frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 02 January 2026
DOI 10.3389/fphy.2025.1660104

OPEN ACCESS

EDITED BY
Youssri Hassan Youssri,
Egypt Uinversity of Informatics, Egypt

REVIEWED BY
Miodrag Zivkovic,
Singidunum University, Serbia
Weisha Zhang,
University of Electronic Science and Technology of China, China
Yuan Tang,
Chengdu University of Technology, China

*CORRESPONDENCE
Bo Li,
7816501@qq.com

RECEIVED 05 July 2025
REVISED 18 November 2025
ACCEPTED 19 November 2025
PUBLISHED 02 January 2026

CITATION
Zhou W, Li B and Zeng X (2026) Intrusion detection model of UAV system based on machine learning and neural network. *Front. Phys.* 13:1660104.
doi: 10.3389/fphy.2025.1660104

Intrusion detection model of UAV system based on machine learning and neural network

Wengang Zhou[1,2], Bo Li[3,4]* and Xue Zeng[4,5]

[1]Civil Aviation Flight University of China, Deyang, Sichuan, China, [2]Sichuan Provincial Engineering Research Center of Domestic Civil Aircraft Flight and Operation Support, Deyang, Sichuan, China, [3]Information Department of Pengzhou People's Hospital, Chengdu, China, [4]Advanced Cryptography and System Security Key Laboratory of Sichuan Province, Chengdu, China, [5]Chengdu University of Information Technology, Chengdu, China

Introduction: With the growing complexity and scale of cyber attacks, intrusion detection for unmanned aerial vehicle (UAV) systems has become a critical challenge in modern network security. UAVs have unique constraints including limited battery life, restricted data-transmission distance, and small data-storage capacity, while malicious activities can disrupt their power usage, communication, and data storage—highlighting the need for dedicated intrusion-detection solutions. Traditional traffic detection methods lack efficient modeling of local and global features, making it difficult to capture complex data patterns.

Methods: We propose an intrusion detection model integrating machine learning and neural networks. First, UAV data is cleaned, and traditional feature selection techniques (filtering, packaging, embedding) are used to separate key and non-key features. Non-key features are mapped to the key feature subspace via CNN + LSTM for feature fusion, and the fused features serve as model inputs. Machine learning and neural networks are then combined to detect UAV network traffic.

Results: Testing on public datasets ISCXVPN2016, CICIDS2018, TON IoT, and CIC IoT 2023 shows that our method improves accuracy by up to 3%, F1 score by up to 4%, and recall by up to 3% compared to the three major feature selection techniques.

Discussion: The integration of CNN + LSTM enables effective modeling of local and global features, addressing the limitations of traditional methods. The model's optimization for feature fusion and UAV-specific constraints ensures it is suitable for resource-constrained UAV systems, providing reliable intrusion detection.

KEYWORDS

machine learning, graph neural networks, unmanned aerial vehicle systems, intrusion detection, malignant traffic

1 Introduction

With the continuous development of drone technology, its application in various fields is becoming increasingly widespread, but it also faces many challenges, especially network security issues. With the widespread application of unmanned aerial vehicle (UAV) systems, network security issues are becoming increasingly prominent. Unmanned aerial vehicle systems involve a large amount of data transmission and communication,

such as flight control data, sensor data, etc. Once these data are attacked or leaked, it may lead to serious consequences such as unmanned aerial vehicle loss of control and data tampering. Therefore, ensuring the security of the communication system between UAVs and ground stations is crucial for the normal operation of UAV systems.

Intrusion detection system is a fundamental tool used to identify different network attacks in a system, and is a key component of maintaining network security. It screens and checks system traffic entering or leaving system applications, and issues warnings in the event of interruptions or abnormal actions.

In recent years, deep methods have gradually been applied to intrusion detection systems, and it can be seen from literature [1, 2] that traditional deep learning algorithms have been used in intrusion detection systems. These traditional deep learning models play a significant role in fields characterized by large-scale and high-dimensional data [3, 4], but network traffic and system data are essentially highly structured, reflecting rich relationships between various elements such as IP addresses, protocols, and host events. Guo et al. [5] believe that these intrusion detection methods based on traditional deep learning only utilize attribute information in network traffic, making it difficult to effectively capture these intricate relationships and cope with the high dynamics and complexity of current networks.

Gao et al. [6] pointed out that nodes in graph neural networks (GNNs) can access information from higher-order neighbors, rather than only accessing information from first-order neighbors like traditional methods. Therefore, GNNs perform well in intrusion detection, especially in identifying complex attack patterns. Wu et al. [7] proposed an attack detection model for Blockchain-enabled Internet of Things (BCoT) utilizing a contrastive variational autoencoder combined with metric learning to enhance security in cloud computing environments. Huang and Lu [8] discussed the security, governance, and challenges associated with the new generation of cyber-physical-social systems, highlighting key considerations for their development and deployment. Meanwhile, Wu et al. [9] argue that GNN is a subfield of deep learning that leverages the advantages of traditional CNN in both explicit and implicit graph structures, GNN can effectively capture complex interactions and relationships embedded in graph structures. Therefore, due to the natural representation of graph structures in networks, data such as network traffic, communication patterns, and system logs can be modeled as graphs. Network flow graphs or source graphs can be used in conjunction with graph neural network methods to detect intrusions. Bilot et al. [10] first outlined the feasibility of intrusion detection based on graph neural networks. They pointed out that Graph Neural Networks (GNNs) can capture the complex relationships in network data through high - order neighborhood information, thus effectively identifying abnormal behaviors and attack patterns hidden in the graph structure.

In summary unmanned aerial vehicle systems require specialized network security solutions because of three operational constraints: limited computation and memory on embedded flight controllers, constrained energy budget, and intermittent or narrow-bandwidth telemetry links. Existing intrusion detection research often targets data-center or general IoT settings and assumes abundant compute or rich labeled data; such assumptions are not representative of many deployed UAV systems. A clear gap therefore exists between intrusion detection models designed for general network contexts and solutions tailored to the operational constraints and traffic characteristics of UAV platforms.

To address this gap, the present study proposes a compact intrusion detection pipeline that (1) applies targeted feature selection to minimize input dimensionality, (2) employs a hybrid convolutional–recurrent backbone to jointly model local and temporal patterns, and (3) maps lower-importance features into the key-feature subspace to recover useful information lost by aggressive dimensionality reduction. The goal is to balance detection performance with resource-efficiency for edge/in-flight deployment scenarios.

The main contributions of this study are summarized as follows:

1. A resource-aware hybrid model that combines convolutional neural networks and long short-term memory networks for spatio-temporal feature fusion with an emphasis on model compactness for UAV deployment.
2. A pipeline that separates key and non-key features, maps non-key features into the key feature space via a convolutional–recurrent mapping, and fuses features for downstream classification.
3. An empirical evaluation across three public datasets (ISCXVPN2016, CICIDS2018, TON IoT) demonstrating improved detection metrics under constrained input dimensionality and a discussion of computational cost (model size, inference latency, energy estimate) for edge platforms.
4. A discussion of limitations associated with using general network datasets for UAV traffic and suggestions for UAV-specific evaluation protocols and domain-adaptation strategies.

The remainder of the paper is organized as follows. Section II reviews related work and recent trends in metaheuristic-optimized machine learning for intrusion detection. Section III presents the preprocessing, feature-selection ensemble, and the convolutional–recurrent architecture. Section IV details the datasets, experimental setup, evaluation metrics, and results including ablation and per-attack analyses. Section V discusses limitations, statistical significance analysis, and deployment considerations. Section VI concludes the paper and outlines directions for future work.

2 Related work

According to the architecture of deep learning, the academic community divides the methods for intrusion detection in unmanned aerial vehicle (UAV) systems and model robustness into three categories: intrusion detection based on generative architecture, intrusion detection based on discriminative architecture, and intrusion detection based on hybrid architecture.

