epsilon = 3/255$	$\epsilon = 4/255$	$\epsilon = 5/255$
Adversarial accuracy	0.9840	0.8893	0.5573	0.2880	0.1960	0.1307
Attack success rate	0.0	0.0962	0.4336	0.7073	0.8008	0.8672
Average L_1 distance	0.0	1.1759	2.3518	3.5277	4.7035	5.8792
Average L_2 distance	0.0	0.0679	0.1358	0.2037	0.2716	0.3395

Table 8. Performance of NHANet model with different epsilons under FGSM attack algorithm.

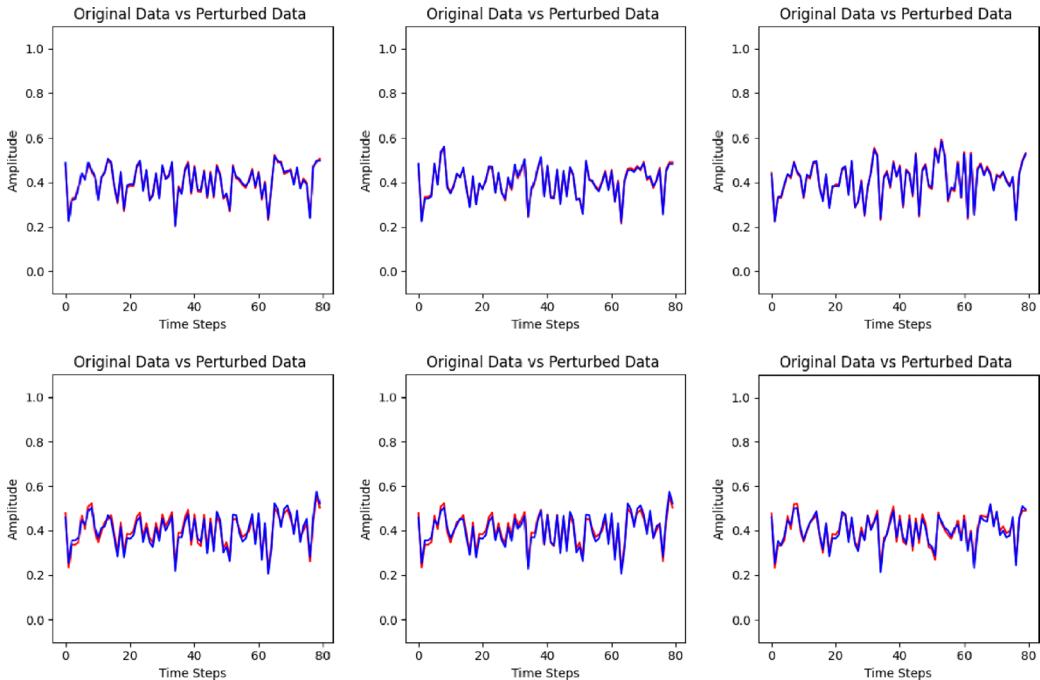
original samples. The average L_1 distance and average L_2 distance of adversarial samples generated by FGSM attack algorithm are both greater than those generated by PGD and BIM attack algorithms, indicating that the adversarial samples generated by PGD and BIM attack algorithms have less difference from the original samples.

In order to verify whether the generated adversarial samples are covert, some adversarial samples are randomly selected and compared with the original data. The first 80 characteristic values of each sample are printed. The red line represents the original data, and the blue line represents the generated adversarial sample. As shown in Fig. 6, the first row shows the comparison between the adversarial samples generated by FGSM, BIM, and PGD and the original data, with the epsilon set to 2/255. The second row shows the comparison

	Original	$\epsilon = 1/255$	$\epsilon = 2/255$	$\epsilon = 3/255$	$\epsilon = 4/255$	$\epsilon = 5/255$
Adversarial accuracy	0.9840	0.8760	0.4800	0.2373	0.1680	0.1413
Attack success rate	0.0	0.1098	0.5122	0.7588	0.8293	0.8564
Average L_1 distance	0.0	1.1553	2.2143	3.1076	3.6757	4.0910
Average L_2 distance	0.0	0.0673	0.1306	0.1861	0.2264	0.2622

Table 9. Performance of NHANet model with different epsilons under PGD attack algorithm.

	Original	$\epsilon = 1/255$	$\epsilon = 2/255$	$\epsilon = 3/255$	$\epsilon = 4/255$	$\epsilon = 5/255$
Adversarial accuracy	0.9840	0.8760	0.4773	0.2133	0.1227	0.0840
Attack success rate	0.0	0.1098	0.5149	0.7832	0.8753	0.9146
Average L_1 distance	0.0	1.1665	2.2306	3.2206	4.0604	4.9083
Average L_2 distance	0.0	0.0676	0.1307	0.1914	0.2445	0.2995

Table 10. Performance of NHANet model with different epsilons under BIM attack algorithm.**Fig. 6.** Comparison of raw samples and adversarial samples generated under different attack algorithms.

between the adversarial samples generated by FGSM, BIM, and PGD and the original data, with the epsilon set to 5/255.

