A Joint Secure Mechanism of Multi-task Learning For A UAV Team Under FDI Attacks

Rongfei Zeng, Chenyang Jiang, Xingwei Wang, and Baochun Li

Abstract—A UAV team shows tremendous potential for various mobile scenarios. However, some evidences reveal their vulnerability to False Data Injection (FDI) attacks, which can significantly jeopardize the flight security or even lead to catastrophic incidents. Existing studies primarily focus on detecting or defending against FDI attacks at the trajectory control of individual UAVs, leaving a gap in a comprehensive secure mechanism that can simultaneously detect, localize, and compensate for such attacks across an entire UAV team. The complexity of developing such a solution is magnified by the multiple design goals, the inherent sophistication of UAV team, and practical attack assumptions. In this paper, we propose a joint secure framework based on multi-task deep learning to simultaneously detect FDI attacks, localize the compromised components, and compensate control signals to mitigate the impact of FDI attacks on promising UAV teams. Specifically, we design an all-in-one deep learning model framework with a temporal-spatial information extraction module and a hierarchical multi-task module to perform three tasks simultaneously. Moreover, we introduce an iterative learning method with experience replay to counteract knowledge decay during model training. Extensive experiments and real flight demonstrations are presented to validate the improved performance and the benefits of our proposed secure method. A demonstration video and our source code can be accessed via https://github.com/WingFeiTsang/JS/.

Index Terms—A UAV Team, FDI Attacks, Detection and Compensation, Multi-task Learning.

1 Introduction

Unmanned Aerial Vehicles (UAVs) have recently spurred a broad spectrum of real-world applications, from agriculture [1] and military defense [2] to disaster rescue [3], [4]. For example, UAVs are transforming traditional farming practices, ushering in a new era of smart and precision agriculture. Their roles span from fertilization and crop monitoring to pesticide spraying [5]. A prominent consultancy company forecasts that the market value of agricultural UAVs will reach $144.8 million dollars by 2025 in USA, with a remarkable CAGR of 9.96% [6]. Other notable applications include the coordination of over 5,000 UAVs for aerial show in China and the deployment of 103 UAVs, known as ”Sparrows”, by the U.S. military for autonomous intelligence gathering and surveillance. In a nutshell, UAV teams are becoming central to the UAV industry, powering advancements across various fields.

Security concerns are increasingly critical in the UAV community [7]. Over the past few years, False Data Injection (FDI) attacks have been reported frequently in real-world scenarios, with UAV actuators being primary targets [8]. For example, the DJI Phantom 3 Standard, a top-selling UAV, has been compromised by FDI attackers through the use of the GNU Radio Companion application [9]. Similarly, the Parrot Bebop 2 Standard, another popular commercial UAV, has been found vulnerable to FDI attacks [10]. In these attacks, attackers manipulate actuators or interfere with data transmission between UAVs and Ground Control Stations (GCS), aiming to disrupting flight trajectories or even damage critical UAV equipments [11]. The potentially catastrophic outcomes underscores an urgent need for defense mechanisms against FDI attacks to safeguard the secure flight of UAV teams along pre-defined trajectories.

Previous studies have been conducted to detect and ameliorate FDI attacks on UAVs, and they can be categorized into three types. The first category focuses on the Bad Data Detection (BDD) techniques, employing residual-based detection methods to identify FDI attacks [12]. For example, [13] compares the l_2-norm of measured residuals with a predefined threshold to detect potential attacks. The other two categories assume that FDI attacks may evade BDD detection and instead concentrate on FDI attacks against UAV sensors and actuators separately. Many solutions in these categories use analytical methods for detection and compensation. Notably, [14] and [11] provide FDI detection and compensation by analyzing trajectory tracking errors and state estimation errors for individual UAVs. They formulate FDI attacks against actuators as additive and multiplicative attacks. In [15], the authors extend Kalman filter to estimate the true status and detect FDI attacks against actuators. Recently, machine-learning-based detection methods such as [16], [17] mushroom and have received a rocketing interest from both academia and industry. For example, Convolutional Neural Networks (CNN) are employed to detection FDI attacks against UAV camera devices. Additionally, the

- *This work was supported in part by the National Key R&D Program of China under Grant 2023YFB3106900, and in part by the National Natural Science Foundation of China under Grants 62172083, 62032013, and 62472079. (Corresponding author: Xingwei Wang)*

- *Rongfei Zeng is with Software College, Northeastern University, Shenyang, China (e-mail: zengrf@swc.neu.edu.cn). Chenyang Jiang is with in AliCloud, Hangzhou, China (jiangchenv11@gmail.com). Xingwei Wang is with Department of Computer Science and Engineering, Northeastern University, Shenyang, China (e-mail:wangxw@mail.neu.edu.cn). Baochun Li is with Department of Electrical and Computer Engineering, University of Toronto, Canada (e-mail: bli@ece.toronto.edu).*

combination of CNN with nearest neighbor interpolation has shown promising performance improvements in FDI detection. In [18], Hassan et al. apply CNN to encoded Wi-Fi traffic data to detect FDI attacks for individual UAVs.

However, several critical issues remain unresolved in previous studies, and manifest challenges are rooted in these under-explored problems. The majority of studies concentrate on FDI attacks against a single UAV device, with limited attention paid to the more prevalent scenario of UAV team, where the complexity of its security solution is exponentially compounded by multiple UAVs in a team being compromised simultaneously. Additionally, it seems to be intractable to provide a comprehensive security solution that can simultaneously address the three critical objectives of FDI attacks detection, localization, and compensation within a single deep learning model. The joint security solution with these goals has been ignored by previous studies. Furthermore, existing studies often assume that attacks occur independently on either sensors or actuators, neglecting more complex situations where both sensors and actuators may be simultaneously compromised by FDI attackers. This practical threat assumption renders previous methods ineffective, as they are left with little reliable information to utilize. Finally, while many deep-learning-based security schemes have been proposed in other domains, such as smart grid, they are not directly applicable to UAV teams. This limitation arises because previous training methods may result in knowledge decay, wherein the performance of deep learning models deteriorates over prolonged periods of detection and compensation.

In this paper, we aim to empower a UAV team to fly along a designated trajectory, even under the strict assumption of simultaneous attacks. Specifically, we focus on designing a joint and comprehensive secure mechanism for the tasks of FDI attack detection, localization, and compensation in multi-UAV scenarios. We propose a **J**oint **S**ecurity framework (**JS**) with a multi-task deep learning model which consists of a feature mining module and a multi-task module. Our JS framework generates results for three interrelated tasks in a hierarchical manner. To address the challenge of missing reliable information under practical threat assumptions, we leverage CNN and LSTM blocks to extract hidden temporal-spatial patterns from the training data. Our design builds on ideas from other domains, where incorporating both temporal and spatial features in data analyses has proven to offer significant advantages for deep learning models [19]. To ensure robust performance in continuous attack scenarios, we draw inspiration from experience replay in reinforcement learning and then introduce an iterative learning method to mitigate knowledge decay during long-term attack detection and compensation tasks. Extensive experiments validate the performance improvements of our proposed JS framework, highlighting its advantages across various metrics. Compared with the baseline [14], our JS achieves an average reduction in deviation area of 65.64%. Similar results can be observed with UAV teams of varying sizes. Finally, a realistic system with three UAVs is implemented to showcase the efficacy of our JS framework in practical settings.

The contributions of this paper are threefold:

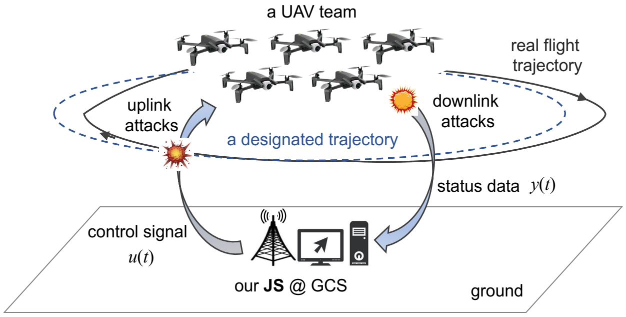


Fig. 1: The system model of a UAV team

- To the best of our knowledge, this is the first paper to investigate FDI attacks on a mainstream yet under-explored UAV team, wherein we adopt a strict assumption of simultaneous FDI attacks against both sensors and actuators. The inclusion of multiple UAV devices and the practical attack assumptions pose significant challenges to the security study in this domain.
- We introduce a novel and comprehensive multi-task deep learning framework to achieve three interrelated sub-goals: FDI attack detection, localization, and compensation. These sub-tasks are facilitated by a shared feature mining module and a hierarchical multi-task module within our JS framework, which fully leverages the temporal-spatial information of UAV teams to counteract the assumed attacks. Our security solution offers some valuable insights for other similar problems in cyber-physical systems like smart grid.
- Our solutions involve three remarkable proprieties: the extraction of temporal-spatial information, a shared and hierarchical model architecture, and iterative learning with experience replay. These three properties endow our JS with performances superior to six baselines, particularly during long-term detection and compensation processes. The practicality and scalability of our JS framework are discussed via experiments and real-world flights.

The remainder of this paper is organized as follows. In Section 2, we present the system model, adversarial model, and our design goal. Section 3 elaborates the proposed joint secure framework (a.k.a JS) for the detection, localization, and compensation of FDI attacks. Extensive experimental results and realistic implementations are presented in Section 4, followed by related work in Section 5. Section 6 concludes the entire paper.

2 Problem Statements and Our Assumption

2.1 System Model

In this paper, we consider a typical UAV team which follows a flight trajectory designated by a GCS to execute cooperative tasks. The UAVs may consist of various types of UAVs, including but not limited to multi-rotor UAVs and fixed-wing UAVs. Each UAV is modeled as a rigid body. The number of UAVs, denoted by N, remains unconstrained throughout this study. These UAVs are individually controlled by a shared GCS on the ground through wireless communication. Specifically, the GCS periodically transmits control signal $u(t)$ at time t to each UAV, a process referred to as uplink communication for simplicity in this paper. In practice, most mainstream UAVs accept three-dimensional target positions as their control inputs. At each time step t, every UAV is assigned a correlated yet distinct target position. Upon receiving $u(t)$, the UAVs navigate directly to the designated positions as instructed by the GCS. In parallel, UAVs periodically send back their instantaneous status data $y(t)$ to the GCS, a process termed downlink communication. The status data includes information such as instantaneous flight velocity, flight attitude, and current three-dimensional position. Notably, the frequency of status responses may differ from that of control signal transmission. Both uplink and downlink communications between UAVs and GCS utilize the widely-adopted MAVLink protocol. The process of control and feedback continues until the task is completed. We show this process in Fig. 1.

2.2 Adversarial Model

In this paper, we focus on trajectory-oriented FDI attacks aimed at diverting multiple UAVs from their designated flight paths. Specifically, we consider uplink attacks and downlink attacks simultaneously. Uplink attacks destroy the integrity of control data, preventing UAVs from following the intended trajectories, while downlink attacks involve the transmission of incorrect status data from UAVs to the GCS, which may result from man-in-the-middle attacks during downlink communication or compromised onboard sensors. In previous studies, defenders could rely on accurate downlink status data to detect only uplink attacks, while accurate uplink control data could be used to defend against only downlink attacks. In this paper, we depart from previous assumptions of isolated attacks and examine a more complex scenario, where adversaries can orchestrate both uplink and downlink attacks simultaneously. This assumption complicates the design of security mechanisms, rendering previous methods ineffective.