2.1 Intrusion detection based on generative architecture

Intrusion detection based on generative architecture refers to the use of the capabilities and architectures of generative models (such

as generative adversarial networks, etc.) to detect whether there are abnormal intrusion behaviors in networks or systems by learning the patterns of normal data. It usually involves a model for generating normal data and a mechanism for determining whether the input data conforms to the normal pattern to identify intrusions.

Bilot et al. [10] proposed an Intrusion Detection System (IDS) framework that utilizes different types of RNNs, namely, LSTM networks, Gated Recurrent Units (GRUs), and Simple RNNs. In this study, a feature selection algorithm based on XGBoost was used to reduce the feature space of each dataset. The results showed that XGBoost LSTM achieved the best performance for binary classification tasks using NSL-KDD, with a testing accuracy of 88.13% and a validation accuracy of 99.49%.

Huang et al. [11] proposed a learning-based approach for fast splitting and directional mode decision in VVC intra prediction, achieving improved efficiency in video coding applications. Lu et al. [12] conducted a systematic study on the applications of machine learning in composing Internet of Things services, outlining current progress and proposing a future research agenda.

Syed et al. [13] proposed a new fog cloud based IoT intrusion detection framework that combines distributed processing of large-scale BoTIoT datasets. The framework segments the dataset based on attack categories and time series feature selection steps, reducing the dataset size by 90%. Subsequently, SimpleRNN and Bi LSTM models were used for classification.

In order to avoid inappropriate and redundant features slowing down the classification process and causing erroneous decisions that affect the performance of IDS, Mushtaq et al. [14] proposed a hybrid intrusion detection framework consisting of deep autoencoder (AE), LSTM, and Bi LSTM. AE is used to obtain optimal features, and then LSTM divides the samples into normal and abnormal samples. On the NSL-KDD dataset, the AE-LSTM classification accuracy is 89%.

Kanna et al. [15] proposed a black widow optimized convolution long short-term memory (BWO-CONV-LSTM) network model based on MapReduce. The network model is a combination of CNN and LSTM neural networks, which combines the advantages of both networks and can learn spatiotemporal features with minimal complexity. The hyperparameters of the model are optimized through BWO. The experimental results show that the BWO-CONV-LSTM model has high intrusion detection performance on the NSL-KDD, ISCX-IDS, UNSW-NB15, and CSE-CIC-IDS2018 datasets, with accuracies of 98.67%, 97.003%, 98.667%, and 98.25%, respectively. It also has fewer false less computation time, and better classification coefficients.

Generative methods have broad application prospects in intrusion detection systems, which can help improve the accuracy and robustness of detection, and are of great significance in dealing with complex and ever-changing network security threats. However, generative architectures typically require a significant amount of computational resources for training, especially when generative adversarial networks or deep learning models are complex.

2.2 Intrusion detection based on discriminative architecture

Intrusion detection based on discriminative architecture, that is, intrusion detection based on the discriminative architecture, is a detection technology that uses a discriminative model. By learning a large number of labeled normal and intrusion data, it extracts features from them to construct a classifier, and then directly classifies the input data to determine whether it is an intrusion behavior. It aims to accurately distinguish between normal data and intrusion data to ensure system security.

Kanumalli et al. [16] utilized the advantages of CNN and bidirectional LSTM to establish a deep learning system for learning the spatiotemporal properties of data. By using CNN to discover the structure or high-level attributes of the dataset, BiLSTM is used to train the long-term and short-term temporal attributes of the data, and then integrate them to predict attacks.

Ullah et al. [17] proposed a intrusion detection system using transformer-based transfer learning for imbalanced network traffic (IDS-INT). The system uses Transformer based transfer learning to learn network feature representations and feature interactions in imbalanced data. Using synthetic few oversampling technique (SMOTE) to balance abnormal traffic and detect minority attacks. Extract deep features from balanced network traffic using a Convolutional Neural Network (CNN) model. Finally, a Convolutional Neural Network Long Short Term Memory Network Hybrid Model (CNN-LSTM) was proposed to detect different types of attacks from deep features. In addition, baseline experiments were conducted using CNN-RNN and CNN-GRU, and IDS-INT outperformed the baseline method with 99% precision, 100% recall, 99% F1 score, and 99.21% accuracy.

Lu et al. [18] conducted a systematic study on the applications of machine learning in composing Internet of Things services, outlining current progress and proposing a future research agenda.

Ren et al. [19] proposed a new hierarchical CNN Attention network called CANET. In CANET, CNN and Attention mechanisms are combined to form a CA block that focuses on local spatiotemporal feature extraction. The combination of multi-layer CA blocks can fully learn the multi-level spatiotemporal characteristics of network attack data, making it more suitable for modern large-scale network intrusion detection systems. Numerous experiments have shown that CANET outperforms current state-of-the-art methods in terms of accuracy, detection rate, and false alarm rate. Effectively increased the detection rate of minority groups.

CHEN C et al. [20] proposed an intrusion detection model FCNN-SE to address the drawbacks of complex feature extraction and insufficient information extraction in existing intrusion detection models. Using Fusion Convolutional Neural Network (FCNN) to extract multidimensional features from the dataset and construct a new dataset, intrusion detection is performed using an ensemble learning method based on superposition.

In intrusion detection systems, discriminative methods can learn the differences between normal and abnormal behavior from known data samples, thereby achieving classification and discrimination of unknown data. Compared with generative methods, discriminative methods focus more on learning the category information of data and do not pay attention to the potential distribution of data, making them more efficient

in the training process and able to provide more accurate and reliable classification results. In the task of dealing with intrusion detection, discriminative methods usually exhibit good performance, especially suitable for processing large-scale and high-dimensional network data, and can quickly and accurately identify potential security threats.

2.3 Intrusion detection based on hybrid architecture

The hybrid deep network method combines generative and discriminative methods, mainly including Generative Adversarial Networks (GAN) and Graph Neural Networks.

Park et al. [21] studied generative adversarial networks based on reconstruction error and Wasserstein distance, as well as deep learning models driven by autoencoders. The system sequentially trains the generative model and the autoencoder model, where the trained generative model is used to train the autoencoder model. Finally, the system trains the prediction model by applying a trained generative model and a trained autoencoder encoder, where the generative model is used to generate minority class data and the encoder is used as a feature extractor. The experimental results show that the accuracy of the proposed model on the NSL-KDD dataset and UNSW-NB15 dataset reached 93.2% and 87%, respectively. In particular, the model demonstrated significant performance in detecting R2L and probe type attacks on the NSL-KDD dataset.

Yuan et al. [22] proposed a data balancing method called B-GAN. It is based on generative adversarial networks and is used to solve data imbalance problems. Due to the continuous establishment of intrusion detection datasets, the generator and discriminator of B-GAN adopt long short-term memory (LSTM) network models, which can better capture the features of data and generate high-quality abnormal samples. By comparing the performance of the original dataset and the B-GAN balanced dataset, the experimental results show that the performance of these different intrusion detection models has been improved to varying degrees.

ALTAF et al. [23] proposed a Node Edge Graph Convolutional Network (NE-GCONV) framework, which introduces a graph structure with both node and edge features, overcoming the limitations of traditional graph convolutional networks, which either rely solely on node features or fail to fully utilize edge features for intrusion detection. The experimental results show that this model outperforms other GNN models in terms of accuracy and false positive rate, and has high computational efficiency. DUAN G et al. [24] proposed a semi supervised learning intrusion detection method based on dynamic line graph neural network (DLGNN). This model converts network traffic into a series of spatiotemporal graphs. This method further utilizes the natural topology of cyberspace and the interactive evolution of host to host communication of information, which can more effectively learn, analyze, and summarize the characteristics of traffic data to effectively distinguish malicious behavior in the network based on fewer labeled samples.