Based on the above analysis, the smaller the perturbation, the more difficult it is for the human eye to detect the adversarial samples, while the larger the perturbation, the more obvious the deviation of the generated adversarial samples from the original data. Therefore, in practical application scenarios, in order to ensure the robustness and security of the deep learning model, it is very important to select appropriate perturbation values. This decision needs to comprehensively consider the robustness requirements and security factors of the model to seek the best balance between the two.

Further investigation into the impact of adversarial attacks on deep learning models

In order to further explore the impact of adversarial attacks on deep learning models, the five aforementioned deep learning models were subjected to adversarial attacks using the PGD attack algorithm. Furthermore, metrics such as adversarial accuracy, attack success rate, average L_1 distance and average L_2 distance were used to evaluate the performance of the generated adversarial samples. Figure 7 shows the line charts of the different models' adversarial accuracy and attack success rate with increasing perturbation value, respectively. Meanwhile, the specific performance of the five deep learning models under different epsilons is detailed in Table 11 below.

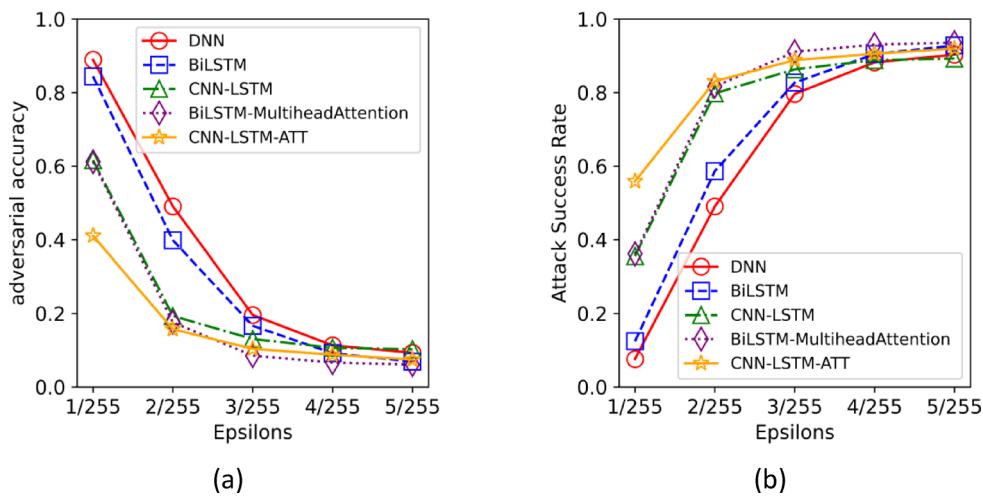


Fig. 7. Line charts illustrating the variation of adversarial accuracy and attack success rate for different models as the perturbation intensity increases. **(a)** Line chart showing the adversarial accuracy of various models in response to changes in perturbation. **(b)** Line chart depicting the attack success rate of various models in response to changes in perturbation.

Neural network model	Evaluation metrics	ϵ (epsilons)					
		original	1/255	2/255	3/255	4/255	5/255
DNN	Adversarial accuracy	0.9627	0.8893	0.4907	0.1960	0.1133	0.0933
	Attack success rate	0.0	0.0762	0.4903	0.7964	0.8823	0.9030
	Average L_1 distance	0.0	1.1702	2.2979	3.3012	3.8961	4.2859
	Average L_2 distance	0.0	0.0677	0.1338	0.1937	0.2355	0.2715
BiLSTM	Adversarial accuracy	0.9640	0.8440	0.3987	0.1667	0.0920	0.0693
	Attack success rate	0.0	0.1245	0.5864	0.8271	0.9046	0.9281
	Average L_1 distance	0.0	1.1719	2.3116	3.3308	3.9296	4.3159
	Average L_2 distance	0.0	0.0678	0.1343	0.1949	0.2369	0.2730
CNN-LSTM	Adversarial accuracy	0.9560	0.6160	0.1933	0.1307	0.1067	0.1027
	Attack success rate	0.0	0.3556	0.7978	0.8633	0.8884	0.8926
	Average L_1 distance	0.0	1.1421	2.1842	3.0447	3.6027	4.0221
	Average L_2 distance	0.0	0.0666	0.1291	0.1832	0.2231	0.2587
BiLSTM-MultiheadAttention	Adversarial accuracy	0.9587	0.6120	0.1760	0.0853	0.0667	0.0613
	Attack success rate	0.0	0.3616	0.8164	0.9110	0.9305	0.9360
	Average L_1 distance	0.0	1.1703	2.2859	3.2548	3.8348	4.2278
	Average L_2 distance	0.0	0.0677	0.1334	0.1919	0.2330	0.2687
CNN-LSTM-ATT	Adversarial accuracy	0.9307	0.4107	0.1573	0.1040	0.0880	0.0747
	Attack success rate	0.0	0.5587	0.8309	0.8883	0.9054	0.9198
	Average L_1 distance	0.0	1.1120	2.0774	2.8439	3.3736	3.8137
	Average L_2 distance	0.0	0.0657	0.1248	0.1746	0.2130	0.2483

Table 11. Performance of various models under different epsilons.

