Physics-Informed Anomaly Detection for Unmanned Aerial Vehicles

Yifan Guo, Kartik A. Pant, *Graduate Student Member, IEEE*, and Inseok Hwang, *Member, IEEE*

***Abstract*—The secure operation of Unmanned Aerial Vehicle (UAV) systems requires assurance of safety and reliability. However, their security aspect can be challenged by various unexpected anomalies, which can significantly compromise a UAV's performance or lead to catastrophic failures. The UAV's runtime safety assurance in dynamic environments remains an open problem. To address this challenge, we propose a novel physics-informed anomaly detection framework called Physics-informed Prediction for Detection (PIPD) and instantiate the framework into two specific models, PIPDall and PIPDres. The proposed framework efficiently integrates the knowledge of physical dynamics into a deep anomaly detection model, exhibiting significantly better detection accuracies and a superior generalization ability to unseen anomalies. Unlike typical physics-informed frameworks, which often sharpen the loss landscape, our framework further smoothens it, thereby facilitating an efficient training process. We validate our framework using emulated Global Navigation Satellite System (GNSS) spoofing attacks with linear and sinusoidal profiles, which are covered under different noise levels. The simulation results present a performance improvement of up to 17.77% in the ROC-AUC score of our models compared to the baselines. Through real-world experiments, we confirm the effectiveness of our framework against multi-noise GNSS spoofing attacks and extend our validation to other complex scenarios, including aperiodic GNSS attacks and wind disturbance.**

***Index Terms*—Anomaly detection, deep learning.**

I. Introduction

UNMANNED Aerial Vehicles (UAVs) are rapidly developing in recent years and have been wildly applied for precision agriculture [1], surveillance [2], etc. UAVs are required to work in crowded regions and carry valuable cargo in dense urban environments, making their operational security a crucial part of the mission. However, the reliability and security of UAVs are challenged by various anomalies such as cyberattacks, system faults, and extreme weather conditions [3]. These anomalies can significantly compromise the performance of the UAV system or even lead to catastrophic results. Thus, a reliable and robust anomaly detection mechanism is essential for safe operations of UAVs in urban environments.

Recently, many deep learning (DL)-based anomaly detection algorithms have been proposed to address the challenge of UAV security, leveraging deep models' strong generalization and automatic feature engineering capabilities. DL models can detect both statistical shifts and suspicious signal patterns in a unified way. Moreover, once deployed, it is challenging to design adversarial attacks against a deep anomaly detector without directly accessing its parameters. However, existing DL anomaly detection models face significant limitations. First, most existing works require the dominant portion of the samples in the dataset to be nominal [4]. Anomaly samples in the dataset are "pollutions" since they mix with nominal data and deteriorate the modeling of nominal behaviors. Second, deep anomaly detection algorithms usually need large amounts of data to model a UAV's behavior in various environments. This requires extensive flight data from real-world experiments. Finally, most existing methods lack an efficient way to incorporate domain knowledge, which is often readily available for dynamic systems like UAVs. Physical knowledge of the system can significantly improve the robustness of the learned model and its generalization ability [5].

To handle these challenges, we propose a physics-informed, semi-supervised deep anomaly detection framework, which we name Physics-informed Prediction for Detection (PIPD). The framework is further instantiated into two specific models, PIPDall and PIPDres. Our PIPD framework first embeds the input features into a latent space with an encoder and uses a predictor to forecast the dynamics/observables of the system, given the embedding. The physics-informed constraint is then added as a Mean Squared Error (MSE) term in the final loss to penalize any forecasting error, instead of constraining the model with the exact nonlinear differential equations of dynamics. This provides a straightforward yet efficient approach to inject physical information into the latent space without complicating the training process or the loss landscape. We observe that the landscape becomes more convex and smooth around the converged local optimum with the physics-informed term in the final loss function. Through GNSS spoofing attack simulations, we empirically demonstrate that the knowledge of system dynamics significantly increases the models' generalization ability and robustness, especially when the test attack profile is unseen during the training process. The proposed method presents up to 17.77% improvement of ROC-AUC score over the baselines in simulations. Finally, we conduct real-world experiments with a Crazyflie drone to further validate our proposed PIPD framework against multi-noise attacks, aperiodic attacks, and wind disturbance. Our PIPD models