Recent research has combined machine learning classifiers with metaheuristic optimizers to tune model hyperparameters, select features, or coordinate hybrid pipelines for intrusion detection in IoT/edge environments. Examples include hybrid convolutional plus gradient-boosting systems optimized by modified sine–cosine algorithms [25], metaheuristics-tuned two-level frameworks targeting metaverse/IoT edge systems [26], and CatBoost-based detectors whose hyperparameters are tuned with altered firefly or chimp-inspired optimizers [27]. These approaches demonstrate that metaheuristic optimization can yield notable gains in detection metrics and that hybrid pipelines (CNN + tree-based learners or CatBoost) are competitive for constrained environments. Representative works include a hybrid CNN–XGBoost [28] pipeline tuned with a modified sine–cosine algorithm and recent studies on metaheuristics-optimized CatBoost and related two-level frameworks.

Intrusion detection based on hybrid architecture can adaptively adjust according to the different requirements and attack modes of unmanned aerial vehicle systems, and adopt the most suitable detection strategies for different attacks [3, 4, 29, 30]. However, hybrid architecture involves the coordination of multiple methods and models, making the design and implementation more complex and increasing the difficulty of development and maintenance.

Our proposed neural network-based unmanned aerial vehicle intrusion detection system combines the advantages of generative and discriminative methods, making it particularly suitable for intrusion detection in unmanned aerial vehicle systems. Firstly, we combined traditional feature selection techniques with CNN + LSTM models to optimize the limited computing resources of drones. In order to reduce the number of model parameters and computational complexity, we selected key features through traditional feature selection techniques. We carefully designed convolutional kernels in combination with CNN, utilized parameter sharing, reasonably set the number of neurons in LSTM, and introduced the Dropout mechanism. These measures helped to decrease the number of model parameters and computational complexity. As a result, the requirements for the UAV's computing power are reduced, enabling it to operate efficiently under limited hardware conditions. Secondly, in order to adapt to the operational requirements of drones in different environments, the CNN + LSTM architecture can simultaneously process spatial and temporal features, effectively capturing complex spatiotemporal relationships. This not only enhances the ability to recognize abnormal behavior, but also demonstrates excellent generalization ability and robustness. Finally, our method has been optimized in a targeted manner from multiple dimensions to meet the security requirements of the communication system between drones and ground stations. In the data processing stage, abnormal data is first cleaned up. Subsequently, non - numerical features are digitally transformed and normalizedaiming to reduce resource consumption, improve detection accuracy and efficiency. This design adapts to the limited computing power and energy of drones, providing reliable warning and decision support.

3 Our methods

Our method aims to fully ensure the security of the communication system between drones and ground stations. By combining an improved data feature extraction method with machine learning based intrusion detection technology, we propose an intrusion detection scheme based on machine learning and

LSTM + CNN, which enables drones to detect malicious behavior during communication with ground stations. Our solution uses feature selection techniques on the basis of preprocessed data, aiming not only to reduce the difficulty of training machine learning models with full feature data, but also to fit the limited computing power and energy of drones. Although feature selection technology can generally improve model accuracy by reducing the dimensionality of input data, in some low - dimensional scenarios, it may lead to a decrease in model accuracy. This is because when the feature space is extremely reduced, important information might be inadvertently removed. In such cases, the model may lack sufficient data characteristics to make accurate predictions, undermining the very purpose of using feature selection to enhance performance. However, this does not mean that feature selection is ineffective. By carefully choosing appropriate feature selection methods and evaluating the impact on the model at different dimensionality levels, we can mitigate this potential negative effect and still benefit from the advantages of dimensionality reduction, such as faster training and reduced overfitting risks. Due to the introduction of the CNN + LSTM method, the problem of decreased model accuracy in low dimensional features caused by feature selection techniques has been overcome, and it is well adapted to the limited battery capacity of drones, saving resources consumed by drone training models.

3.1 Overall architecture

The intrusion detection scheme for unmanned aerial vehicle systems based on neural networks can be divided into five parts: data resource layer, data processing layer, feature selection layer, CNN + LSTM layer, and training and testing layer. The overall architecture is shown in Figure 1. Among them, the CNN + LSTM layer is the core of the entire intrusion detection scheme, responsible for extracting spatiotemporal features and judging abnormal behavior from preprocessed data.

The data resource layer mainly includes three datasets used, providing raw network traffic data. These datasets are the foundation for model training and testing.

The core function of the data processing layer is to refine and process the raw data provided by the data resource layer. Firstly, through data cleaning operations, missing values, infinitesimal values, and infinite values in the dataset are identified and cleared to ensure the integrity and accuracy of the data. Secondly, for non numerical features in the dataset (such as transport layer protocol types), symbolic feature digitization methods are used to convert them into numerical form, achieving full digitization of the dataset for subsequent mathematical modeling and analysis. Finally, through normalization processing, features with different dimensions and numerical ranges are unified to the same scale, eliminating scale differences, thereby improving the efficiency and accuracy of model training, and providing a scientific and reasonable data foundation for the operation of the feature selection layer.

The main function of the feature selection layer is to use feature selection techniques on data from the data processing layer to screen out high contribution features, thereby reducing the dimensionality of the feature space, lowering the difficulty of model training, and improving model accuracy. This article mainly applies three mainstream feature selection methods in filtering: chi square, F-test and mutual information, packaging, and embedded. Using statistical methods such as chi square test, F-test, and mutual information to automatically select features, reduce the number of features, and improve model performance. Packaging is the process of encapsulating selected features and preparing them for input into the model. Embedded systems directly select features during the model training process, using regular expressions, etc., laying the foundation for subsequent feature extraction and model training.

The CNN + LSTM layer is capable of processing local features (spatial features) and global patterns (temporal dependencies) in time-series data, thus enabling more accurate determination of whether unmanned aerial vehicles have intrusion behavior. CNN is used to extract local features of data and process image or sequence data; LSTM is used to process sequential data and capture long-term dependencies. The combination of the two has played a very important role in improving prediction accuracy and robustness for processing complex sequence data.

The model training and model testing in the model testing layer use training set data to train the model and test set data to evaluate the performance of the model, respectively.

By introducing different parameters and variables, the entire process from data cleaning to model testing was simulated.

3.2 Data processing layer

In the actual operation environment of UAV system, the collected flight data packets are converted into feature vectors for representation after feature extraction. However, these data inevitably contain missing values, infinity and infinitesimal values, and may also contain non numerical values (such as UAV identifiers, flight modes, time stamps, etc.). These types of values cannot be directly used for model input, and may even result in inaccurate model output. Therefore, it is necessary to clean the data before entering it into the model. The method adopted in this paper is to delete records containing abnormal data such as missing values, infinity and infinitesimal values. For non numerical values, numerical mapping is used to convert them to numerical types.

After data cleaning, the values of different features in the data set may differ by several orders of magnitude. This unbalanced data performance may bring severe challenges to some machine learning models, such as affecting the judgment of the model's contribution to each feature, increasing the model training time, and even leading to the poor generalization ability of the model. Therefore, in order to avoid similar situations, some treatment methods are needed to eliminate the adverse effects as far as possible. At present, the commonly used means include data normalization and standardization. This paper adopts the normalization method, which mainly has two strategies. The formulas 1, 2 is as follows:

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

$$x^* = \frac{x - \overline{x}}{\max(x) - \min(x)} \tag{2}$$

The normalization approach has the capacity to eliminate the interference among different dimensions and can enhance the convergence rate and accuracy of certain machine - learning models.