The experimental results indicate that, when the values of alpha and steps are set to be the same, as the perturbation value continues to increase, the performance of different neural network prediction models steadily declines. Specifically, the adversarial accuracy decreases with increasing perturbation value, and the generated adversarial samples can effectively deceive the model into making wrong classifications. Meanwhile, the attack success rate increases with increasing perturbation value, which indicates that the model has weaker robustness in the face of adversarial attacks. In addition, the average L_1 distance and average L_2 distance increase with the increase of perturbation value, which indicates that the generated adversarial samples are more and more deviated from the original data, resulting in the decrease of the concealment of the adversarial samples. Through this above analysis, the adversarial attack algorithm can not only cause classification errors in the NHANet model, but also be equally effective for other neural network models. This fully demonstrates the importance and urgency of improving the robustness of deep learning models in practical applications.

	Precision	Recall	F1-score	support
0	0.98571	0.98571	0.98571	140
1	1.00000	1.00000	1.00000	142
2	0.99383	0.98171	0.98773	164
3	1.00000	1.00000	1.00000	166
4	0.98571	1.00000	0.99281	138
Accuracy			0.99333	750
Macro avg.	0.99305	0.99348	0.99325	750
Weighted avg.	0.99335	0.99333	0.99333	750

Table 12. Classification report of raw data after adversarial training under FGSM attack algorithm.

	Precision	Recall	F1-score	Support
0	0.98561	0.97857	0.98208	140
1	1.00000	1.00000	1.00000	142
2	0.99383	0.98171	0.98773	164
3	1.00000	1.00000	1.00000	166
4	0.97872	1.00000	0.98925	138
Accuracy			0.99200	750
Macro avg.	0.99163	0.99206	0.99181	750
Weighted avg.	0.99205	0.99200	0.99199	750

Table 13. Classification report of raw data after adversarial training under BIM attack algorithm.

	Precision	Recall	F1-score	Support
0	0.98561	0.97857	0.98208	140
1	1.00000	1.00000	1.00000	142
2	0.98773	0.98171	0.98471	164
3	1.00000	1.00000	1.00000	166
4	0.97857	0.99275	0.98561	138
Accuracy			0.99067	750
Macro avg.	0.99038	0.99061	0.99048	750
Weighted avg.	0.99069	0.99067	0.99066	750

Table 14. Classification report of raw data after adversarial training under PGD attack algorithm.

Neural hybrid assembly network incremental adversarial training

The effectiveness of incremental adversarial training

In order to verify the effectiveness of the incremental adversarial training algorithm proposed in this paper under different attack algorithms, three attack algorithms (FGSM, BIM, and PGD) were used to generate adversarial samples during the adversarial training process. And use accuracy, precision, recall, and F1-score as evaluation metrics to comprehensively evaluate the performance of the model after adversarial training. The following Tables 12, 13 and 14 are the classification reports of different algorithms.

According to the classification report, after adversarial training, the model performs well in classification performance on each category. For the FGSM attack algorithm, the highest precision for each category reaches 1.0000 and the lowest is 0.98571, and the recall and F1-score for each category also achieve good prediction results, which indicates that the model after adversarial training is able to recognize the samples in each category effectively. Similarly, for the BIM attack algorithm and the PGD attack algorithm, the overall accuracy of the model reaches 0.99200 and 0.99067, respectively, thus further verifying the effectiveness of the adversarial training method proposed in this paper under different attack algorithms.

Variable parameter analysis

When using the IncAT algorithm to continuously learn the generated adversarial samples, λ , as a hyperparameter, is used to adjust the constraint degree of the deep learning model to the original samples when learning the adversarial samples. In this experiment, when using the FGSM attack algorithm to generate adversarial samples for adversarial training, we investigated the impact of the value of λ on the robustness and performance of deep learning models. By systematically testing the model performance under different λ values, an optimal configuration is sought to improve the model's ability to resist adversarial attacks without damaging the model's

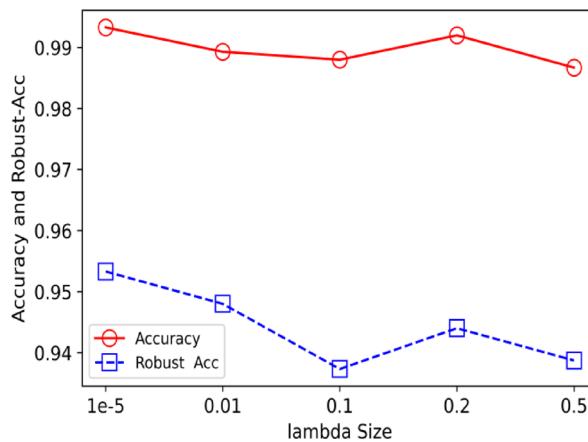


Fig. 8. Different λ Accuracy and Robust Accuracy.

	$\lambda = 1e-5$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$
Accuracy	0.9933	0.9893	0.9880	0.9920	0.9867
Precision	0.9931	0.9890	0.9878	0.9917	0.9862
Recall	0.9935	0.9894	0.9880	0.9923	0.9872
F1-score	0.9933	0.9892	0.9878	0.9919	0.9865
Robust-accuracy	0.9533	0.9480	0.9373	0.9440	0.9387

Table 15. Model performance with different λ parameters.

performance in the original data. In the experiment, we tried different sizes of λ , Specifically including: $1e-5$, 0.01 , 0.1 , 0.2 , 0.5 . The line chart in Fig. 8 shows the variation of accuracy and robust accuracy with increasing λ . The specific performance of NHANet models with different λ values is shown in Table 15.