Received 24 March 2025; accepted 25 July 2025. Date of publication 13 August 2025; date of current version 21 August 2025. This article was recommended for publication by Associate Editor N. Jagannatha Sanket and Editor M. Vincze upon evaluation of the reviewers' comments. *(Corresponding author: Yifan Guo.)*

The authors are with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47906 USA (e-mail: guo781@purdue.edu; kpant@purdue.edu; ihwang@purdue.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3598666, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3598666

2377-3766 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies.
Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Purdue University. Downloaded on September 29,2025 at 02:28:00 UTC from IEEE Xplore. Restrictions apply.

yield more accurate detection in most cases and perform more consistently compared with the baselines. It is worth noting that the proposed framework is potentially applicable to any anomalies that cause discrepancies in the correlation between the control inputs, measurements, and state estimations of a dynamic system. In summary, the main contributions of this letter include:

1) We propose an efficient framework to "softly" inject physical knowledge into a DL anomaly detector. The framework is instantiated into specific models, which we refer to as PIPDall and PIPDres.
2) We validate the effectiveness of the proposed PIPD models using emulated GNSS spoofing attacks against a single UAV. With Gazebo [6] simulation data, we demonstrate that injected knowledge of UAV dynamics helps our models to better detect both seen and unseen attacks. We also observe that our PIPD framework can more efficiently exploit the small amount of labeled data. The proposed models achieve an ROC-AUC score of up to 17.77% better than the baselines.
3) We directly observe the loss landscapes and show that our PIPD framework smoothens the loss landscape. This further facilitates the training process and enhances the model's generalization ability.
4) Finally, we validate the effectiveness of the proposed PIPD models with real-world Crazyflie experiments [7] with multi-noise attacks, aperiodic attacks, and wind disturbance. Our PIPD models detect anomalies more accurately and perform more consistently than the baselines through all experiments.

The rest of the letter is organized as follows. In Section II, we briefly review related works on DL-based anomaly detection. Section III describes the proposed framework and models in detail. Section IV and Section VI validate the proposed method with simulation and real-world experiments, respectively. Finally, Section VII concludes the letter.

II. Deep Learning-Based Anomaly Detection

Anomaly detection is crucial to ensure the safety and reliability of safety-critical systems. Traditional anomaly detection methods struggle with increasing system complexity and sophisticated attacks [8]. Many research studies in the literature have explored the power of deep learning to handle these challenges. These can be categorized into three classes [8]: (i) *Deep learning for feature extraction*, where deep models are utilized to extract low-dimensional features from high-dimensional data for downstream anomaly detection [9]; (ii) *Learning feature representations of normality*, where feature learning is integrated with anomaly scoring, rather than fully decoupling these two modules. Representative approaches include data reconstruction [10], generative modeling [11], [12], predictability modeling [13], [14], and self-supervised classification [4], [15], [16]; and (iii) *End-to-end anomaly score learning*, where an scalar anomaly score is directly learned in an end-to-end fashion [17], [18]. Most deep models for anomaly detection operate in an unsupervised or self-supervised setting. This relieves the demand for expensive expert labels, but the model loses its ability to utilize precious labels when available. Some works craft their algorithms in a semi-supervised learning environment, providing an efficient way to incorporate label information and making anomaly data samples beneficial for model training [15], [19], [20].

Despite their excellent performance on benchmark datasets or computer vision applications, most existing deep anomaly detection algorithms cannot be trivially extended to dynamical systems such as UAVs, as discussed in the introduction. To this end, the idea of physics-informed machine learning (PIML) has been explored, where the dataset, network structure, and/or loss function are designed based on prior physical knowledge. The physics-informed anomaly detection method has been used primarily for power grids. In [21], a multi-target multivariate regression model is used to predict electrical power grid measurements, where a False Data Injection (FDI) attack is claimed if the difference between predicted measurements and real measurements is above an adaptive threshold, which is calculated from the system dynamics. The authors in [22] proposed a hybrid framework that combines a graph convolutional network with spectral analysis, where spectral analysis generates an association graph based on measurement similarity, and the network uses this graph to detect anomalies.