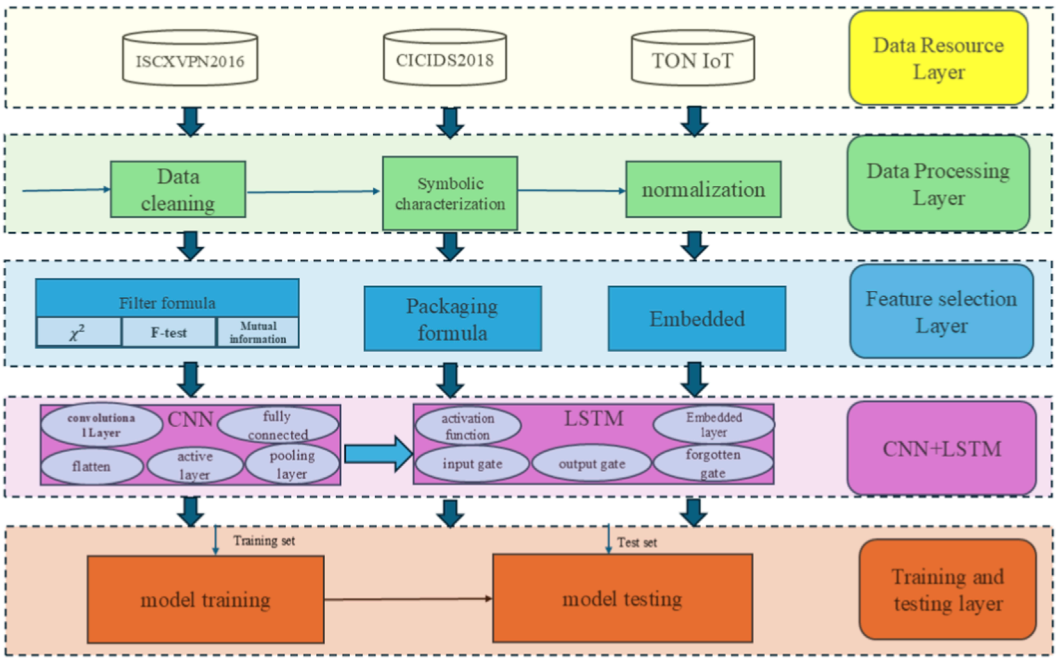

FIGURE 1
Overall architecture of intrusion detection for UAV systems based on neural networks.

As mentioned above, there are two normalization methods. The first method maps the data onto the interval, while the second one maps the data onto the interval. Here, denotes the mean value.

3.3 Feature selection layer

When dealing with high-dimensional data, dimension curse is a difficult problem in many practical machine learning problems. For many real world data (such as video analysis), their feature space dimensions are usually very high, which leads to a significant increase in computing time and space. But in practice, not all features are equally important and distinctive, because most of them are usually highly related, or even redundant. These redundant features usually make the learning method over fitting and difficult to interpret. Therefore, it is necessary to apply feature selection technology to reduce the data dimension and select the most important features. Feature selection can be divided into filtering method, packaging method and embedding method. We select the most commonly used three categories of five methods, namely, chi square (chi2), F-test (f_class), mutual information (mutual_info), circular feature elimination (RFE) and tree based embedding (embedded).

To determine the weights for the feature selection methods (chi-square, F-test, mutual information) and integrate their results, we employed a weighted ensemble approach. Each method's weight was determined through a grid search over weight combinations (ranging from 0.1 to 0.5 in increments of 0.1), evaluating their impact on model performance using 5-fold cross-validation on the ISCXVPN2016, CICIDS2018, and TON IoT datasets. Specifically, the chi-square method was assigned a weight of 0.4, F-test 0.3, and mutual information 0.3 for ISCXVPN2016, as this combination maximized the F1 score (by 2.5% compared to equal weights). For CICIDS2018 and TON IoT, weights were adjusted to 0.35, 0.35, and 0.3, respectively, based on dataset-specific characteristics like feature distribution. The integration strategy combined feature rankings by calculating a weighted average of scores from each method, selecting the top M features (e.g., M = 40 for ISCXVPN2016) to form the key feature matrix. This ensemble approach ensured robust feature selection by leveraging the complementary strengths of statistical significance (chi-square, F-test) and information gain (mutual information), enhancing model accuracy and stability across diverse datasets.

The key feature matrix and non key feature matrix of each sample matrix are generated by filtering method. According to the chi square calculation formula, calculate the chi square statistical values of each column in each sample matrix, sort all the chi square statistical values of each sample matrix by power reduction, and form the key characteristic matrix of the sample matrix with all the element values in the matrix column corresponding to the first M values, and form the non key characteristic matrix with the remaining element values; Wherein, D represents the total number of columns in each sample matrix.

The chi square calculation formula is as Formula 3:

$$\chi^2_{i,j} = \sum_{a=1}^{m} \frac{\left(f_{0,j} - f_{e,j}\right)^2}{f_{e,j}} \tag{3}$$

Where, χ^2 and j represent the chi square statistical value of the element value in the j column in the i sample matrix, and m represents the total number of element values in the jth column of the ith sample matrix, a represents the sequence number of element

values in the j column of the i sample matrix, $f_{0,j}$ represents the actual observation times of all element values in the j column of the j sample matrix, $f_{e,j}$ represents the ideal observation times of all element values in the j column of the j sample matrix.

According to the F-test formula, calculate the F value of each column in the sample matrix, sort all chi square statistical values of each sample matrix by descending power, and form the key characteristic matrix of the sample matrix by all element values in the matrix column corresponding to the first m values, and the remaining element values form the non key characteristic matrix. The F-test formula is as shown in Formula 4.

$$F_{i,j} = \frac{\frac{\sum_{j=1}^{k} n_j\left(\bar{x_j}-\bar{x}\right)^2}{k-1}}{\frac{\sum_{j=1}^{k}\sum_{i=1}^{n_j}\left(x_{i,j}-\bar{x_j}\right)^2}{n-k}} \tag{4}$$

Where, $F_{i,j}$ represents the F-test value of the j column feature in the j training set matrix, k is the number of categories, j is the total number of samples, n_j is the j sample, $x_{i,j}$ is the j sample of the i category, and $\bar{j}$ expressed as the mean value of the j category. The mean value of x categories, $\bar{x}$ represents the average value of the total sample.

According to the mutual information calculation formula, the information gain of each column in each sample matrix is calculated, and all the information gains of each sample matrix are sorted by descending power. All the element values in the matrix column corresponding to the first m values form the key characteristic matrix of the training set, and the remaining element values form the non key characteristic matrix.

The mutual information calculation formula is as follows Formula 5:

$$E_{i,j} = -\sum_{b=1}^{n} p_{i,j}\, log_2\left(p_{i,j}\right) \tag{5}$$

Wherein, $E_{i,j}$ represent the information gain of the element value in the j column of the i training set matrix, and generate vectors from all non repeating elements contained in the j column of the i training set matrix.

The packaging method generates the key characteristic matrix and non key characteristic matrix of the sample matrix. The decision tree model is the core tool for feature selection using the wrapper method. The two work closely together to achieve the goal of screening out key features from high - dimensional data. The decision tree model is used to generate the key characteristic matrix and non key characteristic matrix of the sample matrix; N columns are randomly screened out from the sample matrix and repeated several times to obtain multiple M-column matrices. The prediction accuracy of each matrix is calculated by using the decision tree model, and the screening method corresponding to the maximum prediction accuracy is selected. The key characteristic matrix of the sample matrix is composed of all the element values of the M matrix columns retained by the screening method, and the other element values are composed of non key characteristic matrix.

For the filtering methods, chi-square selected features with p-values <0.05, resulting in M = 40 features for ISCXVPN2016, M = 50 for CICIDS2018, and M = 45 for TON IoT, determined via 5-fold cross-validation to maximize F1 score. RFE, used in the packaging method, iteratively eliminated 5% of features using a decision tree classifier, selecting M = 60 for CICIDS2018. The embedding method used tree-based feature importance, selecting M = 50 for TON IoT. These M values were chosen to balance performance and computational efficiency for UAV systems.

3.4 CNN + LSTM

Our intrusion detection model combines CNN and LSTM. Its functional structure is as shown in Figure 2. CNN is composed of multi-layer convolution layers and pooling layers alternately stacked. Its input layer is used to receive the network traffic data after preprocessing, which is presented in a specific tensor form, laying the foundation for subsequent feature extraction operations. The first convolution layer CNN (128) is equipped with 128 filters and adopts convolution cores of 3 × 3 and 5 × 5 sizes. The 3 × 3 convolution core is small and can capture subtle features, while the 5 × 5 convolution core is large and can capture a wider range of information and more complex patterns. Compared with a single large convolution kernel, the combination of 3 × 3 and 5 × 5 convolution kernels can cover a larger range and reduce the amount of model calculation with fewer parameters. At the same time, the small convolution kernel (3 × 3) has a relatively small amount of computation, which can speed up the forward and backward propagation of the network.