By observing the experimental results, it can be seen that when λ is $1e-5$, the model's reaches the highest accuracy of 0.9933, as well as the highest level of robust accuracy. However, as the value of λ increases, the robust accuracy of the model shows a decreasing trend, which suggests that too large a value of λ may affect the model's ability to learn adversarial samples. Therefore, choosing an appropriate λ value is crucial to balance the robust accuracy of the deep learning model and the accuracy of the original dataset. In practical applications, in order to balance the performance of the model on both the adversarial samples and the original samples, the value of the parameter λ should be flexibly adjusted according to the needs of specific tasks.

Comparison of methods

In order to comprehensively evaluate the performance and advantages of the incremental adversarial training algorithm proposed in this paper, we conducted a detailed comparative analysis with existing adversarial training methods. In the comparative experiments, we used three attack algorithms, namely FGSM, BIM, and PGD, to generate adversarial samples during the adversarial training process. In terms of evaluation metrics, we used robust accuracy to measure the ability of deep learning models to resist adversarial attacks after adversarial training. At the same time, to evaluate the classification ability of the model on the original data after adversarial training, we also used four indicators: accuracy, precision, recall, and F1-score. These indicators reflect the classification performance of the model from different perspectives and can comprehensively evaluate the model's performance on the original data. The specific methods listed in the table are as follows: Method one involves mixing the original training data with the adversarial samples generated for the current model in a 1:1 ratio to form an expanded augmented training set. Then, this integrated training set is used to retrain the deep learning model. This method aims to improve the ability to resist adversarial attacks by simultaneously learning the features of raw data and perturbed data. Method two first calculates the loss values of the original data and the adversarial samples separately; then, the two loss values are weighted and summed to obtain the total loss; finally, backpropagation is performed based on this total loss to enhance the robustness of the deep learning model. The formula of the loss function of Method two is shown in (8).

$$loss = alpha * loss_original + (1 - alpha) * loss_adversarial \quad (8)$$

Where alpha is the weight value, loss-original is the loss value of the original data, and loss-adversarial is the loss value of the adversarial sample. The scatter plots of accuracy and robust accuracy of different methods under different attack algorithms are shown in Fig. 9.

From Table 16, it can be analyzed that when facing the three different attack algorithms, FGSM, PGD, and BIM, the method proposed in this paper has demonstrated significant performance advantages. Specifically,

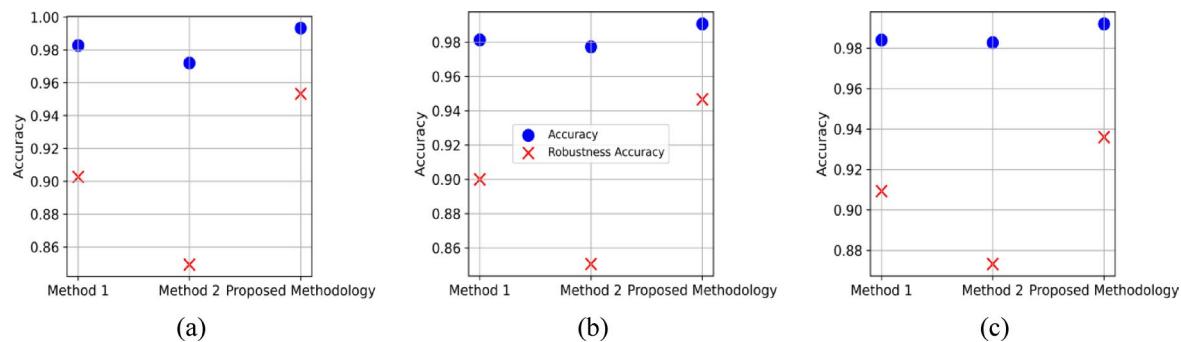


Fig. 9. Scatter plots depicting the accuracy and robust accuracy of various methods under different attack algorithms. (a) Scatter plot for the accuracy and robust accuracy of different adversarial training methods under the FGSM attack algorithm. (b) Scatter plot for the accuracy and robust accuracy of different adversarial training methods under the PGD attack algorithm. (c) Scatter plot for the accuracy and robust accuracy of different adversarial training methods under the BIM attack algorithm.

Attack methods	Adversarial training methods	Evaluation metrics				
		Accuracy	Precision	Recall	F1-score	Robust-accuracy
FGSM	Method one	0.9827	0.9823	0.9829	0.9826	0.9027
	Method two	0.9720	0.9715	0.9721	0.9716	0.8493
	Proposed methodology	0.9933	0.9931	0.9935	0.9933	0.9533
PGD	Method one	0.9813	0.9808	0.9817	0.9811	0.9000
	Method two	0.9773	0.9973	0.9779	0.9773	0.8507
	Proposed methodology	0.9907	0.9904	0.9906	0.9905	0.9467
BIM	Method one	0.9840	0.9838	0.9847	0.9840	0.9093
	Method two	0.9827	0.9824	0.9827	0.9825	0.8733
	Proposed methodology	0.9920	0.9916	0.9921	0.9918	0.9360

Table 16. Model performance of different adversarial training methods.