These methods cannot be readily extended to the UAV anomaly detection scenario, since they rely on the network topology and neighbors' information in power grids. In addition, UAVs face several unique challenges, such as multipath effects of GNSS signals in urban environments, variation in operating environment, etc. The PIML solution for UAV anomaly detection is underexplored. In the pioneer work [23], a single-layer NN is used for state estimation, where the parameters of the NN are updated based on the Extended Kalman Filter (EKF) algorithm. However, their approach is exclusive to the single-layer NN and cannot be trivially extended to deep models. In a recent work [24], PIML is used to predict UAV trajectories and to infer potential malicious intentions. A physical model is used to validate and stabilize the trajectory predictions from a deep multi-expert module. However, their work focuses on detecting suspicious UAV trajectories instead of their internal anomalies. The highly nonlinear dynamics of UAVs also challenge existing PIML anomaly detection algorithms. The typical approach to physically inform an anomaly detection model is to constrain the output of the network to conform to the differential equation of system dynamics. As pointed out in [25], this approach can lead to a sharp loss landscape for non-trivial system dynamics, resulting in a huge performance deterioration against a small distribution shift in data. In the proposed framework, the constraint of system dynamics is posed as an MSE term in the loss function, which even makes the landscape more convex and smoother. In summary, our work provides an efficient and effective way to incorporate the dynamics of complex physical systems for anomaly detection.

III. Methodology

In this section, we first formulate the anomaly detection problem for UAVs. Then, we introduce the proposed framework and models.

Authorized licensed use limited to: Purdue University. Downloaded on September 29,2025 at 02:28:00 UTC from IEEE Xplore. Restrictions apply.

A. Problem Formulation

We first formulate the anomaly detection problem for a UAV. Let $\mathbf{s} \in \mathbb{R}^{n_s}$, $\mathbf{o} \in \mathbb{R}^{n_o}$, and $\mathbf{u} \in \mathbb{R}^{n_u}$ denote the states, measurements, and control inputs, respectively. n_s, n_o, and n_u are the dimensions of internal states, outputs, and control inputs of the system, respectively. Using $f(\cdot)$ to represent the state transition dynamics and $g(\cdot)$ for the observation process, the UAV discrete dynamics can be written as:

$$\mathbf{s}_{t+1} = \mathbf{s}_t + \Delta t \cdot f(\mathbf{u}_t, \mathbf{s}_t), \tag{1a}$$

$$\mathbf{o}_{t+1} = g(\mathbf{u}_t, \mathbf{s}_t), \tag{1b}$$

where Δt represents the system's sampling period and subscript t is the discrete time step. The internal state $\mathbf{s}_t$ cannot typically be observed directly and is estimated using a state estimator (e.g., Kalman filter). The control inputs $\mathbf{u}_t$ are usually calculated by an automatic controller given the reference state $\bar{\mathbf{s}}_t$ and the estimated state $\hat{\mathbf{s}}_t$. Formally:

$$\begin{aligned} \hat{\mathbf{s}}_t &= h(\mathbf{u}_{t-1}, \hat{\mathbf{s}}_{t-1}, \mathbf{o}_t), \\ \mathbf{u}_t &= c(\mathbf{o}_t, \hat{\mathbf{s}}_t, \bar{\mathbf{s}}_t), \end{aligned} \tag{2}$$

where $h(\cdot): \mathbb{R}^{n_u} \times \mathbb{R}^{n_s} \times \mathbb{R}^{n_o} \rightarrow R^{n_s}$ is the state estimator and $c(\cdot): \mathbb{R}^{n_u} \times \mathbb{R}^{n_s} \times \mathbb{R}^{n_s} \rightarrow R^{n_u}$ is the automatic controller. The state estimator also provides a residual signal $\mathbf{r} \in \mathbb{R}^{n_s}$, representing the uncertainty of the estimates.