Our framework **JS** does not rely on assumptions about specific attacks implementations. Instead, our threat model encompasses a wide range of potential threats, including malware [20], physical layer attacks [21], network layer attacks [22], and additive and multiplicative attacks [23]. The attack-agnostic design enhances the practicality of our scheme, ensuring its applicability to diverse real-world scenarios. However, we assume that adversaries may initiate continuous attacks. Without the assumption of continuous attacks, GCS could transmit correct control signal to UAVs in subsequent time steps, mitigating catastrophic consequences. Additionally, we assume that several UAVs within a team may be susceptible to similar FDI attacks. A further critical assumption is that these attacks can evade the detection by the Bad Data Detection (BDD) module, a primary security mechanism in UAVs. This assumption underscores the importance of our proposed framework, which focuses on identifying and mitigating more sophisticated and elusive attacks.

2.3 Design Goal

Our goal is to develop a comprehensive security scheme that enables a UAV team to maintain its intended trajectory despite the attack scenarios described above. This overarching goal is decomposed into three specific sub-objectives: FDI attack detection, attack localization, and control signal compensation. The FDI attack detection, a binary classification task, serves as a prerequisite module to determine whether a UAV team suffers from attacks. Upon identifying victim UAVs, the localization module classifies potential attacks into either uplink attacks or downlink attacks. For uplink attacks, the compensation module adjusts the victim UAVs' flight trajectories by sending compensated control signals. In contrast, if only downlink attacks are detected for a UAV, no further action is necessary since the UAV can still fly along its designated trajectory. The rationale behind this joint design is to provide end-to-end security guarantees by addressing these three interrelated objectives within a single, unified deep learning model. This joint framework might mitigate the incompatibility or inefficiency issues observed in previously separated security mechanisms [11].

3 JS: THE JOINT SECURITY FRAMEWORK

In this section, we elaborate our proposed joint security framework **JS** of temporal-spatial information extraction from the following aspects: data processing, model architecture, loss function, and iterative learning method.

3.1 Data Processing

In JS, we utilize supervised learning to extract temporal-spatial patterns from routine flight data. In this context, a training sample is represented as $(x_i, y_i) \in \mathcal{D}_{tr}$, where x_i denotes the input data to JS and y_i represents its corresponding annotation (a.k.a label). The input data x_i has a shape of (T, N, M), where T represents the number of time steps, N corresponds to the number of UAVs in a team, and M denotes the number of features.

For a specific UAV j, the input features include status data collected from onboard sensors as well as control signals sent by the GCS. Formally, the input feature data for j-th UAV at time step t can be expressed as $F_j^t = [F_{j,1}^t, F_{j,2}^t, \ldots, F_{j,M}^t]$. In this paper, we exemplify these features as instantaneous position, velocity, acceleration, attitude, equivalent control force, and control signals. It should be noted that our JS is not constrained to these specific features and advanced feature engineering methods [24] can be applied within our framework. The definition of input x_i reveals that it encapsulates temporal information that characterizes the flight status of UAVs over a series of time steps. Additionally, x_i includes features for all UAVs at each time step, capturing spatial information that reflects hidden patterns and interrelations among the UAVs. These temporal-spatial patterns are crucial for addressing the absence of reliable information under our practical threat assumption.

Our JS employs multi-task learning to perform FDI attack detection, localization, and compensation. The corresponding data annotations (a.k.a label) consist of an attack

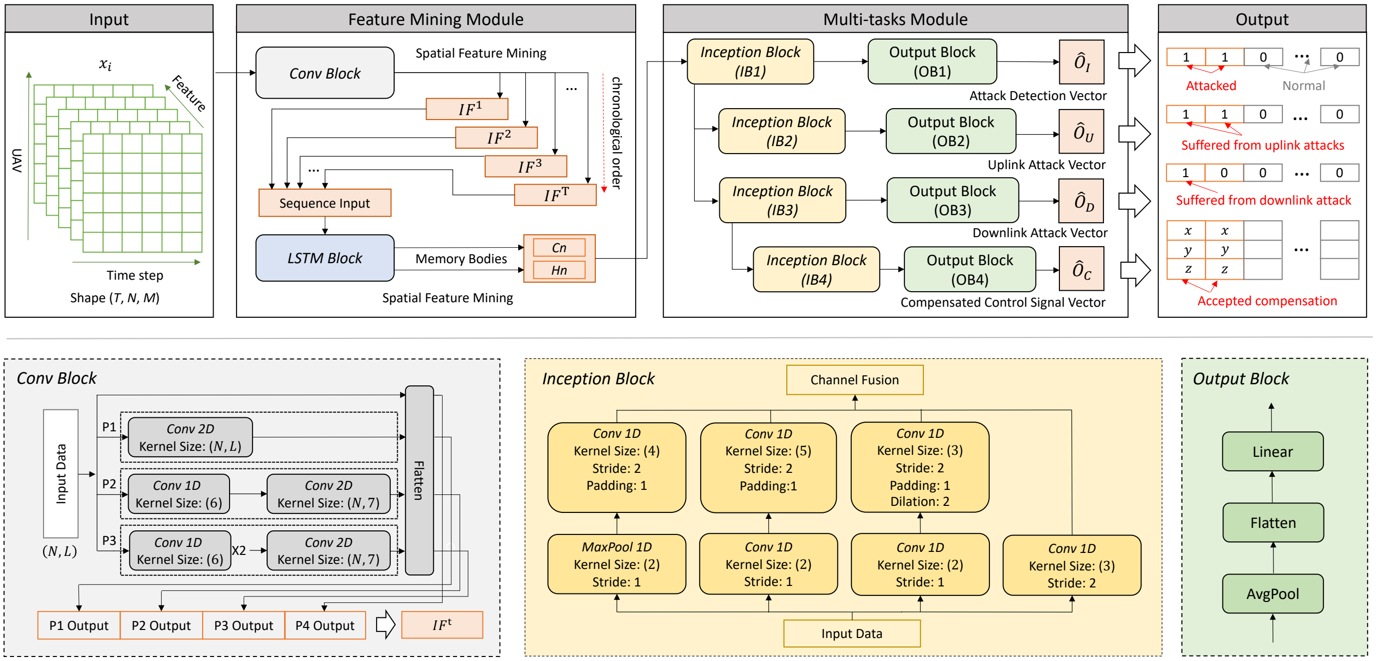


Fig. 2: The model architecture of our JS and the zoom in of $Conv$ block, inception block, and output block

detection vector O_I[1], two attack localization vectors O_U and O_D, and a compensated control signal vector O_C. The attack detection vector O_I, with a shape of shape $(1, N)$, indicates the probability that each UAV suffers from FDI attacks. During the model training phase, each element of the ground-truth vector O_I is binary. In the inference phase, the j-th UAV is identified to be attacked if $\widehat{O}_{I,j} \geq \alpha$, where α is a predefined threshold hyperparameter. Moreover, we use two attack localization vectors O_U and O_D to separately denote the probabilities of uplink attacks and downlink attacks. Importantly, $\widehat{O}_{U,j}$ and $\widehat{O}_{D,j}$ are activated only when the attack detection vector indicates that the j-th UAV is under attack, i.e., $\widehat{O}_{I,i} \geq \alpha$. Finally, the vector O_C represents the compensated control signal required to correct the flight trajectory of the UAV team. Similarly, the compensated control signal $\widehat{O}_{C,j}$ is enabled only when the model detects FDI attacks on the j-th UAV. This ensures that control signal compensation is only triggered when necessary.

3.2 Model Architecture Design

The model architecture of JS consists of two main modules: a feature mining module and a multi-task module. The details of these modules are illustrated in Fig. 2.

Feature Mining Module. The goal of the feature mining module is to efficiently explore temporal-spatial patterns from training data, which provides a solid foundation for the subsequent multi-task module. In this paper, we design the feature mining module with CNN blocks and LSTM blocks to extract temporal-spatial information.

Initially, the input data x_i is fed to the $Conv$ block, which processes the input data x_i at a single time step. Given the diverse granularity requirements of spatial information across our three tasks, we adopt the idea of feature pyramid to design the $Conv$ block with three distinct paths. Each path utilizes different receptive fields and numbers of kernels to extract hidden information at different granularities. The outputs from three paths are concatenated, resulting in a high-dimensional feature vector IF^t that represents the spatial information in x_i at time step t.

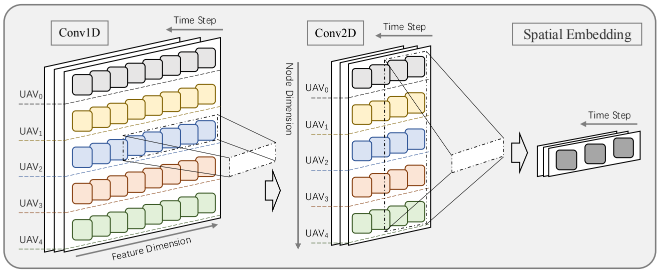


Fig. 3: The feature mining process of P2 in $Conv$ block

Specifically, we use $Conv1D$ and $Conv2D$ as basic operators in each path. $Conv1D$ captures potential patterns among features within a single UAV, while $Conv2D$ uncovers hidden information among different UAVs in the team. Taking the path P2 in the $Conv$ block as an example, we first apply the $Conv1D$ operator along the feature dimension of a single UAV at an individual time step. The result is then processed by the $Conv2D$ operator, which performs convolutional calculations on the previous output within the same time step. This process is illustrated in Fig. 3. As shown in Fig. 3, the combination of these two operators enhances data mining capabilities.

The outputs of $Conv$ block are passed to the $LSTM$ block to further explore temporal patterns along the time dimension. Specifically, a sequence of feature vectors IF_i^t $(t = 1, \cdots, T)$ is input into the $LSTM$ block, which consists of L

[1] In this paper, the ground-truth label is denoted as O_I, with the corresponding model output denoted as $\widehat{O}_I$. Similar notations apply to two attack localization vectors and a compensated control signal vector as well.