Execution time	Method one	Method two	Proposed methodology
FGSM	780.24 s	503.91 s	696.42 s
PGD	2142.36 s	1356.21 s	2051.28 s
BIM	2133.87 s	1279.33 s	2038.46 s

Table 17. The execution times of different adversarial training methods.

the robust accuracy rates reached 95.33%, 94.67%, and 93.60% respectively. Compared with Method One, they increased by 5.06%, 4.67%, and 2.67% respectively, and compared with Method Two, they increased by 10.40%, 9.60%, and 6.27% respectively. This proves the outstanding effect of this method in improving the robustness of the model. Moreover, the method proposed in this paper not only achieved excellent robust accuracy rates on adversarial samples, but also maintained a high classification accuracy on clean samples. This phenomenon indicates that during the incremental adversarial training process, the model can effectively learn the features of adversarial samples without damaging its understanding and classification ability of the original dataset. In conclusion, the adversarial training method proposed in this paper significantly improves the robustness and security of the model without the need to retrain the entire model.

Furthermore, this study not only focuses on the robustness assessment of deep learning models, but also incorporates time efficiency into the comprehensive evaluation system to comprehensively measure the practical deployment value of the method. The methods listed in the table are consistent with the definitions mentioned earlier. The running times of different adversarial training methods is shown in Table 17.

According to the data in the table, method two has the shortest time consumption in all three attack scenarios, significantly lower than method one and the method proposed in this paper. However, the method proposed in this paper utilizes the Fisher matrix to accurately evaluate the importance of each parameter and impose constraints, enabling deep learning models to continuously learn adversarial features with only a controllable increase in computational overhead, thus achieving an improvement in model robustness. Based on the above analysis, IncAT incremental adversarial training can not only effectively improve the robustness of deep learning models, but also enhance the model's ability to resist adversarial attacks in real-time in application scenarios.

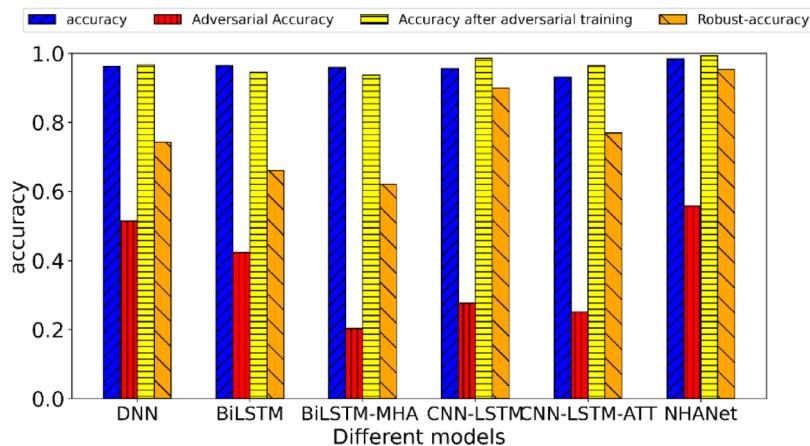


Fig. 10. Comparison of accuracy of various deep learning models before and after FGSM algorithm attack and after adversarial training.

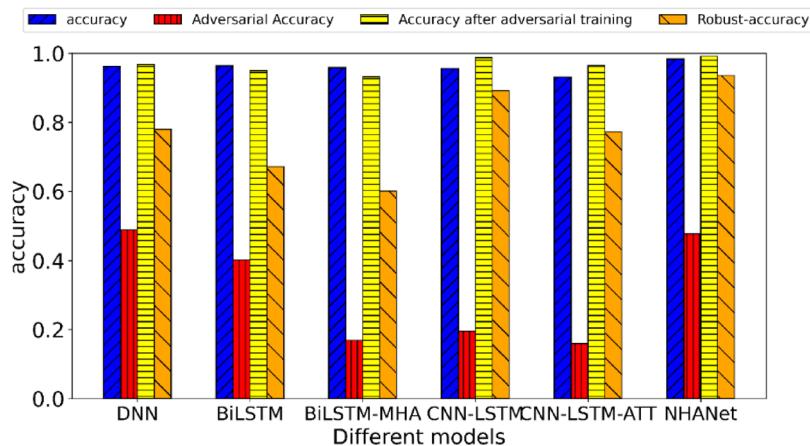


Fig. 11. Comparison of accuracy of various deep learning models before and after BIM algorithm attack and after adversarial training.

that require rapid iteration and response. In addition, the core idea also provides reference and inspiration for building a fast and real-time adversarial training system, and is expected to promote further breakthroughs in this field.

The universality of incremental adversarial training

In order to verify the universality of the proposed method in this paper to different deep learning models, the above six deep learning models are tested. At the same time, to demonstrate the effectiveness of the proposed method in improving accuracy, the accuracy of the model without adversarial attacks, the accuracy of the model after attacks, and the robustness of the model after adversarial training were compared with the accuracy of the original dataset. The comparison chart of the accuracy of various deep learning models before and after attacks by different attack algorithms, as well as after adversarial training, is shown in Figs. 10, 11 and 12. The specific metrics of the deep learning models after adversarial training under different attack algorithms are presented in Tables 18, 19 and 20.