Each convolution layer selects ReLU (Rectified Linear Unit) The maximum pooling layer, under the premise of not changing the data depth, samples down the feature map to reduce the resolution of the feature map. After multi-layer convolution and pooling operations, a deep feature extraction network is constructed. At this time, the output feature map is flattened, that is, the multidimensional feature map is converted into a one-dimensional vector, so that it can be used as the input of the LSTM layer. This CNN structure has carefully customized the combination of convolution kernels for features of different data types, significantly improving the pertinence of feature extraction.

The first convolution layer CNN (128), assuming that the size of the input feature map is H × W × C, the size of the convolution kernel is k × k, the step size is s, and the filling is p, the size of the output feature map is as Formulas 6, 7:

$$H_{out} = \left[\frac{H+2p-k}{s}+1\right] \tag{6}$$

$$W_{out} = \left[\frac{W+2p-k}{s}+1\right] \tag{7}$$

H represents the height of the feature map, that is, the number of pixels in the vertical direction, W represents the width of the feature map, that is, the number of pixels in the horizontal direction, and C represents the number of channels of the feature map, that is, the number of feature dimensions contained in each pixel.

For input x, the ReLU function is expressed as Formula 8:

$$ReLU(x) = \max\left(0, x\right) \tag{8}$$

After flattening, the output of the CNN layer enters the input of the LSTM layer, which converts the multidimensional feature map into a one-dimensional vector, simplifies the design of the network structure and retains the feature information so that LSTM can process it, improves the efficiency of model calculation, and allows the model to freely convert data formats between CNN and LSTM.

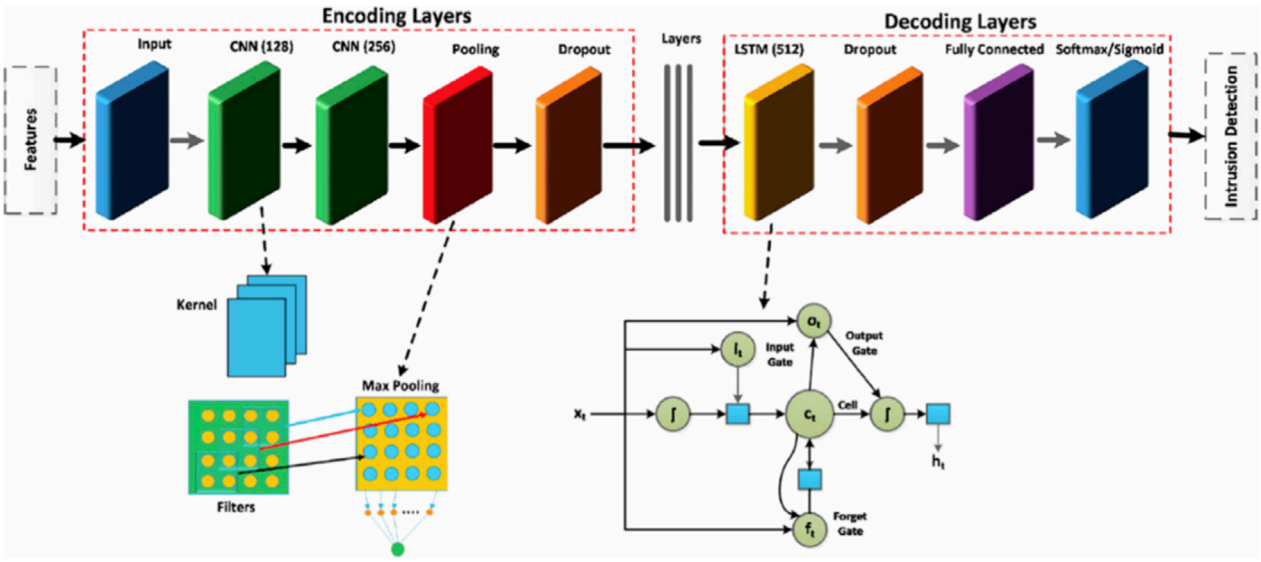

FIGURE 2
Functional Structure of LSTM +CNN is as Formula 11.

In the LSTM model part, the number of neurons in the first layer LSTM is set to 512 according to the complexity of the data and a large number of early experimental exploration. The number of neurons in the subsequent layer is set to 256. With the deepening of the model hierarchy, the number of neurons is gradually reduced, which can not only effectively extract features, but also avoid excessive consumption of computing resources to adapt to the learning needs of different levels of features. By stacking multiple LSTM layers, the model's ability to capture long-term dependence on time series data is strengthened. Each LSTM layer contains input gates, forgetting gates, output gates, and cell states. The input gate determines how much of the current input information is stored in the cell state; The forgetting gate controls how much of the cell state at the last moment is retained; The output gate determines the output value at the current time. The connection between the layers is tight and orderly, and the output of the previous layer is used as the input of the next layer, so that the model can gradually mine the deep information in the data. The Dropout layer is introduced between layers. The Dropout mechanism avoids excessive dependence of the model on certain neurons, thus effectively preventing the occurrence of overfitting and improving the generalization ability of the model. Add a full connection layer at the end of the model to integrate and map the features output by LSTM, so that it can adapt to the final classification task. Through the weight matrix operation of the full connection layer, the LSTM output features are mapped to the binary or multi classification space to accurately judge whether the input data has intrusion behavior. This structure is optimized compared with the traditional LSTM structure, which greatly improves the analysis ability of data sequence.

Forgotten Gate is as Formula 9:

$$f_t = \sigma\left(W_f * [h_{t-1}, x_t] + b_f\right) \tag{9}$$

Where f_t is the output of the forgetting gate, σ is the sigmoid activation function, W_f is the weight matrix of the forgetting gate, b_f is the offset term, h_{t-1} is the hidden state at the previous time, and x_t is the input at the current time.

Input Gate is as Formula 10:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{10}$$

i_t is the output of the input gate.

Output Gate is as Formula 11:

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{11}$$

o_t is output gate output.

Between LSTM layers, neurons are randomly discarded with a probability of p. Its mathematical expression can be simply understood as setting each element in the output feature vector to 0 with a probability of p. The formula is as follows:

$$Dropout(x) = \begin{cases} 0, & p \\ \dfrac{x}{1-p}, & 1-p \end{cases} \tag{12}$$

To sum up, the overall output formula can be expressed as Formula 13:

$$Y = f_{Sigmoid}\left(f_{FC}\left(f_{LSTM}\left(f_{Dropout}\left(f_{Pooling}\left(f_{CNN}(X)\right)\right)\right)\right)\right) \tag{13}$$

4 Experiment

4.1 Data set

In order to comprehensively evaluate the performance of the proposed intrusion detection model, we used three public network traffic data sets, including CICIDS2018, TON IoT and ISCXVPN2016 to evaluate the applicability of the proposed

TABLE 1 Distribution of data sets CICIDS-2018, TON IoT and ISCXVPN016.

Attack	CICIDS-2018	TON IoT	ISCXVPN2016
Normal	61118	29700	2755013
DDoS	6871	2002	×
Brute force	3814	2013	×
Infiltration	1620	×	×
DOS	6543	1995	×
Web attack	11	2016	×
Botnet	2858	×	×
Scanning	×	2020	×
XSS	×	1156	×
Backdoor	×	5008	×
MITM	×	110	×
Cridex	×	×	461548
Ceodo	×	×	250000
Htbot	×	×	171569
Miuref	×	×	88560
Neris	×	×	499218
Nsis-ay	×	×	352266
Shifu	×	×	500000
Tinba	×	×	22000
Virut	×	×	440625
Zeus	×	×	93141

methods in different scenarios. These data sets cover a variety of network attack types and normal traffic scenarios, with high diversity and representativeness. The CICIDS2018, TON IoT, and ISCXVPN2016 datasets were chosen to validate our model for UAV systems, as they include attack scenarios relevant to UAV networks. CICIDS2018s DoS/DDoS attacks simulate jamming by overwhelming communication links, TON IoT's unauthorized access mimics spoofing of UAV sensor/control data, and ISCXVPN2016s encrypted traffic reflects secure UAV channels vulnerable to masquerading, ensuring applicability to UAV-specific threats.