Algorithm 1 Model Parameter Update Algorithm of JS

```
Input: initial parameters $\theta$, learning rate $\eta$, dataset $\mathcal{D}_{tr}$
Output: new parameters $\theta'$
1:  for each batch $(X_b, Y_b)$ from $\mathcal{D}_{tr}$ do
2:      $\widehat{O}_I, \widehat{O}_U, \widehat{O}_D, \widehat{O}_C = model_{JS}(\theta, X_b)$;
3:      $\mathcal{L}_I = CE(\widehat{O}_I O_I)$;
4:      if $\exists i \in [0, N)\ s.t.\ Q_I[i] == 1 \parallel \widehat{Q}_I[i] \geq \alpha$ then
5:          $\mathcal{L}_U = CE(\widehat{O}_U, O_U)$;
6:          $\mathcal{L}_D = CE(\widehat{O}_D, O_D)$;
7:          if $\exists i \in [0, N)\ s.t.\ Q_D[i] == 1 \parallel \widehat{Q}_D[i] \geq \alpha$ then
8:              $\mathcal{L}_C = MSE(\widehat{O}_C, O_C)$;
9:      Compute $G_I^{[FM]}, G_I^{[IB_1]}, G_I^{[OL_1]}$ from $\mathcal{L}_I$
10:     Compute $G_U^{[FM]}, G_U^{[IB_1]}, G_U^{[IB_2]}, G_U^{[OL_2]}$ from $\mathcal{L}_U$
11:     Compute $G_D^{[FM]}, G_D^{[IB_1]}, G_D^{[IB_3]}, G_D^{[OL_3]}$ from $\mathcal{L}_D$
12:     Compute $G_C^{[FM]}$,$G_C^{[IB_1]}$,$G_C^{[IB_3]}$,$G_C^{[IB_4]}$,$G_C^{[OL_4]}$ from $\mathcal{L}_C$
13:     $G_{FM}, G_{ML}$ are calculated by $G_I$,$G_U$,$G_D$,$G_C$;
14:     $G = G_{FM} | G_{ML}$;
15:     $\theta' = Optimizer(\theta, G)$;
```

layers, each containing T cells. The output of the $LSTM$ block contains two elements, c_L and h_T, which are the cell state and the hidden state of the last cell, respectively. These two states are then stacked and forwarded to the next module.

Multi-task Module. The multi-task module is designed to produce three interrelated outputs. The FDI attack detector $\widehat{O}_I$ provides a simple and coarse-grained result, while the compensated control signal vector $\widehat{O}_C$ offers a more complex and fine-grained output. Each output has specific prerequisites for activation, reflecting their varying levels of complexity. To accommodate these characteristics, we propose a hierarchical architecture for the multi-task module, as depicted in Fig. 2.

The rationale for this hierarchical design is based on three observations: (1) shallow layers in a deep learning model are effective for predicting simple and high-level results, such as detecting whether a UAV suffers from FDI attacks; (2) deeper layers are capable of extracting precise and fine-grained flight details, which are crucial for generating compensated control signals; and (3) all these correlated tasks benefit from share layers to improve model efficiency and facilitate training.

As illustrated in Fig. 2, the multi-task module consists of four GooLeNet-inspired inception blocks (i.e., $IB1$, $IB2$, $IB3$, and $IB4$) organized in a hierarchical manner. Specifically, the output of inception block $IB1$ is forwarded to both $IB2$ and $IB3$, and the output of $IB3$ is further passed to $IB4$. These inception blocks share similar structures and configurations, differing only in their input and output channel sizes. Finally, the outputs of these inception blocks are separately processed by four output blocks, each implemented as a fully connected network. This hierarchical and shared-layer design enables the multi-task module to effectively balance the demands of coarse-grained detection and fine-grained control signal generation while maintaining efficiency and scalability.

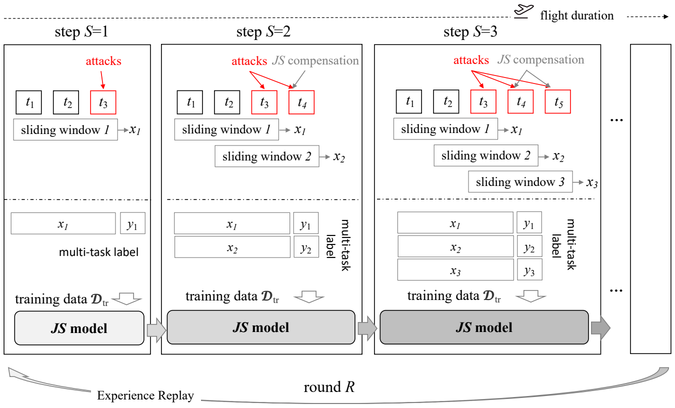


Fig. 4: The iterative model training method with experience replay

3.3 The Computation of Losses and Gradients

In supervised learning, computing gradients from losses is a fundamental step for model parameter updates. In our JS framework, the attack detection and localization can be categorized as the classification task, for which we employ the Cross-Entropy (CE) loss, while control signal compensation is a regression task, addressed using the Mean-Squared-Error (MSE) loss. Notably, some losses become only valid when their prerequisites are satisfied. For example, we compute the CE loss for the output $\widehat{O}_U$ only when certain UAVs are identified as victims of FDI attacks.

Gradient computation in JS is more complex due to the interdependence of tasks. Gradients for some modules originate from multiple losses. For example, the gradients of the feature mining module are influenced by all the losses, whereas the gradient of the inception block IB_4 are computed solely based on $\widehat{O}_C$. The gradient computation for each module in JS is formalized in Eq. 1.

$$\begin{aligned}G_{FM} =& G_I^{[FM]} + G_U^{[FM]} + G_D^{[FM]} + G_C^{[FM]} \\ G_{IB_1} =& G_I^{[IB_1]} + G_U^{[IB_1]} + G_D^{[IB_1]} + G_C^{[IB_1]} \\ G_{IB_3} =& G_D^{[IB_3]} + G_C^{[IB_3]} \\ G_{ML} =& G_{IB_1} \mid G_I^{[OB_1]} \mid G_U^{[IB_2]} \mid G_U^{[OB_2]} \mid G_{IB_3} | \\ & G_D^{[OB_3]} \mid G_C^{[IB_4]} \mid G_C^{[OB_4]}\end{aligned} \tag{1}$$

where G_I, G_U, G_D, and G_C represent gradient matrices obtained from the losses of $\mathcal{L}_I$, $\mathcal{L}_U$, $\mathcal{L}_D$ and $\mathcal{L}_C$, respectively. The superscript $[FM]$ of G_I denotes the tensor slices related to feature mining module in G_I. Similar notations are applied to the subscripts of $[ML]$ (a.k.a. multi-task module), $[IB]$ (a.k.a. Inception block) and $[OB]$ (a.k.a. output block). The operator | represents matrix concatenation operation.

The process for updating model parameters is outlined in Algorithm 1. This algorithm computes gradient matrices for each module based on Eq. (1). Lines 3-8 compute loss values for each task, and Line 9-14 calculate the corresponding gradient matrices. Line 15 performs the parameter updates using the computed gradients.

3.4 An Iterative Model Training Method

In practical scenarios, attackers may exploit the inherent vulnerabilities of UAVs to launch continuous FDI attacks,

Algorithm 2 An Iterative Training with Experience Replay

```
Input: epoch $E$, step $S$, round $R$, learning rate $\eta$
Output: model parameters $\theta_{R,S}$
Randomly initialize model parameters $\theta_{0,S}$;
for $r$=1 to $R$ do
    $\theta_{r,0}$ = $\theta_{r-1,S}$;
    for $s$=1 to $S$ do
        Dataset $\mathcal{D}_{tr} = []$;
        $\theta_{r,s}$ = $\theta_{r,s-1}$;
        for $i$=1 to $s$ do
            $x_i = [t_i, t_{i+1}, t_{i+2}]$ and get multi-task label $y_i$;
            $\mathcal{D}_{tr}.append((x_i, y_i))$;
            $\widehat{O}_I, \widehat{O}_U, \widehat{O}_D, \widehat{O}_C = model_{JS}(\theta_{r,s}, x_i)$;
            for $j$=0 to $N-1$ do
                if $\widehat{O}_I[j] \geq \alpha$ && $\widehat{O}_D[j] \geq \alpha$ then
                    Replace $t_{i+3,j}$'s control signal by $\widehat{O}_{C,j}$;
            $t_{i+3} = FDIA(t_{i+3})$;
        for $e$=1 to $E$ do
            Algorithm 1 $(\theta_{r,s}, \eta, \mathcal{D}_{tr})$;
```

necessitating our JS framework to provide a long-term predication and compensation for a multi-UAVs team. However, empirical experiments reveal that it is impractical to train a satisfying JS model for a long-term security defense by loading all the training data directly to JS model. We hypothesize that deep learning models may accumulate errors over prolonged flight, which may stem from inaccurate predictions or noisy training data. Additionally, deep learning models may suffer from catastrophic forgetting, where simple and short-term fight patterns are overlooked after extensive training epochs. These challenges highlight the need for a well-designed training method to effectively identify FDI attacks and compensate a UAV team during long-term flights.

In this paper, we draw inspiration from the principle that deep learning models can incrementally learn complex patterns, and we propose an iterative training method with experience replay to achieve accurate long-term predication and compensation. Our approach begins by training the model from scratch with data for single-step attacks, equipping it with the ability to handle short-term scenarios. Gradually, we increase the number of attacks steps in the training data, enabling the model to improve its long-term predication capability. Once the model demonstrates satisfactory performance for longer-term scenarios, we initiate another training cycle and start with single-step attacks, leveraging experience replay to meliorate knowledge forgetting. This iterative training process continues until the final model achieves satisfactory performance across all training datasets.

The iterative training method with experience replay is illustrated in Fig. 4. Initially, we consider a one-step attack scenario in $Step = 1$, where data samples spanning three time steps (t_1, t_2, t_3) are used to predict t_4. The model is trained on sufficient samples to accurately detect and compensate for one-step attacks. In $Step = 2$, we increase the attack step to two and employ a sliding window approach to generate training samples such as $(t_1, t_2, t_3) \rightarrow t_4, (t_2, t_3, t_4) \rightarrow t_5$, where $\rightarrow$ denotes prediction, and time steps t_3 and t_4 might be affected by attacks. Then, we retrain our model to enhance its prediction capability. The process is repeated for subsequent steps until the predefined maximum step threshold is reached. Subsequently, we restart the training process from $Step = 1$, incorporating previous training samples to reinforce earlier knowledge. We summarize the complete iterative training method in Algorithm 2. In Algorithm 2, the function $FDIA(\cdot)$ represents the impacts of FDIA attacks on control signals.

4 Performance Evaluation

This section evaluates the proposed JS framework through extensive simulation experiments to validate its efficacy. We also implement JS on a three-UAVs testbed to showcase its feasibility in real-world scenarios.

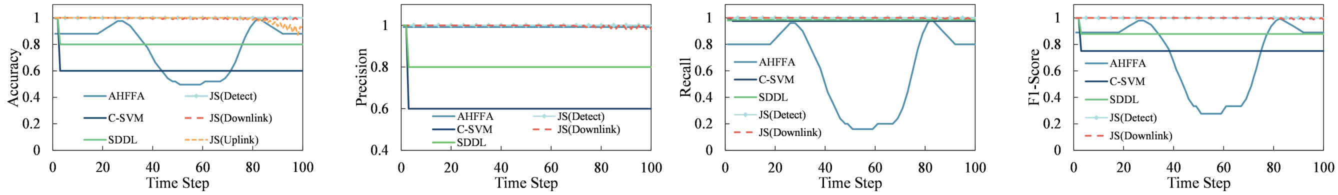
4.1 Experimental Setup

The Flight of a UAV Team. In our experiments, we consider a typical UAV team consisting of five UAVs ($N = 5$). These UAVs follow circular trajectories with a radius of $0.75m$, completing one full circle in 10 seconds, as described in [14], [25]. The centers of these circular trajectories are located at $(0m, 0m)$, $(1m, 1m)$, $(1m, -1m)$, $(-1m, 1m)$, and $(-1m, -1m)$, respectively. During the flight, each UAV receives the control data of its target position from GCS at a frequency of $0.1s$ per target position.