In summary, after incremental adversarial training, not only has the performance of the NHANet model been significantly improved, but the performance of other deep learning models has also been notably enhanced after adopting this method. Meanwhile, when facing different attack algorithms, this method also exhibits good performance in other deep models. Thus, it can be concluded that the adversarial training method proposed in this paper is not only effective for the NHANet model, but also has wide generality, and can significantly enhance other deep learning models' ability to resist adversarial attacks.

Related works

Adversarial Training, as a core defensive technology for enhancing the robustness of models, strengthens the model's ability to resist interference by injecting adversarial samples during the training process. At present, research on adversarial training is mainly divided into the following categories²³: 1) Accelerated adversarial

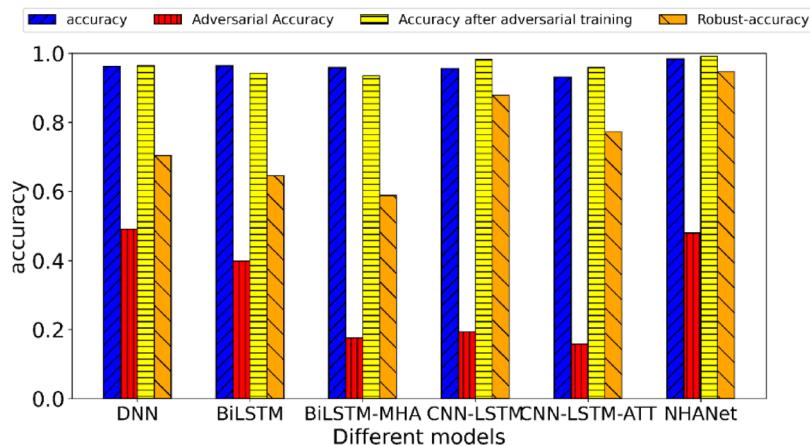


Fig. 12. Comparison of accuracy of various deep learning models before and after PGD algorithm attack and after adversarial training.

	Accuracy	Precision	Recall	F1-score	Robust-accuracy
DNN	0.9667	0.9678	0.9678	0.9668	0.7427
BiLSTM	0.9453	0.9466	0.9452	0.9454	0.6600
BiLSTM-MultiheadAttention	0.9373	0.9389	0.9395	0.9369	0.6200
CNN-LSTM	0.9853	0.9852	0.9856	0.9853	0.9000
CNN-LSTM-ATT	0.9640	0.9633	0.9647	0.9636	0.7693
NHANet	0.9933	0.9931	0.9935	0.9933	0.9533

Table 18. Performance of each model after adversarial training under FGSM attack algorithm.

	Accuracy	Precision	Recall	F1-score	Robust-accuracy
DNN	0.9680	0.9677	0.9692	0.9680	0.7800
BiLSTM	0.9493	0.9491	0.9511	0.9492	0.6720
BiLSTM-MultiheadAttention	0.9320	0.9326	0.9353	0.9318	0.6013
CNN-LSTM	0.9880	0.9878	0.9882	0.9880	0.8920
CNN-LSTM-ATT	0.9653	0.9649	0.9648	0.9647	0.7720
NHANet	0.9920	0.9916	0.9921	0.9918	0.9360

Table 19. Performance of each model after adversarial training under BIM attack algorithm.

	Accuracy	Precision	Recall	F1-score	Robust-accuracy
DNN	0.9640	0.9651	0.9651	0.9641	0.7040
BiLSTM	0.9427	0.9434	0.9447	0.9429	0.6453
BiLSTM-MultiheadAttention	0.9347	0.9365	0.9353	0.9346	0.5880
CNN-LSTM	0.9827	0.9823	0.9831	0.9827	0.8787
CNN-LSTM-ATT	0.9587	0.9579	0.9589	0.9579	0.7733
NHANet	0.9907	0.9904	0.9906	0.9905	0.9467

Table 20. Performance of each model after adversarial training under PGD attack algorithm.

training, which aims to improve the efficiency of adversarial training; 2)Parameter adaptive adversarial training, which can automatically adjust parameters according to the actual training situation; 3)Semi supervised or unsupervised adversarial training, which expands the dataset by utilizing unlabeled samples and applies them to adversarial training to enhance the model's generalization ability.

Accelerated adversarial training

In order to reduce the time of adversarial training, researchers have proposed some efficient adversarial training methods. Shafahi et al.²⁴ proposed Free AT, which generates adversarial samples by recovering the gradient information calculated when updating model parameters, thereby eliminating the computational cost of generating adversarial samples. This method can significantly improve the efficiency and scalability of adversarial training while maintaining robustness similar to standard adversarial training algorithms. Zheng et al.²⁵ generated similar or stronger adversarial samples with fewer iterations by accumulating adversarial perturbations over epochs. In addition, researchers use single step adversarial training to accelerate, optimizing the FGSM attack algorithm to generate adversarial samples. However, in the single step adversarial training method, if the attack step size of FGSM is too large, the deep learning model will produce distorted decision boundaries, leading to the occurrence of CO phenomenon. Therefore, Wong et al.¹⁶ proposed a randomly initialized FGSM internal maximization attack method to alleviate the occurrence of CO phenomenon.