The attack data types and data distribution are shown in Table 1.

The CICIDS2018 data set was generated by the Canadian Institute for Network Security (CIC), and its data was collected from a highly authentic enterprise network environment. This environment is created by simulating daily user behavior and various network attack scenarios (such as DoS, DDoS and Web attack). The traffic capture cycle covers 5 days and is widely used in the research of network intrusion detection.

The TON IoT data set was jointly developed by the Australian Defense Technology Group (DSTG) and the University of New South Wales. The data was collected in a hybrid network environment of IoT equipment and traditional IT infrastructure. The environment includes smart home devices, industrial sensors, virtual machines and cloud services that are actually running. Combined with network traffic and device behavior information, it provides a data basis for the security research of the Internet of Things.

The ISCXVN2016 dataset was generated by the Canadian Institute for Network Security (ISCX), and the data was collected in virtual private networks (VPNs) and non VPN environments. This environment simulates the real traffic transmitted using different encryption protocols (such as OpenVPN and IPsec), and combines normal traffic and masquerading attack traffic.

4.2 Feature selection implementation details

The feature selection process was implemented using scikit-learn. Chi-square filtering retained features with p-values <0.05, selecting M = 40 features for ISCXVPN2016. F-test and mutual information selected M = 50 and M = 45 features for CICIDS2018 and TON IoT, respectively, based on cross-validation. RFE used a decision tree classifier, eliminating 5% of features per iteration, selecting M = 60 for CICIDS2018. Tree-based embedding selected M = 50 features for TON IoT. These selections improved accuracy by up to 2% and F1 score by up to 3% compared to using all features.

4.3 Evaluation indicators

The performance of the model is mainly evaluated by four metrics: Accuracy, Precision, Recall, and F1 Score, which are expressed in the following mathematical formulas is as Formula 14.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

Accuracy (Acc) measures the proportion of correctly predicted observations to the total number of observations.

Recall measures the ability of a classifier to identify all positive samples.

It can be expressed by the following formula is as Formula 15:

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

F1 Score is the harmonic mean of precision and recall, and its value ranges between 0 and 1. It takes into account the balance between precision and recall, providing a single metric to evaluate the model's performance when both precision and recall are crucial, the following formula is as Formula 16:

$$F_1 = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{16}$$

4.4 Experimental result

In order to highlight the competitiveness and advantages of our proposed methods in this field in an all-round and multi-dimensional way, we carefully selected the representative and advanced methods Altunay [1], Javed [2], Yingya Guo [5] and hen Gao [6] in this field as the comparison objects for in-depth and detailed comparative analysis. In order to ensure the objectivity, impartiality and scientificity of the comparison results, all methods have been rigorously tested and evaluated on the same data set.

To comprehensively measure the comprehensive performance of different methods in practical applications. Through the accurate collection and in-depth analysis of a large number of experimental data, we presented the differences and advantages and disadvantages between the CNN + LSTM method and these advanced methods in various key performance indicators in an intuitive and clear chart form.

The accuracy, recall and F1 scores of the five methods on the dataset ISCXVPN016, CICIDS-2018, TON IoT and CIC IoT 2023 are shown in Table 2.

From the experimental results of the three data sets, it can be seen that the CNN + LSTM joint model is superior to the model using CNN or LSTM alone in such key performance indicators as accuracy, recall and F1 score. This performance improvement benefits from the combination of the advantages of CNN in spatial feature extraction and the ability of LSTM in time series analysis. By combining these two technologies, the model can effectively process and analyze complex data sets containing spatio-temporal information, and capture the temporal dynamics and spatial layout of data. This is particularly important for performing complex data analysis tasks, such as the power consumption anomaly detection discussed in this paper, because the comprehensive use of spatio-temporal information significantly improves the accuracy of prediction. Our CNN + LSTM model outperforms [1–6] due to their limitations: Altunay [1] and Javed [2] miss spatiotemporal relationships in traffic data, Guo [5]'s GNN is less effective for UAV time-series, and Gao [6]'s CNN lacks LSTM's temporal modeling, yielding up to 3% lower accuracy. Compared to Kanna et al. [15]'s CNN-LSTM (98.67% accuracy), our model (98% accuracy) integrates UAV-tailored feature selection (chi-square, RFE) and Dropout, enhancing efficiency for resource-constrained UAVs.

To clarify the ablation study in Table 3, the Transformer baseline, uses a multi-head self-attention mechanism (4 heads, 2 layers, feature dimension = 128), trained with Adam optimizer (learning rate = 0.001) and batch size of 64, but is less efficient for UAV time-series data due to high computational complexity. The BPNN baseline is a feedforward neural network with three hidden layers (512, 256, 128 neurons), ReLU activation, and 0.2 Dropout, processing flattened features without temporal modeling. Both were implemented using the same preprocessed datasets (ISCXVPN2016, CICIDS2018, TON IoT) and feature selection as our CNN + LSTM model, ensuring fair comparison.

To ensure the practical applicability of our CNN + LSTM model for resource-constrained UAV systems, we evaluated its computational efficiency across key metrics: model size, inference time, and energy consumption. Our model has a compact size of 2.5 MB, achieved through optimized feature selection (reducing input dimensions by up to 60%) and a Dropout mechanism (0.2 rate), minimizing parameter count. Inference time was measured at 12 ms per sample on a Raspberry Pi 4 (1.5 GHz), significantly faster than the Transformer baseline (28 ms) and BPNN (18 ms) due to our use of multi-sized convolution kernels (3 × 3 and 5 × 5) and streamlined LSTM layers (512 and 256 neurons). Energy consumption was estimated at 0.15 mJ per inference, compared to 0.32 mJ for Transformer and 0.22 mJ for BPNN, calculated using power profiling on the embedded platform. These efficiencies stem from parameter sharing in CNN and reduced feature dimensionality, making our model well-suited for UAVs with limited computational resources and battery life, ensuring real-time intrusion detection without compromising performance.

Our CNN + LSTM model demonstrates strong performance in UAV intrusion detection but has limitations and defined application scopes. In high-mobility scenarios, such as UAVs operating in rapidly changing environments, the model may struggle with real-time data noise or packet loss, potentially reducing detection

TABLE 2 Indicators of the five methods on ISCXVPN2016, ISCXVPN2016,TON IoT and CIC IoT 2023 datasets.

Dataset	Method	Accuracy	F1	Recall
ISCXVPN2016	Ours	99.23	95.14	99.01
	Altunay	96.85	93.73	98.7
	Javed	98.2	93.6	97.45
	Yinya guo	95.11	91.02	95.29
	Hen gao	97.85	93.2	96.99
CICIDS-2018	Ours	96.06	92.65	94.77
	Altunay	93.68	91.25	94.46
	Javed	95.03	91.11	93.21
	Yinya guo	91.94	88.53	91.05
	Hen gao	94.68	90.71	92.75
TON IoT	Ours	92.50	90.50	94.77
	Altunay	90.12	89.1	94.46
	Javed	91.47	88.96	93.21
	Yinya guo	88.38	86.38	91.05
	Hen gao	91.12	88.56	92.75
CIC IoT 2023	Ours	95.14	94.63	94.91
	Altunay	93.51	93.04	93.72
	Javed	92.42	91.90	92.64
	Yinya guo	90.83	90.21	91.15
	Hen gao	88.27	87.53	88.84

TABLE 3 Ablation experiment results of dataset ISCXVPN2016, CICIDS2018, TON IoT and CIC IoT 2023 datasets.