Model and Training. The default configurations of the $Conv$ block, Inception block, and output block are presented in Fig. 2. The $LSTM$ block consists of a single layer with a hidden size of 1024. During model training, we employ the Adam optimizer with a learning rate of $3.5e-5$. Additionally, the default parameters for Algorithm 2 are set as $R = 5$, $S = 60$, and $E = 50$.

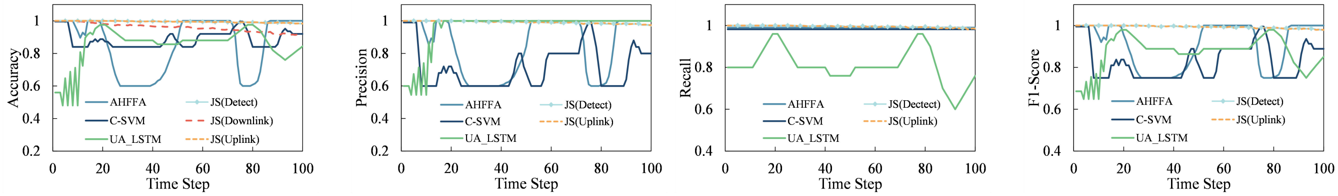
Dataset and Attacks. In our datasets, each training sample consists of five features ($M = 5$): position, velocity, acceleration, attitude, and the control signal of the target position. We adopt the approach from [14], [25] to generate data samples following circular trajectories. Additionally, we use the iterative model training method described in Section 3.4 to construct the training dataset[2]. For uplink attacks, we implement the multiplicative attack from [11] and the additive attack from [26], ensuring that the parameters remain consistent with these prior studies. Two UAVs are randomly selected to inject these uplink attacks. For downlink attacks, we follow the methodologies outlined in [27], [28]. Specifically, two attack vectors y_0^a and y_1^a are alternately added to the flight status data. Once injected, the attack vectors are incrementally updated with $y_\triangle^a$. The parameters of these vectors are provided in Table 1, where (x, y, z) represents the three-dimensional position, (v_x, v_y, v_z) denotes the velocity components, and (a_x, a_y, a_z) indicates the acceleration for each dimension. Finally, these attacks are continuously applied to the status data under downlink attack scenarios.

[2] https://github.com/WingFeiTsang/JS/dataset



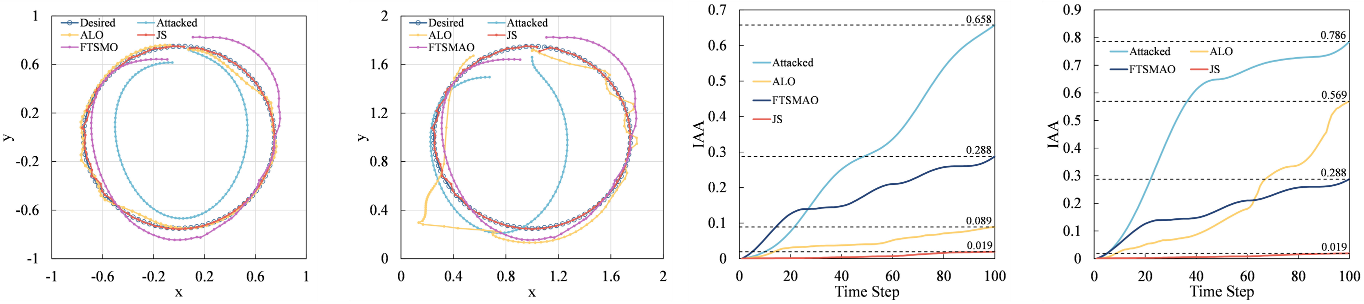
(a) Accuracy (b) Precision (c) Recall (d) F1-score

Fig. 5: The detection and localization performance under downlink attacks.



(a) Accuracy (b) Precision (c) Recall (d) F1-score

Fig. 6: The detection and localization performance under uplink attacks



(a) The compensated trajectories of two compromised UAVs (b) The IAA results of two compromised UAVs

Fig. 7: The compensation performance of JS and baselines under the uplink attacks

TABLE 1: The parameters of downlink attacks

	x	y	z	v_x	v_y	v_z
y_0^a	-0.18	-0.17	0	-0.14	-0.13	0
y_1^a	0.18	0.17	0	0.14	0.13	0
y_Δ^a	-0.002	0.003	0	-0.004	-0.0032	0
	a_x	a_y	a_z	$roll$	$pitch$	yaw
y_0^a	-0.13	-0.12	0	-0.013	-0.012	0
y_1^a	0.13	0.12	0	0.013	0.012	0
y_Δ^a	-0.0012	-0.0009	0	-0.004	-0.001	0

4.2 Metrics and Baselines

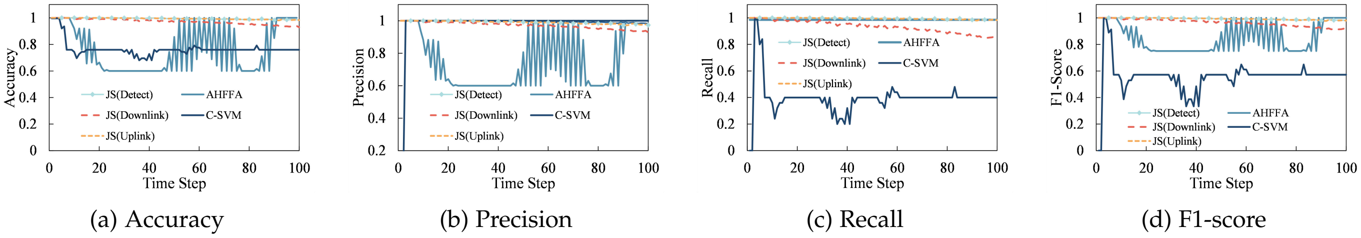
Metrics. In this paper, we address both multi-task classification and regressive compensation tasks, employing two categories of quantitative metrics to evaluate their performance. (1) *The Multi-task Classification Task*. We utilize the widely-adopted metrics, including accuracy, recall, precision, and $F1$-score, to comprehensively assess the performance. (2) *The Regressive Compensation Task*. To evaluate the performance of compensation, we propose a novel metric termed as **Internal Accumulated Area (IAA)**. IAA quantifies the area enclosed by the expected flight trajectory and the compensated flight trajectory. Specifically, we compute this area using Qin Jiushao and Heron's Formula [29]. A smaller IAA value signifies superior compensation performance of the proposed JS.

Baselines. In this paper, we employ UA-LSTM [30], SDDL [31], AHFFA [32], C-SVM [33] as our baseline models to evaluate the attack detection capabilities of our JS. Furthermore, we compare JS against two prevalent schemes ALO [14] and FTSMAO [25] for assessing the compensation performance. Unlike these baselines, our JS has the capabilities of attack detection, localization, and compensation for uplink attacks and downlink attacks simultaneously, which represents a significant contribution of our work. A qualitative comparison between JS and the baselines is provided in Table 2.

4.3 Experimental Results

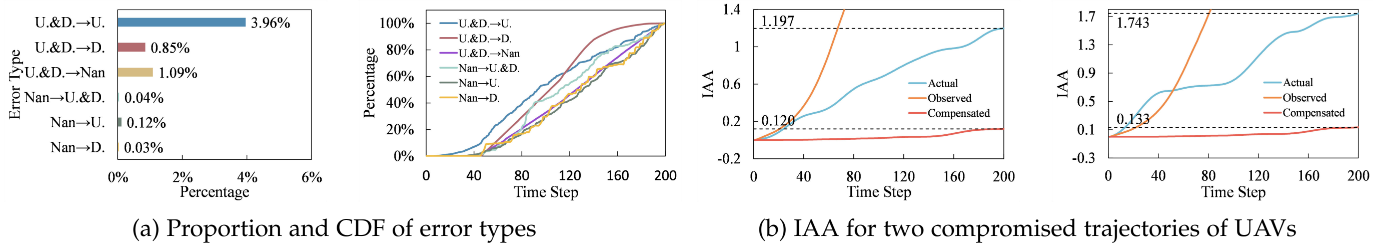
Considering that some baselines are tailored specifically for uplink attacks or downlink attacks, while others provide only partial security defense solutions, we evaluate our scheme against these baselines in three distinct scenarios: downlink attack scenarios, uplink attack scenarios, and simultaneous uplink and downlink attack scenarios.

Downlink Attack Scenarios. When only downlink attacks are the sole threat within a UAV team, the primary



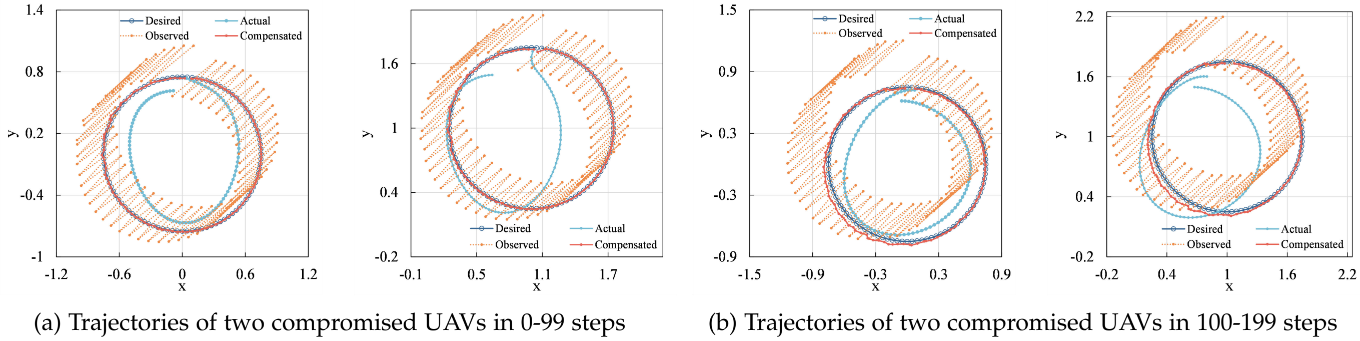
(a) Accuracy (b) Precision (c) Recall (d) F1-score

Fig. 8: The detection and localization performance under uplink and downlink attacks



(a) Proportion and CDF of error types (b) IAA for two compromised trajectories of UAVs

Fig. 9: The statistics of errors and IAA results under both uplink and downlink attacks



(a) Trajectories of two compromised UAVs in 0-99 steps (b) Trajectories of two compromised UAVs in 100-199 steps