Parameter adaptive adversarial training

The setting of parameters has a crucial impact on improving the robustness of deep learning models when generating adversarial samples using gradient information. Therefore, carefully selecting and adjusting these parameters is a key step in improving the stability and reliability of the model in the face of potential adversarial attacks. Cheng et al.²⁶ proposed an adversarial training method with adaptive perturbation constraints, which adaptively seeks the minimum perturbation that can cause deep learning models to misclassify. Specifically, the initial perturbation constraint is 0, and after each iteration, a specific constant is added until the deep learning model misclassifies. WU Jinfu²⁷ proposed a Fast AT method based on random noise and adaptive step size. This method utilizes random noise for data augmentation and accumulates the gradient of adversarial samples during the training process to adjust the step size of adversarial samples, thereby generating adversarial samples that are more conducive to model training.

Semi-supervised or unsupervised adversarial training

Existing research has found that adversarial training requires the use of much larger datasets than standard training, which incurs significant costs. Therefore, researchers adopt semi supervised or unsupervised methods for adversarial training. Aim to improve the adversarial robustness of deep learning models solely by adding unlabeled data. Carmon et al.²⁸ proposed the robust adversarial training method RST. This method utilizes the target model to predict pseudo labels for unlabeled samples, and simultaneously uses both labeled and unlabeled samples to predict pseudo labels for retraining the deep learning model. This method also demonstrates the importance of unlabeled samples in improving the robustness of deep learning models. Uesato et al.²⁹ also verified through experiments that unlabeled samples can improve the robustness of the model. The more unlabeled samples used, the stronger the generalization ability of the deep learning model.

In conclusion, with the continuous development of research on adversarial training, a variety of innovative adversarial training methods have emerged, enabling them to demonstrate greater robustness when dealing with various adversarial attacks. However, the current research integrating incremental learning theory into adversarial training is still relatively limited. Based on this, this paper proposes to introduce the incremental learning algorithm into the adversarial training framework. By continuously learning from dynamically generated adversarial samples, it endows the deep learning model with more flexible dynamic adaptability to cope with complex and variable adversarial scenarios.

Conclusions and future work

In response to the problems of low efficiency and high computational resource consumption in existing adversarial training methods, this paper proposes an incremental adversarial training method. This method enables deep learning models to enhance their security and robustness without the need to retrain the entire network, fundamentally addressing the efficiency and performance bottlenecks of traditional adversarial training algorithms in the field of brain-computer interfaces. Taking the medical diagnosis system as an example, the advantages of IncAT are particularly significant in complex medical data scenarios. When encountering sudden equipment noise or malicious attacks from attackers, IncAT can learn and adapt to new adversarial sample features in real time, and promptly update its recognition strategy. This ensures the reliability of the diagnostic model, reduces the misdiagnosis rate caused by adversarial samples, and thus provides doctors with more accurate diagnostic basis. Moreover, to verify the effectiveness of this adversarial training method, experiments were conducted on the publicly available epilepsy brain-computer interface dataset from the University of Bonn, and the FGSM, BIM, PGD, etc. attack algorithms were used to conduct comparative experiments with this method and other adversarial training methods. The experimental results show that the proposed method effectively solves the problem of deep neural networks being unable to continuously learn adversarial samples and demonstrates excellent adversarial robustness to various adversarial attacks and different deep learning models. However, although this method has achieved certain results in improving the robustness and security of the model, it shows certain limitations when the data category changes. Therefore, future research will focus on studying the adversarial training methods based on category increment, in order to more efficiently cope with the dynamic changes of data categories. At the same time, this method will be extended to cross-domain scenarios such as autonomous driving and financial risk control. This will help verify its universality and further comprehensively evaluate its generalization ability.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 5 February 2025; Accepted: 10 September 2025