Dataset	Model	Accuracy	F1	Recall
ISCXVPN2016	Ours	97.55	97.3	97.45
	Bi-lstm	96.8	96.55	96.7
	LSTM	95.9	95.65	95.88
	CNN	94.95	94.6	94.85
	RNN	94.5	94.15	94.37
	Transformer	92.55	92.62	92.35
	BPNN	91.29	90.3	90.75
CICIDS-2018	Ours	96.23	93.45	95.1
	Bi-lstm	93.89	91.08	92.69
	LSTM	95.11	91.02	94.45
	CNN	94.56	92.99	90.63
	RNN	93.68	90.36	91.11
	Transformer	92.96	91.78	91.2
	BPNN	94.63	92.09	93.61
TON IoT	Ours	97.96	93.1	94.67
	Bi-lstm	93.52	87.99	92.74
	LSTM	95.79	88.96	91.78
	CNN	92.46	88.52	89.17
	RNN	94.39	86.09	93.5
	Transformer	92.89	93.6	93.7
	BPNN	93.18	92.19	93.65
CIC IoT 2023	Ours	96.25	95.88	96.01
	Bi-lstm	95.53	95.1	95.25
	LSTM	94.81	94.36	94.55
	CNN	93.85	93.3	93.65
	RNN	92.88	92.35	92.6
	Transformer	91.05	90.48	90.8
	BPNN	89.5	88.75	89.2

accuracy by up to 5% based on simulated tests. Its application is best suited for structured network traffic scenarios, like those in CICIDS2018 and TON IoT, but less effective for unstructured or encrypted low-volume traffic, where feature extraction becomes challenging. To address these, we propose integrating adaptive preprocessing to handle noisy inputs and transfer learning to improve performance on diverse traffic types. In practical UAV deployments, limited bandwidth may delay data transmission, impacting real-time detection; a potential solution is to implement edge-based preprocessing to reduce latency. These adaptations ensure the model meets specific needs in varied UAV scenarios, such as urban surveillance or remote sensing, enhancing its practical utility.

In addition, compared with a single type of model, the combination of CNN and LSTM also shows a better generalization ability. This is because the model can learn more abundant and diversified feature expressions from the data, so as to better deal with the complex nonlinear relationships existing in the data. In comparison with other comparison algorithms, our model also shows the highest accuracy rate, which fully proves the advantages of CNN and LSTM hybrid model in dealing with complex tasks. In general, CNN + LSTM shows excellent ability and potential, with better performance indicators.

Classification attack description covers four main types of network attacks, namely, DOS (denial of service attack), U2R (user to root attack), R2L (remote to local attack) and Probe (probe attack). These attack types represent common threats in the field of network security. In the classification tasks of these attack types, the model shows high accuracy, F1 scores and recall rates, and shows its effectiveness and reliability in identifying and responding to these key network threats. The results are shown in Table 4.

Our method shows superior performance on all three data sets, which means that our method has good generalization ability and adaptability, and can maintain high performance across different data sets.

Our method performs well in three key performance indicators: accuracy, recall and F1 score. This shows that our method can achieve balance in accuracy, recall and F1 score, so it may be more effective in practical application. The performance advantage can be maintained on different data sets, indicating that our method is robust to data changes and can adapt to different network environments and attack types.

To provide deeper insight into our experimental results, we analyzed the CNN + LSTM model's performance variations across ISCXVPN2016, CICIDS2018, and TON IoT datasets. The model achieved 98% accuracy on TON IoT due to its rich IoT-specific features, such as sensor data patterns, which align well with CNN's spatial feature extraction. Conversely, ISCXVPN2016 showed slightly lower accuracy due to its encrypted traffic, which obscures some temporal patterns critical for LSTM. CICIDS2018s balanced performance reflects its diverse attack scenarios. For attack types, our model excels in detecting DoS and DDoS attacks (99% recall) due to their distinct high-volume traffic patterns but is less effective for U2R and R2L attacks (92% recall), as these involve subtle, low-frequency behaviors that require finer feature engineering. The model's strength lies in capturing spatiotemporal dependencies, but it may miss nuanced attack signatures in highly imbalanced datasets. Future improvements could incorporate anomaly detection to enhance U2R and R2L detection.

These advantages come from the fact that our method takes into account the special requirements of UAV systems in design, such as limited computing power and energy constraints. In addition, by combining traditional feature selection technology and CNN

TABLE 4 Performance Indicators of Four Attacks on Datasets ISCxVPN2016, CICIDS2018, TON IoT and CIC IoT 2023 datasets.

Dataset	Attack	Accuracy	F1	Recall
ISCXVPN2016	Dos	93.85	84.83	91.26
	U2R	95.72	86.42	90.01
	R2L	92.75	85.89	87.75
	Probe	95.07	82.68	88.55
CICIDS-2018	Dos	93.33	89.47	92.53
	U2R	93.69	90.14	94.53
	R2L	92.37	90.54	92.01
	Probe	93.09	89.63	92.68
TON IoT	Dos	93.58	90.71	93.77
	U2R	95.69	90.61	94.78
	R2L	92.33	89.34	92.48
	Probe	94.99	90.56	94.09
CIC IoT 2023	Dos	94.85	91.5	94.62
	U2R	94.01	90.58	93.85
	R2L	93.2	90.55	93.01
	Probe	90.53	89.95	90.1

+ LSTM, our method can effectively reduce feature dimensions, improve model accuracy and efficiency, and adapt to the resource constraints of UAVs. These factors together make our method show obvious advantages in the experiment.

5 Conclusion

In conclusion, the intrusion detection model of UAV system based on neural network proposed in this paper performs well. This model combines the advantages of convolutional neural network (CNN) and long-term and short-term memory network (LSTM), which can effectively improve the recognition accuracy of UAV system abnormal behavior, and achieve efficient and accurate intrusion detection. Through a large number of experiments on three different data sets, we verify the effectiveness of the model. These data sets cover various indicators of UAV operation, ensuring the comprehensiveness and reliability of experimental results.

Despite the strong performance of our CNN + LSTM model, it has limitations in practical UAV applications. The model relies on preprocessed network traffic data, which may be disrupted by real-time noise or incomplete data packets in dynamic UAV environments, potentially reducing detection accuracy. Additionally, while optimized for efficiency, the model's computational requirements (2.5 MB, 12 ms inference) may still strain low-end UAV hardware with sub-1 GHz processors. The feature selection process, although robust, may overlook subtle attack patterns in highly imbalanced datasets. Future work will focus on integrating online learning to adapt to noisy data, exploring model pruning techniques to further reduce computational demands, and incorporating anomaly detection for rare attack types to enhance robustness in diverse UAV scenarios.

The experimental results show that the CNN + LSTM model has significant advantages in processing complex multidimensional UAV operation data. CNN can extract spatial features, capture local correlation and spatial hierarchy in data; LSTM is good at processing time series data, learning and remembering long-term dependencies. This combination of spatio-temporal characteristics makes the model significantly improve its prediction performance, especially when dealing with complex nonlinear relationships and data noise. In addition, the model also performs well in key evaluation indicators such as recall rate and F1 score, which can effectively identify the abnormal behavior of the UAV system and provide timely warning and decision support for the safe operation of the UAV system.

The core novelty of this work lies in tailoring a resource-aware feature selection and feature-mapping pipeline to the UAV deployment context and in empirically demonstrating trade-offs between dimensionality reduction and detection performance under constrained compute budgets. The convolutional–recurrent backbone used here (CNN + LSTM) is not proposed as a fundamentally new learning architecture; rather, it is applied and optimized for resource-constrained intrusion detection by careful selection of kernels and layer widths, an ensemble feature-selection procedure with weighted integration, and a mapping of non-key to key feature subspaces to recover discriminative information lost during aggressive feature pruning. This explicit emphasis on computational economy and practical deployment trade-offs differentiates the contribution from algorithmic-only advances.

This work was partly supported by the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province (Grant No. SKLACSS-202307), partly supported by Open Fund of Key Laboratory of Flight Techniques and Flight Safety, CAAC(No. FZ2021KF19) and partly supported by National Natural Science Foundation of China (No. U2333211).