Fig. 10: The trajectories of two compromised UAVs

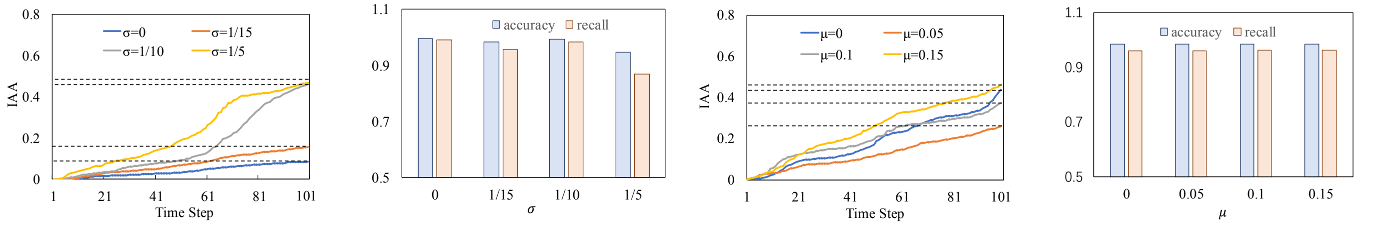
TABLE 2: Qualitative comparisons between JS and baselines

Method	Technique	Functionality			Threats		
		Det.	Loc.	Com.	U.	D.	U.&D.
UA-LSTM [30]	LSTM	√	×	×	√	×	×
SDDL [31]	CNN	√	×	×	×	√	×
AHFFA [32]	LSTM	√	×	×	√	√	√
C-SVM [33]	SVM	√	×	×	√	√	√
ALO [14]	Observers	√	×	√	√	×	×
FTSMAO [25]	Observers	√	×	√	√	×	×
JS	Multi-task NN	√	√	√	√	√	√

Note: “Det.” , “Loc.” and “Com.” represent the attacks detection, localization, and compensation respectively. The notations “U.” , “D.” , and “U.&D.” denote uplink attacks, downlink attacks, and simultaneous uplink and downlink attacks. They are used consistently throughout the entire paper.

focus of JS shifts to attack detection and localization, as no compensation is required. We present the performance of attack detection and localization in Fig. 5. Our findings reveal continuously high performances across metrics such as accuracy, precision, recall, and $F1$-score, highlighting the effectiveness of the proposed JS in detecting and localizing downlink attacks. Furthermore, JS outperforms all baseline methods across these metrics. Although AHFFA and C-SVM exhibit acceptable performance during the initial steps, their effectiveness diminishes under continuous and prolonged attacks scenarios. Additionally, these baselines lack the capability to perform localization, further emphasizing the advantages of our JS. Finally, we validate the correctness of the uplink attack outputs and observe that JS also accurately identifies UAVs not subjected to attacks, underscoring its robustness in diverse scenarios.

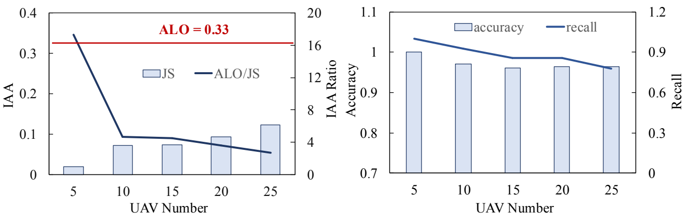
Uplink Attacks Scenarios. In this experiment, we focus on the scenario involving only uplink attacks. The detection and localization performance of uplink attacks is shown in Fig. 6. This results reveal that JS can continuously detects and localizes the uplink attacks across various metrics, demonstrating robust performance even during long-term flights. This reliable detection and localization are critical as they lay the foundation for the subsequent control signal compensation. Notably, our JS demonstrates superior performance compared to the baselines across all four metrics simultaneously. While AHFFA achieves acceptable recall, its precision fall short. In the uplink attack scenario, it becomes imperative to provide compensated control signals to miti-



(a) Compensation and detection performance with different σ

(b) Compensation and detection performance with different μ

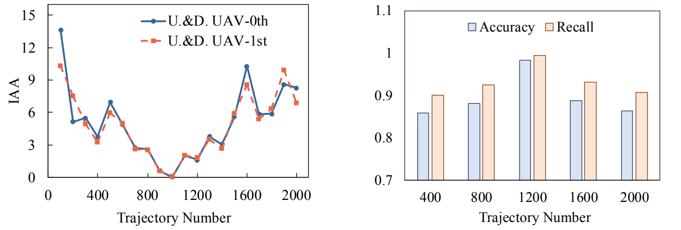
Fig. 11: The robustness property of JS against transmission noise



(a) Compensation performance

(b) Detection performance

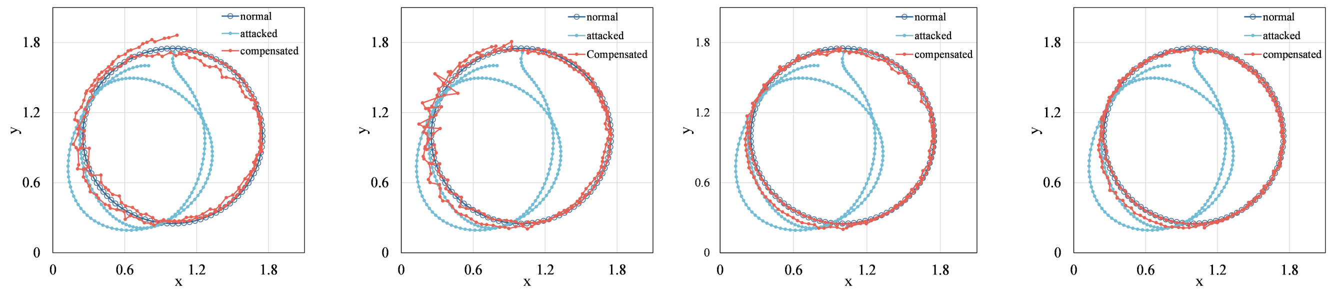
Fig. 12: The scalability performance of JS with varying UAV team size



(a) Compensation performance

(b) Detection performance

Fig. 13: The performance of JS with different numbers of training samples



(a) R=1

(b) R=4

(c) R=7

(d) R=10

Fig. 14: Compensated performance under different rounds.

gate the impacts of such attacks. To evaluate the compensation performance, we compare our JS with two prominent schemes ALO [14] and FTSMAO [25]. It is noteworthy that the performance of these schemes depends on specific hyperparameters, and we present their optimal performance after an extensive hyperparameter search. In Fig. 7(a), we visualize the target and compensated trajectories of two compromised UAVs. These results clearly demonstrate that our JS effectively mitigates the impacts of uplink attacks, enabling compromised UAVs to closely follow the desirable trajectory. In addition, the compensation performance of JS outperforms both ALO and FTSMAO. Furthermore, we provide the quantitative IAA results in Fig. 7(b). IAA results cross-validate the superior compensation capabilities of JS in long-term flights. Notably, our JS exhibits remarkable compensation performance even when multiple UAVs are compromised, whereas ALO and FTSMAO struggle to simultaneously compensate for all compromised UAVs.

The Scenario of Simultaneous Uplink and Downlink Attacks. We begin by presenting the detection and localization results for both uplink and downlink attacks in Fig. 8. The results highlight the exceptional performance of JS again, with high accuracy, precision, recall, and $F1$-score achieved. Interestingly, the detection and localization performance in this simultaneous attack scenario surpasses that in the standalone uplink attack scenario. The improvement may stem from underlying feature pattern that deep learning models leverage to classify potential attacks more effectively. Compared to the baselines, our JS demonstrates significantly enhanced detection and localization capabilities, further reinforcing its advantages in counteracting simultaneous attacks.

In our JS, detection and localization errors predominantly occur during prolonged attacks durations. Notably, 65.07% of these errors arise from detecting only uplink attacks in scenarios where both uplink and downlink attacks coexist within a UAV team. Despite this, the negative impacts of these errors on the secure flight are minor. We show the statistical results in Fig. 9(a), which also shows that other types of errors are minor and exert negligible effects on overall flight security. In summary, our JS achieves satisfactory performance in attack detection and localization, even during long-term flights.

For the compensation task, we compare the performance of our JS with ALO, as shown in Fig. 10. The results show that JS effectively mitigates joint uplink and downlink attacks by compensating the control signals to the compromised UAVs within 200 steps. The compensated flight trajectory aligns closely with the target trajectory, exhibiting only negligible deviations. Furthermore, JS significantly outperforms ALO, particularly as the time steps increase. This conclusion is echoed by the quantitative results in Fig. 9. The IAA of ALO shows a noticeable increase between 100 to 200 time steps, indicating inappropriate compensations during the second cycle of UAV flight. In contrast, our JS demonstrates remarkable stability, enabling the compromised UAVs to consistently follow the target trajectories over extended durations.

Robustness. We evaluate the robustness of the JS framework against transmission noise without retraining our model. In this experiment, both uplink and downlink communications are corrupted by FDI attacks and Gaussian noise with varying means μ and variances σ. Unless explicit stated otherwise, the default experimental configurations are $R = 1$, $S = 60$, $E = 25$, $\mu = 0$, and $\sigma = 0$. The results, shown in Fig. 11, indicate that transmission noise has a limited impact on detection performance across different means or variances. Additionally, Gaussian noise with varying means mu has a negligible effect on the compensation task. When the variance of transmission noise is not excessively high, its impact on compensation performance remains minimal. In conclusion, our JS framework demonstrates robustness against transmission noise.

Scalability. In this experiment, we evaluate the scalability performance of our JS framework with varying UAV team size in Fig. 12. The default settings are $R = 5$, $S = 60$, and $E = 25$. The results in Fig. 12 show that our JS framework achieves stable and high detection performance in terms of accuracy and recall. Additionally, the compensation performance of JS outperforms the baseline ALO in terms of the IAA metric. These results confirm the scalability of JS for middle-scale UAV teams. Due to computational constraints, we leave the investigation of large-scale deployments and the optimization of experimental settings for future work.

Dataset Size. This experiment investigates the model performance with the number of training samples. All experimental settings are kept constant, except for the number of training samples. We present the results of the detection and compensation performance in Fig. 13, where the x-axis represents the number of circular flights in the training dataset. The results in Fig. 13 show that a moderate increase in the number of training samples significantly improves the model performance in terms of IAA and accuracy. In contrast, an excessively large dataset necessitates further optimization of the experimental settings, which is left for future work.

4.4 An Ablation Study of the Iterative Training Method

In this subsection, we investigate the performance of our iterative training method, a critical component of the JS framework, by analyzing the effects of three hyperparameters: the number of rounds R, the step size S, and the number of epochs E. In the following discussion, we present the impact of these parameters on model's performance.

The Number of Round R. In this experiment, we vary R from 0 to 10 and analyze the IAA results, as shown in Fig. 15(a). Note that we keep $S = 60$ and $E = 25$ constant. Additionally, We present the compensated trajectories of compromised UAV under different settings of R in Fig. 14. Experimental results reveal that the IAA metric decreases as R increases, indicating that larger R values enhance the compensation performance. However, this improvement comes at the cost of increased computational overhead. Moreover, excessively large R values may lead to overfitting problems, potentially reducing the model's generalization ability. Based on these observations, we conclude that R values between 2 to 9 provide a suitable balance between performance and training efficiency for our experiments.