Published online: 14 October 2025

References

1. Amin, S. U., Alsulaiman, M., Muhammad, G., Mekhtiche, M. A. & Shamim Hossain, M. Deep learning for EEG motor imagery classification based on multi-layer CNNs feature fusion. *Future Gener. Comput. Syst.* **101**, 542–554. <https://doi.org/10.1016/j.future.2019.06.027> (2019).
2. Li, D. L., Xu, J. C., Wang, J. H., Fang, X. K. & Ji, Y. A Multi-Scale fusion convolutional neural network based on attention mechanism for the visualization analysis of EEG signals decoding. *IEEE Trans. Neural Syst. Rehabil. Eng.* **28**(12), 2615–2626. <https://doi.org/10.1109/TNSRE.2020.3037326> (2020).
3. Zou, Z. X., Chen, K. Y., Shi, Z. W., Guo, Y. H. & Ye, J. P. Object detection in 20 years: A survey. *Proc. IEEE* **111**(3), 257–276. <https://doi.org/10.1109/JPROC.2023.3238524> (2023).
4. Cai, Z. W. & Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(5), 1483–1498. <https://doi.org/10.1109/TPAMI.2019.2956516> (2021).
5. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* 580–587 (IEEE, USA, 2014). <https://doi.org/10.1109/CVPR.2014.81>.
6. Zhang, H., Xue, J. & Dana, K. Deep ten: Texture encoding network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition* 2896–2905 (IEEE, USA, 2017). <https://doi.org/10.1109/CVPR.2017.309>.
7. Luo, Y. Z., Huang, Q. H. & Li, X. L. Segmentation information with attention integration for classification of breast tumor in ultrasound image. *Pattern Recognit.* **124**, 108427. <https://doi.org/10.1016/j.patcog.2021.108427> (2022).
8. Roy, S. K., Krishna, G., Dubey, S. R. & Chaudhuri, B. B. Hybridsn: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **17**(2), 277–281. <https://doi.org/10.1109/LGRS.2019.2918719> (2020).
9. Li, Y., Zhang, H. & Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **9**(1), 67. <https://doi.org/10.3390/rs9010067> (2017).
10. Zhang, Q. et al. Recyclable waste image recognition based on deep learning. *Resour. Conserv. Recy.* **171**, 105636. <https://doi.org/10.1016/j.resconrec.2021.105636> (2021).
11. Li, Y. F. & Liu, W. Deep learning-based garbage image recognition algorithm. *Appl. Nanosci.* **13**(2), 1415–1424. <https://doi.org/10.1007/s13204-021-02068-z> (2023).
12. Szegedy, C. et al. Intriguing properties of neural networks. In *2014 International Conference on Learning Representations* 1312.6199 (ICLR, Canada, 2013).
13. Zhang, Q. L. et al. Beyond imagenet attack: Towards crafting adversarial examples for black-box domains. In *The Tenth International Conference on Learning Representations* 1–18 (ICLR, Virtual, 2022).
14. Chen, J. Q. et al. Diffusion models for imperceptible and transferable adversarial attack. *IEEE Trans. Pattern Anal. Mach. Intell.* <https://doi.org/10.48550/arXiv.2305.08192> (2024).
15. Mądry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. Towards deep learning models resistant to adversarial attacks. In *The Sixth International Conference on Learning Representations* 1706.06083 (ICLR, Canada, 2017).
16. Wong, E., Rice, L. & Kolter, J. Z. Fast is better than free: Revisiting adversarial training. In *The Eighth International Conference on Learning Representations* 1–12 (ICLR, Virtual, 2020).
17. Rice, L., Wong, E. & Kolter, Z. Overfitting in adversarially robust deep learning. In *The Thirty-Seventh International Conference on Machine Learning* 8093–8104 (PMLR, Vienna, 2020).
18. Zhang, H. Y. et al. Theoretically principled trade-off between robustness and accuracy. In *The Thirty-Sixth International Conference on Machine Learning* 7472–7482 (PMLR, USA, 2019).
19. Goodfellow, I. J., Shlens, J. & Szegedy, C. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations* 1412.6572 (ICLR, Singapore, 2014).
20. Kurakin, A., Goodfellow, I. & Bengio, S. Adversarial machine learning at scale. In *5th International Conference on Learning Representations* 1–17 (ICLR, France, 2016).
21. Andrzejak, R. G. et al. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Phys. Rev. E* **64**(6), 061907. <https://doi.org/10.1103/PhysRevE.64.061907> (2001).
22. Li, X. Y., Yang, K., Tu, G. Q. & Liu, S. B. Adversarial sample generation method for time-series data based on local augmentation. *J. Comput. Appl.* 1–11 (2024).
23. Sui, C. H. et al. A survey on adversarial training for robust learning. *Int. J. Image Graph.* **28**(12), 3629–3650. <https://doi.org/10.11834/jig.220953> (2023).
24. Shafahi, A. et al. Adversarial training for free! In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019* 3358–3369 (NeurIPS, Canada, 2019).
25. Zheng, H. Z., Zhang, Z. Q., Gu, J. C., Lee, H. & Prakash, A. Efficient adversarial training with transferable adversarial examples. In *2020 IEEE/CVF Conference on Computer Vision and Pattern* 1181–1190 (IEEE, USA, 2020). <https://doi.org/10.1109/CVPR42600.2020.00126>.
26. Cheng, M. H., Lei, Q., Chen, P. Y., Dhillon, I. & Hsieh, C. J. Cat: Customized adversarial training for improved robustness. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence* 2002.06789 (IJCAI, Austria, 2020).
27. Wu, J. F. & Liu, Y. Fast adversarial training method based on random noise and adaptive step size. *J. Comput. Appl.* **44**(6), 1807–1815 (2024).
28. Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P. & Duchi, J. C. Unlabeled data improves adversarial robustness. In *2019 Conference on Neural Information Processing Systems* 1905.13736 (NeurIPS, Canada, 2019).
29. Uesato, J. et al. Are labels required for improving adversarial robustness? In *2019 Conference on Neural Information Processing Systems* 6609 (NeurIPS, Canada, 2019).

Author contributions

Yuxin Ge executed the program, designed the work, and wrote the manuscript. Yanhua Dong and Hongyu Sun controlled the direction and progress of the experiment. Yuetong Liu and Chengli Wang organized the experimental results and typeset the paper.

Funding

This work was supported in part by the Science and Technology Development Project of the Department of Education of Jilin Province (JJKH20250945KJ), New Generation Information Technology Innovation Project of China University Industry, University and Research Innovation Fund (2022IT096), Jilin Province Innovation and Entrepreneurship Talent Project (2023QN31), Natural Science Foundation of Jilin Province (No.YDZ-J202301ZYTS157, 20240601034RC, 20240304097SF), and Innovation Project of Jilin Provincial Development and Reform Commission (2021C038-7).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.D. or H.S.

Reprints and permissions information is available at www.nature.com/reprints.

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, 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 you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. 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-nc-nd/4.0/>.

© The Author(s) 2025