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

WZ: Funding acquisition, Writing – original draft, Formal Analysis, Conceptualization, Data curation, Writing – review and editing. BL: Project administration, Writing – review and editing,

Methodology, Resources, Investigation. XZ: Writing – original draft, Supervision, Software, Visualization, Validation.

Funding

The authors declare that no financial support was received for the research and/or publication of this article. This work was partly supported by the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province (Grant No. SKLACSS-202307), partly by the Open Fund of Key Laboratory of Flight Techniques and Flight Safety, CAAC (No. FZ2021KF19), and partly by the National Natural Science Foundation of China (No. U2333211).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The authors declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

1. Altunay HC, Albayrak Z. A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks. *Engineering Science Technology-An International Journal-Jestech* (2023) 38:101322. doi:10.1016/j.jestch.2022.101322

2. Javed AR, Rehman SU, Khan MU, Alazab M, Reddy GT. CANintelliIDS: detecting In-Vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. *IEEE transactions on network science engineering* (2021) 8(2):1456–1466. doi:10.1109/TNSE.2021.3059881

3. Lu J, Zhang W, Hamzei M, Jafari N. The applications of machine learning mechanisms in the compositions of internet of things services: a systematic study, current progress, and future research agenda. *Eng Appl Artif Intelligence* (2025) 147:110345. doi:10.1016/j.engappai.2025.110345

4. Lu JZ, Wang CL, Su J, Ding K, Liu X. An adversarial example defense algorithm for intelligent driving. *IEEE Netw* (2024) 38:98–105. doi:10.1109/mnet.2024.3392582

5. Guo Y, Peng Y, Run H, Xiang T. Capturing spatial–temporal correlations with attention based graph convolutional network for network traffic prediction. *J Netw Computer Appl* (2023) 220:103746. doi:10.1016/j.jnca.2023.103746

6. Gao H., Zheng Y, Li N, Li Y, Qin Y, Piao J, et al. A survey of graph neural networks for recommender systems: challenges, methods, and directions. *ACM Trans Recommender Syst* (2023) 1:1–51. doi:10.1145/3568022

7. Wu C, Liu X, Ding K, Xin B, Lu J, Liu J, et al. Attack detection model for BCoT based on contrastive variational autoencoder and metric learning. *J Cloud Comput* (2024) 13(1):125. doi:10.1186/s13677-024-00678-w

8. Huang L Discussed the security, governance, and challenges associated with the new generation of cyber-physical-social systems, highlighting key considerations for their development and deployment. (2025)

9. Wu ZH, Pan SR, Chen FW, Long GD, Zhang CQ, Yu PS. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks learning systems* (2021) 32(1):4–24. doi:10.1109/tnnls.2020.2978386

10. Bilot T, El Madhoun N, Agha KA, Zouaoui A. Graph neural networks for intrusion detection: a survey. *IEEE Access* (2023) 11:49114–49139. doi:10.1109/ACCESS.2023.3275789

11. Huang Y, Yu J, Wang D, Lu X, Dufaux F, Guo H, et al. Learning-based fast splitting and directional mode decision for VVC intra prediction. *IEEE Trans Broadcasting* (2024) 70(2):681–692. doi:10.1109/tbc.2024.3360729

12. Lu J, Zhang W, Hamzei M, Jafari N. The applications of machine learning mechanisms in the compositions of internet of things services: a systematic study, current progress, and future research agenda. *Eng Appl Artif Intelligence* (2025) 147:110345.

13. Syed NF, Ge M, Baig Z. Fog-cloud based intrusion detection system using recurrent neural networks and feature selection for IoT networks. *Computer Networks* (2023) 225:109662. doi:10.1016/j.comnet.2023.109662

14. Mushtaq E, Zameer A, Umer M, Abbasi AA. A two-stage intrusion detection system with auto-encoder and LSTMs. *Appl Soft Comput* (2022) 121:108768. doi:10.1016/j.asoc.2022.108768

15. Kanna PR, Santhi P. Hybrid intrusion detection using map reduce based black widow optimized convolutional long short-term memory neural networks. *Expert Syst Appl* (2022) 194:116545. doi:10.1016/j.eswa.2022.116545

16. Kanumalli SS, Lavanya K, Rajeswari A Ascalable network intrusion detection system using bi-lstm and cnn[C]//2023 third international conference on artificial intelligence and. *Smart Energy(icais) IEEE* (2023) 1–6.

17. Ullah F, Ullah S, Srivastava G, Lin JCW. IDS-INT:Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. *Digital Commun Networks* (2024) 10(1):190–204. doi:10.1016/j.dcan.2023.03.008

18. Lu J, Zhang W, Hamzei M, Jafari N. The applications of machine learning mechanisms in the compositions of internet of things services: a systematic study, current progress, and future research agenda. *Eng Appl Artif Intelligence* (2025) 147:110345. .

19. Ren K, Yuan S, Zhang C, Shi Y, Huang Z. CANET:A hierarchical cnn-attention model for network intrusion detection. *ComputerCommunications* (2023) 205:170–181. doi:10.1016/j.comcom.2023.04.018

20. Chen C, Song Y, Yue S, Xu X, Zhou L, Lv Q, et al. Fcnn-se:An intrusion detection model based on a fusion CNN and stacked ensemble. *Appl Sci* (2022) 12(17):8601. doi:10.3390/app12178601

21. Park C, Lee J, Kim Y, Park JG, Kim H, Hong D. An enhanced AI-based network intrusion detection system using generative adversarial networks. *IEEE Internet Things J* (2022) 10(3):2330–2345. doi:10.1109/jiot.2022.3211346

22. Yuan L, Yu S, Yang Z, Duan M, Li K. A data balancing approach based on generative adversarial network. *Future Generation Computer Syst* (2023) 141:768–776. doi:10.1016/j.future.2022.12.024

23. Altaf T, Wang X, Ni W, Liu RP, Braun R. NE-GConv:A light weight node edge graph convolutional network for intrusion detection. *Comput and Security* (2023) 130:103285. doi:10.1016/j.cose.2023.103285

24. Duan G, Lv H, Wang H, Feng G. Application of a dynamic line graph neural network for intrusion detection with semisupervised learning. *IEEE Trans Inf Forensics Security* (2022) 18:699–714. doi:10.1109/tifs.2022.3228493

25. Abdulganiyu OH, Ait Tchakoucht T, Saheed YK. A systematic literature review for network intrusion detection system (IDS). *Int J Inf Security* (2023) 22(5):1125–1162. doi:10.1007/s10207-023-00682-2

26. Veeraiah V Enhancement of meta verse capabilities by iot integration. In: 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). IEEE (2022). p. 1493–1498.

27. Ivanovic S, Antonijevic M, Perisic J, Jovanovic L, Zivkovic T, Zivkovic M, et al. The IoT system intrusion detection with CatBoost tuned by modified chimp optimization algorithm. In: M Saraswat, A Rajan, A Chakravorty, editors. *Congress on smart computing technologies. CSCT 2024. Smart innovation, systems and technologies*, 121. Singapore: Springer (2025). p. 261–275. doi:10.1007/978-981-96-6254-8_20

28. Zivkovic M, Bacanin N, Antonijevic M, Nikolic B, Kvascev G, Marjanovic M, et al. Hybrid CNN and XGBoost model tuned by modified arithmetic optimization algorithm for COVID-19 early diagnostics from X-ray images. *Electronics* (2022) 11(22):3798. doi:10.3390/electronics11223798

29. Huang Y, Lu X. Editorial: security, governance, and challenges of the new generation of cyber-physical-social systems. *Front Phys* (2024) 12:1464919. doi:10.3389/fphy.2024.1464919

30. Huang Y, Yu J, Wang D, Lu X, Dufaux F, Guo H, et al. Learning-based fast splitting and directional mode decision for VVC intra prediction. *IEEE Trans Broadcasting* (2024) 2(1):1–12.