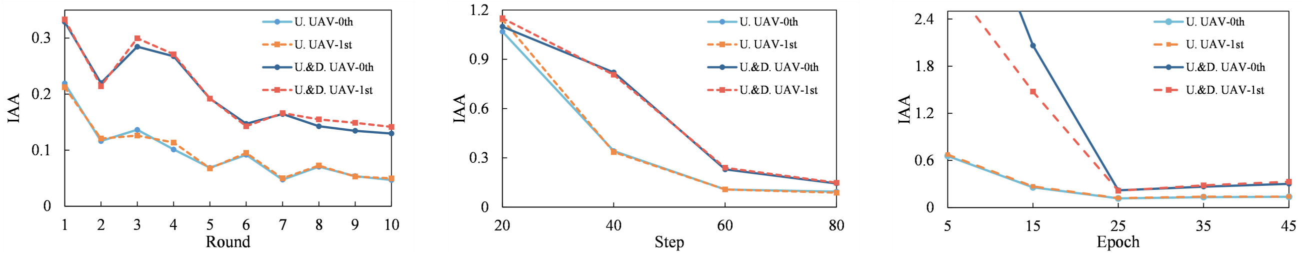
The Step Size S. The parameter S relates to the long-term prediction capability within each round. We set $R = 2$ and $E = 25$ in this experiment. The IAA results for different S values are shown in Fig. 15(b). The results indicate that IAA is a decrease function of the step size S. However, once S exceeds the flight period, the marginal improvement diminishes. As shown in Fig. 15(b), IAA approaches and stabilizes near the lower bound when $S \geq 60$. It is also evident that larger S values significantly increase computational overhead. In this experiment, $S = 60$ emerges as an optimal design choice for this setting.

The Number of Epoch E. In this experiment, we investigate the impact of E on model training, with $R = 2$ and $S = 60$. As shown in Fig. 15(c), increasing E enhances the model's training performance, allowing the deep learning model to better capture patterns from the training data. However, excessively large E values are not appropriate for our JS. Once E exceeds 25, the IAA metric begins to increase, indicating a decline in performance. We hypothesize that excessively large E values may lead to overfitting on prior experiences, limiting the model's ability to adatp to new patterns.

4.5 An Ablation Analysis of the Model Architecture

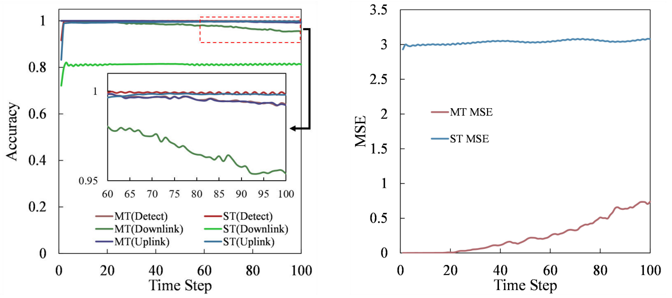
The feature mining module and multi-task training modules are two key components of the JS model. In this subsection, we evaluate their impact on performance by conducting two experiments: replacing the feature mining module with alternative backbones and training each task independently instead of using the multi-task module.

The Feature Mining Module. Previous studies have demonstrated that VGG16 and Transformer models are effective at extracting hidden patterns from training data. In this experiment, we replace our feature mining module with VGG16 and a Transformer model, while keeping the other components fixed. For the Transform, we use a single encoder with eight attention heads and a hidden size of 1024 for the feed-forward neural network. Compared with VGG16 and Transformer, our native feature mining module has the smallest number of learnable parameters, with almost half the parameter size of VGG16. All models are trained using identical hyperparameters ($R = 2, S = 60, E = 25$) under a scenario involving simultaneous uplink and downlink attacks. We show the results of detection and



(a) IAA with different rounds (b) IAA with different steps (c) IAA with different epoches

Fig. 15: The hyperparameters studies of round R, step S, and epoch E



(a) Accuracy (b) MSE

Fig. 16: JS with multi-task learning and single-task learning

localization performance and different trajectories in Fig. 17. The results highlight that our native feature mining module delivers comparable and stable attack detection and localization performance compared to VGG16 and Transformer. In addition, our module outperforms both alternatives in terms of compensation performance, demonstrating its efficiency and effectiveness in the JS framework.

The Multi-task Module. In this experiment, we train some blocks for each task independently and use single-task models as the comparison baselines to evaluate the performance of our multi-task module design. Specifically, only one output block is enabled for each single-task model. All the training settings ($R = 2, S = 60, E = 25$) are kept identical across experiments. The experimental results are shown in Fig. 16. The results show that the native multi-task model achieves accuracy comparable to the single-task models for the FDI attack detection and localization. However, for the compensation task, the native multi-task module significantly outperforms the single-task baselines. This improvement highlights the advantages of the multi-task design, which enhances trajectory compensation capabilities for a UAV team while maintaining competitive performance in other tasks.

4.6 A Real-flight Experiment

In this subsection, we present a real-fight experiment using a team of three UAVs controlled by a GCS. A mobile PC served as the GCS is configured with an AMD Ryzen 7 CPU, 16GB of RAM, an NVIDIA GeForce RTX 3060 GPU, and the Windows 11 operating system. The UAV team consists of three DJI Tello Robomaster TT drones. The target trajectory for the real flight is a circular path with a radius of $5m$. We implement an attack module capable of launching both uplink and downlink FDI attacks by tampering with control signals and status data, following the methodologies in [14], [27]. A snapshot of the real-flight experiment is presented in Fig. 18(a), and the compensation performances are shown in Fig. 18(c). These experimental results underscore the effectiveness of our proposed JS in practical systems.

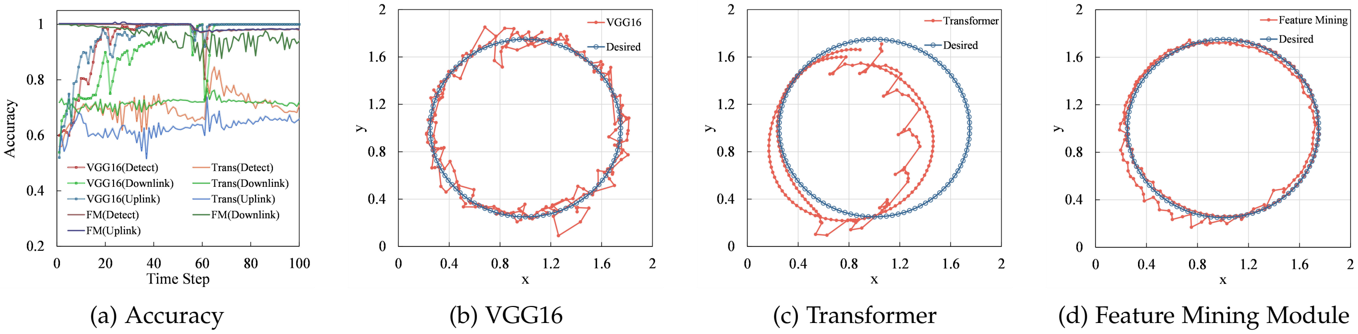
4.7 Discussion

When deploying our JS into large-scale UAV teams, model training becomes a primary challenge that must be addressed in practical scenarios. While we have demonstrated performance improvements and the feasibility of model training for $N = 25$, scaling to team sizes of several hundred or even a few thousand UAVs makes it impractical to train a large JS model on a single GPU. Distributed machine learning, utilizing data parallelism and model parallelism, offers a potential solution for training excessively large models. We leave the investigation of large-scale model training for future work. Another consideration relates to the heterogeneity of UAV devices. Our JS framework relies on the foundational UAV features, such as three-dimensional location information and attitude features, which are commonly available across most UAVs. Therefore, the heterogeneity of UAV teams is unlikely to impact the effectiveness of the proposed scheme in practical applications.

5 Related Work

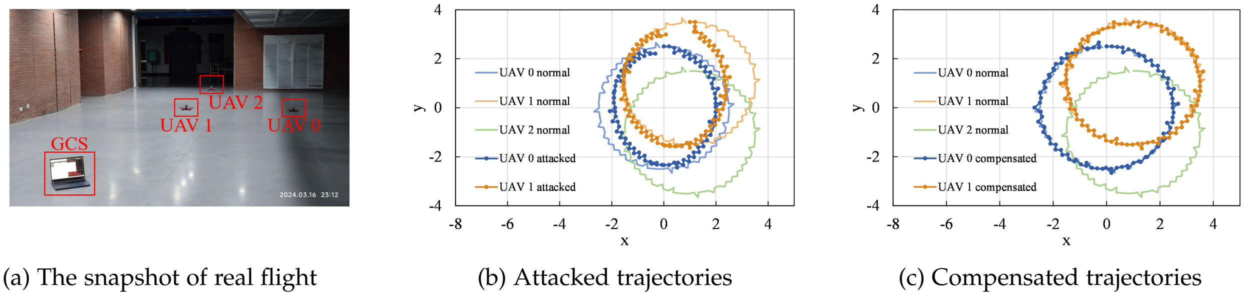
FDI Attacks on UAVs. The proliferation of FDI attacks, which aim to compromise the GCS or individual UAVs, has been increasingly observed across various layers of UAV networks. Some physical-layer attacks include intentional electromagnetic interference [34], [35], [36], Pulse Width Modulation (PWM) [21], etc. For example, Yue et al. employ the intentional electromagnetic interference to disrupt both the uplink and downlink communication simultaneously on a single UAV [34]. Similarly, the work [21] manipulates the pulse width of control signal at actuators to maliciously control the equivalent control force of motors in UAVs. In addition, [36] shows that FDI attacks can compromise GCS by tampering with the current or voltage of UAV sensors, which may expose sensitive information.

At the network layer, some vulnerabilities in protocol such as MAVLink have been exploited to launch FDI attacks successfully [37]. The study [10] reveals that adversaries can use the LabSat3 GPS simulator to perform GPS spoofing and develop a crack SDK to bypass the authentication defense,



(a) Accuracy (b) VGG16 (c) Transformer (d) Feature Mining Module

Fig. 17: Performance of detection, localization, and compensation with different backbones



(a) The snapshot of real flight (b) Attacked trajectories (c) Compensated trajectories

Fig. 18: Real flight trajectories and compensation performance of JS

thereby facilitating FDI attacks. At higher layers, [27], [28], [38] present some impressive attack strategies to generate FDI attack sequences, which can orchestrate global errors and local errors to further bypass bad data detection and other defenses. In summary, FDI attacks have emerged as a significant threat to UAVs, particularly in the context of multi-UAVs teams.

Defenses Against FDI Attacks on UAVs. A series of defense schemes have been proposed to protect UAVs from FDI attacks [39], [40]. A line of studies focus on downlink attacks that compromise status data. Some studies uses extended Kalman filter in BDD, a commonly-used and prevalent defense module in UAVs, to obtain the optimal status estimation and compare it with measured data to detect potential FDI attacks [41], [42]. Similarly, [43] has shown that it is intractable to perform accurate estimation. On the other hand, Luo et al. propose an adaptive Kalman filter by dynamically setting the threshold and decision matrix, showing efficient reduction in detection delay [44]. Another impressive study [45] provides a new method to model the distribution of variables and relax the reversibility requirement of covariance matrix during attack detection based on χ^2. Meanwhile, Lin et al. design a step-by-step estimation framework that initially uses Bayesian estimation to derive two multiplicative variables, which are then applied to Kalman filter to improve the estimation accuracy [46]. When collected data demonstrate high frequency of fluctuation, [47] provides an iterative optimization algorithm for the Kalman filter and computes the norm (e.g., the infinity norm and l_2 norm) of the residual sequence within a given sliding window to alleviate negative impacts of this fluctuation on the FDI attack detection. In addition to variants of BDD, some defense schemes apply encryption techniques like watermarking [48] and coding matrix technique [49] to sensor outputs at UAVs to render compromised status data unacceptable by BDD. Recently, a few studies apply promising deep learning techniques to explore implicit relationships among status data to identify potential attacks and empower the capability of controller to correct decisions under attacks [40], [50].

Another line of security schemes defends against uplink attacks, and they mainly adopt observer-based algorithms to protect a actuator from both external disturbance and potential FDI attacks [25], [51], [52]. For instance, Bai et al. argue that communication delays between GCS and actuator caused by FDI attacks can be detected using delayed physical signals [53]. Gu et al. analyze the average power of received signals and the residual of the authentication signals to identify potential attacks and design attack and disturbance observers to provide compensation to mitigate negative impacts of attacks [11]. Impressively, [14] considers a more complex scenario where both additive attacks and multiplicative attacks coexist in UAVs. The authors design a learning observer for multiplicative attacks, cooperating with an attack observer of additive attacks, to re-enable the compromised UAV to fly along desired trajectories. Recently, bounded additive FDI attacks is studied by [25], and authors propose a fast reactive scheme to defend against attacks in short-term flights. In [54], occasional attacks are studied for controller-to-actuator channel, and authors implement a finite-time sliding-mode control algorithm to counteract these attacks. The work [55] presents an adaptive observer framework where a periodic attack observer collaborates with a bias injection observer to defend against periodic and asymptotic attacks. For the instantaneous increments of time-varying uplink attacks, Dong et al. introduce an additional output estimation feedback and a differential item of estimation error to the observer to improve the estimation and compensation performance [56].

Previous defense studies have primarily considered a single UAV instead of a multi-UAVs team. Additionally, they focus on a single aspect of protection, without the simultaneous consideration of detection, localization, and compensation. Furthermore, prior studies often make an ideal assumption that sensors and actuators are not compromised together, which is impractical and should be relaxed. Finally, few data-driven methods fully utilize temporal-spatial information of UAV flights, but experiences from other domains have witnessed its benefits in improving the performance of deep learning models [19].

Trajectory Planning and Control for UAV. Trajectory planning and real-time control are prevalent and interesting topics that are relevant to our paper. A comprehensive survey [57] systematically summarizes existing results concerning space/aerospace vehicles. For real-time trajectory planning and control, various techniques have been explored, including fuzzy multi-objective transcription optimization [58], highly nonlinear optimization [59], and 3D real-time trajectory optimization based on the pseudo-spectral method [60]. Recently, deep neural networks have been employed to learn hidden patterns between flight states and optimal actions to enhance the capability of predicting next actions [61], [62], [63]. However, these schemes do not consider the existence of FDI attacks which can disrupt their flights. Compared to these studies, our work also enables UAVs to follow target trajectories while considering practical attacks. In addition, we employ under-explored temporal-spatial information to address three interrelated tasks.

Machine Learning-based Schemes for Defending Against FDI Attacks. FDI attacks have been extensively studied across various domains, including smart grid [64] and nuclear systems. Researchers have proposed several machine learning-based methods, such as graph convolutional networks [65], federated learning [66], and artificial neural networks [67], to address the FDI detection problem. However, prior studies have largely overlooked the concept of joint security defense, which integrates detection, localization, and compensation. This integrated approach introduces the challenge of knowledge decay during model training, particularly in long-term flight scenarios. Furthermore, our work assumes a practical and challenging threat model, rendering deep learning models designed for other scenarios unsuitable for UAV applications. Finally, Mao et al. propose an impressive secure model aggregation method for the Byzantine problem in cross-silo federated learning, focusing on robustness and fairness [66], while Tran et al. present another privacy-preserved federated learning, using double-layer encryption to defend against FDI attacks in smart grid [64]. Our proposed method is orthogonal to these secure aggregation solutions.

6 Conclusion

In this paper, we proposed a novel joint secure framework (JS) with multi-task learning to safeguard a multi-UAVs team against FDIA attacks. JS incorporates a feature mining module and a multi-task module to extract spatial-temporal pattern for detecting, localizing, and compensating both uplink and downlink attacks. In addition, we presented an iterative model training method with experience replay to enhance the model's performance under prolonged and continuous attacks. Through extensive experiments and a real flight, we demonstrated that our proposed JS exhibits impressive and superior detection and compensation performance. While we present our JS on a small-scale UAV team, extending our framework to large-scale UAV teams is an intriguing avenue for future work.

References

[1] J. Su, X. Zhu, S. Li, and W. Chen, "Ai meets uavs: a survey on ai empowered uav perception systems for precision agriculture," *Neurocomputing*, vol. 518, no. 7, pp. 242–270, 2023.

[2] A. Gassara and I. Rodriguez, "Describing correct uavs cooperation architectures applied on an anti-terrorism scenario," *Journal of Information Security and Applications*, vol. 58, no. 3, pp. 102775–102776, 2021.

[3] N. Jin, J. Gui, and X. Zhou, "Equalizing service probability in uav-assisted wireless powered mmwave networks for post-disaster rescue," *Computer Networks*, vol. 225, no. 9, pp. 109644–109645, 2023.

[4] Y. Wang, W. Liu, J. Liu, and C. Sun, "Cooperative usv–uav marine search and rescue with visual navigation and reinforcement learning-based control," *ISA Transactions*, vol. 137, no. 4, pp. 222–235, 2023.

[5] A. Mukherjee, S. Misra, A. Sukrutha, and N. S. Raghuwanshi, "Distributed aerial processing for iot-based edge uav swarms in smart farming," *Elsevier Computer Networks*, vol. 167, no. 7, pp. 107038–107039, 2020.

[6] V. Kumar, "Us agricultural drone market: Current analysis and forecast (2019-2025)." Available at https://univdatos.com/report/us-agricultural-drone-market/, 2019.

[7] X. Mu, Z. Gu, and Q. Lu, "Memory-event-triggered consensus control for multi-uav systems against deception attacks," *ISA Transactions*, vol. 139, no. 2, pp. 95–105, 2023.

[8] S. Padhan and A. Turuk, "Design of false data injection attacks in cyber-physical systems," *Information Sciences*, vol. 608, no. 8, pp. 825–843, 2022.

[9] J. Saputro, E. Hartadi, and M. Syahral, "Implementation of gps attacks on dji phantom 3 standard drone as a security vulnerability test," in *Proceedings of the 2020 International Conference on Information Technology, Advanced Mechanical and Electrical Engineering*, pp. 95–100, 2020.

[10] V. Dey, V. Pudi, A. Chattopadhyay, and Y. Elovici, "Security vulnerabilities of unmanned aerial vehicles and countermeasures: an experimental study," in *Proceedings of the 2018 31st international conference on VLSI design*, pp. 398–403, 2018.

[11] Y. Gu, X. Yu, K. Guo, J. Qiao, and L. Guo, "Detection, estimation, and compensation of false data injection attack for uavs," *Information Sciences*, vol. 546, no. 3, pp. 723–741, 2021.

[12] H. Reda, A. Anwar, A. Mahmood, and N. Chilamkurti, "Data-driven approach for state prediction and detection of false data injection attacks in smart grid," *Journal of Modern Power Systems and Clean Energy*, vol. 11, no. 2, pp. 455–467, 2023.

[13] H. Feng, Y. Han, K. Li, F. Si, and Q. Zhao, "Locational detection of the false data injection attacks via semi-supervised multi-label adversarial network," *International Journal of Electrical Power Energy Systems*, vol. 155, no. 9, pp. 109682–109692, 2024.

[14] Y. Gu, K. Guo, L. Guo, J. Qiao, J. Jia, X. Yu, and L. Xie, "An enhanced uav safety control scheme against attacks on desired trajectory," *Aerospace Science and Technology*, vol. 119, no. 10, pp. 107212–107224, 2021.

[15] H. Lin, P. Sun, C. Cai, S. Lu, and H. Liu, "Secure lqg control for a quadrotor under false data injection attacks," *IET Control Theory & Applications*, vol. 16, no. 9, pp. 925–934, 2022.

[16] T. Hickling, N. Aouf, and P. Spencer, "Robust adversarial attacks detection based on explainable deep reinforcement learning for uav guidance and planning," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 10, pp. 4381–4394, 2023.

[17] S. Wu, Y. Li, Z. Wang, Z. Tan, and Q. Pan, "A highly interpretable framework for generic low-cost uav attack detection," *IEEE Sensors Journal*, vol. 23, no. 7, pp. 7288–7300, 2023.

[18] J. H. Hassan, C. Yue, L. Sifan, H. Yulin, W. Juan, and W. Shoufeng, "Real-time collaborative intrusion detection system in uav networks using deep learning," *IEEE Internet of Things Journal*, vol. 11, no. 20, pp. 33371–33391, 2024.

[19] Z. Liu, D. Li, X. Zhang, Z. Zhang, P. Zhang, C. Shan, and J. Han, "Pedestrian attribute recognition via spatio-temporal relationship learning for visual surveillance," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 6, pp. 1–15, 2024.

[20] A. Rugo, C. Ardagna, and N. Ioini, "A security review in the uavnet era: threats, countermeasures, and gap analysis," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–35, 2022.

[21] G. Dayan kl, S. Sinha, D. Muniraj, R. M. Gerdes, M. Farhood, and M. Mina, "Physical-layer attacks against pulse width modulation-controlled actuators," in *Proceedings of the 31st USENIX Security Symposium (USENIX Security 22)*, pp. 953–970, 2022.

[22] A. Barua and M. Faruque, "Sensor security: current progress, research challenges, and future roadmap," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–7, 2022.

[23] W. Xu, Z. Wang, L. Hu, and J. Kurths, "State estimation under joint false data injection attacks: dealing with constraints and insecurity," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 6745–6753, 2022.

[24] H. Yadav and A. Thakkar, "Noa-lstm: An efficient lstm cell architecture for time series forecasting," *Expert Systems with Applications*, vol. 238, no. 7, pp. 122333–122343, 2024.

[25] Y. Gu, K. Guo, C. Zhao, X. Yu, and L. Guo, "Fast reactive mechanism for desired trajectory attacks on unmanned aerial vehicles," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 8, pp. 8976–8984, 2023.

[26] L. Li, H. Yang, Y. Xia, and L. Dai, "Distributed filtering for nonlinear systems under false data injection attack," *Automatica*, vol. 145, no. 3, pp. 110521–110431, 2022.

[27] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *Proceedings of the 1st Workshop Secure Control System*, pp. 1–6, 2010.

[28] A. Baniamerian, K. Khorasani, and N. Meskin, "Monitoring and detection of malicious adversarial zero dynamics attacks in cyber-physical systems," in *Proceedings of the 2020 IEEE Conference on Control Technology and Applications*, pp. 726–731, 2020.

[29] C. Sangwin, *On Heron's formula for the area of a plane triangle*. The College Mathematics Journal, 2023.

[30] K. Guo, N. Wang, D. Liu, and X. Peng, "Uncertainty-aware lstm based dynamic flight fault detection for uav actuator," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, no. 6, pp. 1–13, 2023.

[31] J. Galvan, A. Raja, Y. Li, and J. Yuan, "Sensor data-driven uav anomaly detection using deep learning approach," in *Proceedings of the MILCOM 2021-2021 IEEE Military Communications Conference*, pp. 589–594, 2021.

[32] F. Tlili, S. Ayed, and L. Fourati, "A new hybrid adaptive deep learning-based framework for uavs faults and attacks detection," *IEEE Transactions on Services Computing*, vol. 16, no. 6, pp. 4128–4139, 2023.

[33] H. Slimane, S. Benouadah, K. Shamaileh, V. Devabhaktuni, and N. Kaabouch, "Ads-b message injection attack on uavs: assessment of svm-based detection techniques," in *Proceedings of the 2022 IEEE International Conference on Electro Information Technology*, pp. 405–410, 2022.

[34] W. Yue, Y. Zhao, and L. Wang, "Adaptive tracking control of quadrotor uav with electromagnetic attacks," in *Proceedings of the 2022 13th Asian Control Conference*, pp. 167–172, 2022.

[35] D. Gokcen, S. Sourav, M. Devaprakash, G. Ryan, F. Mazen, and M. Mani, "Physical layer attacks against pulse width modulation-controlled actuators," in *Proceedings of the 31st USENIX Security Symposium (USENIX Security 22)*, pp. 953–970, 2022.

[36] Y. Zhang and K. Rasmussen, "Detection of electromagnetic interference attacks on sensor systems," in *Proceedings of the 2020 IEEE Symposium on Security and Privacy (S&P)*, pp. 203–216, 2020.

[37] P. Pekarčík, E. Chovancová, M. Havrilla, and M. Hasin, "Security analysis of attacks on uav," in *Proceedings of the 2023 IEEE 21st World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 57–62, 2023.

[38] Z. Zhao, Y. Xu, Y. Li, Z. Zhen, Y. Yang, and Y. Shi, "Data-driven attack detection and identification for cyber-physical systems under sparse sensor attacks," *IEEE Transactions on Automatic Control*, vol. 68, no. 10, pp. 6330–6337, 2023.

[39] S. C. Hassler, U. A. Mughal, and M. Ismail, "Cyber-physical intrusion detection system for unmanned aerial vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1–12, 2023.

[40] U. Mughal, S. Hassler, and M. Ismail, "Machine learning-based intrusion detection for swarm of unmanned aerial vehicles," in *Proceedings of the 2023 IEEE Conference on Communications and Network Security*, pp. 1–9, 2023.

[41] J. Hua and H. Fei, "Fusion and detection for multi-sensor systems under false data injection attacks," *ISA Transactions*, vol. 132, no. 13, pp. 222–234, 2023.

[42] G. Zhang, W. Gao, Y. Li, X. Guo, P. Hu, and J. Zhu, "Detection of false data injection attacks in a smart grid based on wls and an adaptive interpolation extended kalman filter," *Energies*, vol. 16, no. 20, pp. 7203–7213, 2023.

[43] Y. Li, Y. Yang, T. Chai, and T. Chen, "Stochastic detection against deception attacks in cps: performance evaluation and game-theoretic analysis," *Automatica*, vol. 144, no. 8, pp. 110461–110472, 2022.

[44] X. Luo, M. Zhu, X. Wang, and X. Guan, "Detection and isolation of false data injection attack via adaptive kalman filter bank," *Journal of Control and Decision*, vol. 11, no. 1, pp. 60–72, 2022.

[45] G. Chen, Y. Zhang, S. Gu, and W. Hu, "Resilient state estimation and control of cyber-physical systems against false data injection attacks on both actuator and sensors," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 500–510, 2021.

[46] H. Lin, J. Lam, and Z. Wang, "Secure state estimation for systems under mixed cyber-attacks: security and performance analysis," *Information Sciences*, vol. 546, no. 4, pp. 943–960, 2021.

[47] J. Xiao and M. Feroskhan, "Cyber attack detection and isolation for a quadrotor uav with modified sliding innovation sequences," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7202–7214, 2022.

[48] A. Naha, A. Teixeira, A. Ahlén, and S. Dey, "Quickest physical watermarking-based detection of measurement replacement attacks in networked control systems," *European Journal of Control*, vol. 71, no. 5, pp. 100804–100804, 2023.

[49] X. Zhang and Z. Xi, "Coding matrix based detection of false data injection attack for coulomb satellite formation systems with lqg control," in *Proceedings of the 2022 IEEE 17th Conference on Industrial Electronics and Applications*, pp. 1200–1207, 2022.

[50] M. Mazare, "Reinforcement learning-based fixed-time resilient control of nonlinear cyber physical systems under false data injection attacks and mismatch disturbances," *Journal of the Franklin Institute*, vol. 360, no. 18, pp. 14926–14938, 2023.

[51] B. Han, J. Jiang, and C. Yu, "Fuzzy adaptive observer–based resilient formation control for heterogeneous multiple unmanned aerial vehicles with false data injection attacks and prescribed performance," *Transactions of the Institute of Measurement and Control*, vol. 45, no. 6, pp. 1021–1036, 2023.

[52] M. Taheri, K. Khorasani, I. Shames, and N. Meskin, "Cyberattack and machine-induced fault detection and isolation methodologies for cyber-physical systems," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 2, pp. 502–517, 2024.

[53] P. Bai, H. Zhang, J. Zhang, and H. Li, "Usv control with adaptive compensation under false data injection attacks," in *Proceedings of the IEEE INFOCOM 2022*, pp. 1–2, 2022.

[54] Z. Cao, Y. Niu, and J. Song, "Finite-time sliding-mode control of markovian jump cyber-physical systems against randomly occurring injection attacks," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1264–1271, 2019.

[55] L. Dong, H. Xu, X. Wei, and X. Hu, "Security correction control of stochastic cyber–physical systems subject to false data injection attacks with heterogeneous effects," *ISA transactions*, vol. 123, no. 11, pp. 1–13, 2022.

[56] L. Dong, H. Xu, L. Zhang, Z. Li, and Y. Chen, "Adjustable proportional-integral multivariable observer-based fdi attack dynamic reconstitution and secure control for cyber-physical systems," *Applied Mathematics and Computation*, vol. 443, no. 1, pp. 127762–127774, 2023.

[57] R. Chai, K. Chen, L. Cui, S. Chai, G. Inalhan, and A. Tsourdos, *Review of advanced guidance and control methods*. Springer Nature Singapore, 2023.

[58] D. Mishra and G. Sushnigdha, "A novel re-entry trajectory design

strategy enforcing inequality and terminal constraints in height-velocity plane," *Advances in Space Research*, 2023.

[59] R. Zhang and N. Cui, "Entry trajectory optimization with general polygonal no-fly zone constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 6, pp. 9205–9218, 2023.

[60] J. Wang, Y. Wu, M. Liu, M. Yang, and H. Liang, "A real-time trajectory optimization method for hypersonic vehicles based on a deep neural network," *Aerospace*, vol. 9, no. 4, pp. 188–201, 2022.

[61] P. Dai, D. Feng, W. Feng, J. Cui, and L. Zhang, "Entry trajectory optimization for hypersonic vehicles based on convex programming and neural network," *Aerospace Science and Technology*, vol. 137, no. 7, pp. 108259–108271, 2023.

[62] Y. Liu, H. Wang, T. Wu, Y. Lun, J. Fan, and J. Wu, "Attitude control for hypersonic reentry vehicles: an efficient deep reinforcement learning method," *Applied Soft Computing*, vol. 123, no. 3, pp. 108865–108875, 2022.

[63] J. Kim, C. Justin, D. Mavris, and S. Briceno, "Data-driven approach using machine learning for real-time flight path optimization," *Journal of Aerospace Information Systems*, vol. 19, no. 1, pp. 3–21, 2022.

[64] H.-Y. Tran, J. Hu, X. Yin, and H. R. Pota, "An efficient privacy-enhancing cross-silo federated learning and applications for false data injection attack detection in smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2538–2552, 2023.

[65] E. Vincent, M. Korki, M. Seyedmahmoudian, A. Stojcevski, and S. Mekhilef, "Detection of false data injection attacks in cyber-physical systems using graph convolutional network," *Elsevier Electric Power Systems Research*, vol. 217, p. 109118, 2023.

[66] Y. Mao, Z. Ye, X. Yuan, and S. Zhong, "Secure model aggregation against poisoning attacks for cross-silo federated learning with robustness and fairness," *IEEE Transactions on Information Forensics and Security*, pp. 6321–633, 2024.

[67] M. Rahman, J. Yan, and E. T. Fapi, "Adversarial artificial intelligence in blind false data injection in smart grid ac state estimation," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 6, pp. 8873–8883, 2024.

Rongfei Zeng received the B.Sc. degree in computer science from the Northeastern University, Shenyang, China, in 2006, and the Ph.D. degree in computer science with honors from Tsinghua University, Beijing, China, in 2012. Currently, he is an Associate Professor at the College of Software, Northeastern University. His research interests include federated learning, distributed machine learning, and network security.

Chenyang Jiang received the B.Sc. degree in software engineering from Northeastern University, Shenyang, China, in 2021. Now, he is pursuing his M.Sc. degree in software engineering at Northeastern University. His research interests include the security in cyber-physical system and UAVs.

Xingwei Wang received the B.Sc., M.Sc., and Ph. D. degrees in computer science from the Northeastern University, Shenyang, China, in 1989, 1992, and 1998, respectively. He is currently a Professor of Northeastern University, China. His research interests include network security and future Internet. He has published over 100 journal articles, books, and conference papers.

Baochun Li (Fellow, IEEE) received the B.Eng. degree from Tsinghua University in 1995 and the M.S. and Ph.D. degrees from the University of Illinois at Urbana–Champaign in 1997 and 2000, respectively. Since 2000, he has been with the Department of Electrical and Computer Engineering, University of Toronto, where he is currently a Professor. Since August 2005, he has been with the Bell Canada Endowed Chair in computer engineering. His current research interests include cloud computing, security and privacy, distributed machine learning, federated learning, and networking. He is a Fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada. He was a recipient of the IEEE Communications Society Leonard G. Abraham Award in the Field of Communications Systems in 2000, the Multimedia Communications Best Paper Award from the IEEE Communications Society in 2009, the University of Toronto McLean Award in 2009, the Best Paper Award from IEEE INFOCOM in 2023, and the IEEE INFOCOM Achievement Award in 2024.