We consider three representative anomaly conditions in this work: multi-noise GNSS spoofing attack, aperiodic GNSS spoofing attack, and wind disturbance. GNSS spoofing attacks modify position measurements of a UAV, where the measurements with the anomaly can be represented as:

$$\mathbf{o}_t^a = \mathbf{o}_t + \mathbf{a}_t + \mathbf{n}_t, \quad \mathbf{n}_t \sim \mathcal{N}(\boldsymbol{\mu}_o, \boldsymbol{\Sigma}_o), \tag{3}$$

$\mathbf{o}_t^a$ is the attacked measurements and $\mathbf{a}_t$ is the attack signal. $\mathbf{n}_t$ is the measurement noise with Gaussian distribution $\mathcal{N}$ whose mean and covariance matrix are $\boldsymbol{\mu}_o$ and $\boldsymbol{\Sigma}_o$, respectively. We design attack signals to induce cumulative errors while ensuring that the residuals remain below the detection threshold of the onboard anomaly detector, so that they cannot be detected.

According to [26], the impact of wind disturbance can be formulated as an additive component to a UAV's acceleration, i.e.,

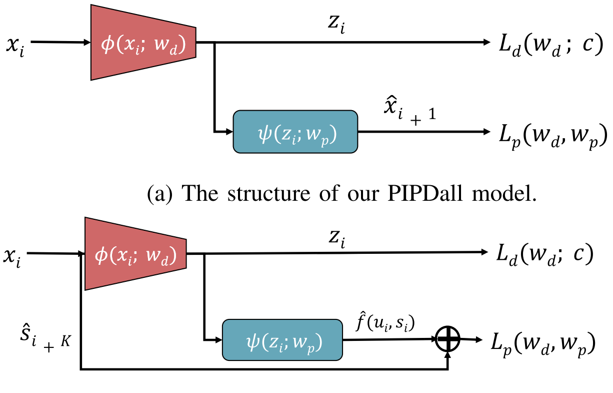
$$\boldsymbol{\delta}_t^a = \boldsymbol{\delta}_t + \boldsymbol{\xi}_t, \quad \boldsymbol{\xi}_t \sim \mathcal{N}(\boldsymbol{\mu}_\xi, \boldsymbol{\Sigma}_\xi), \tag{4}$$

where δ is the nominal acceleration, ξ is the acceleration caused by wind disturbance, which is subject to a normal distribution with mean value $\boldsymbol{\mu}_\xi$ and covariance matrix Σ_ξ.

B. PIPD Framework

We now explain the details of the proposed PIPD framework. The overall framework consists of two branches: the detection branch and the prediction branch. Formally, the framework can be written as:

$$\begin{aligned} \mathbf{z}_i &= \phi(\mathbf{x}_i; \mathbf{w}_d), \\ y_i &= \xi(\mathbf{z}_i), \end{aligned}$$



(a) The structure of our PIPDall model.

(b) The structure of our PIPDres model.

Fig. 1. The network structures of the proposed PIPD models. The PIPDall model predicts all the entries at the next time step. The PIPDres model predicts only the future internal states.

$$f(\cdot) \approx \phi(\mathbf{x}_i; \mathbf{w}_d) \circ \psi(\mathbf{z}_i, \mathbf{w}_p), \tag{5}$$

where $\circ$ means function composition, and $\mathbf{w}_d$ and $\mathbf{w}_p$ are trainable parameters. The detection branch only contains the encoder $\phi(\mathbf{x}_i; \mathbf{w}_d)$, which embeds a sample $\mathbf{x}_i$ in a latent space, obtaining the embedded representation $\mathbf{z}_i$. A decision function $\xi(\mathbf{z}_i)$ decides the flag y given the latent representation. The prediction branch contains the encoder $\phi(\mathbf{x}; \mathbf{w}_d)$ and the predictor $\psi(\mathbf{z}_i, \mathbf{w}_p)$. This branch embeds the physical dynamics into the latent space by predicting information of the next time step. The combination of the encoder and the predictor approximates the system dynamics. The overall loss function is defined as $L := L_d + \gamma L_p$, where γ is a hyperparameter, $L_d(\mathbf{w}_d)$ and $L_p(\mathbf{w}_d, \mathbf{w}_p)$ are the loss function for detection and prediction, respectively. This is a "soft" way to inform the model since we do not directly constrain the output of the neural network with the dynamics. Instead, we guide the latent space to embed the system dynamics by predicting the next step information from the latent variable.

C. Specific PIPD Models

Now, we introduce our specific PIPD models starting with the semi-supervised setting. To train a PIPD model, a labeled dataset $\tilde{\mathbf{X}} = \{(\tilde{\mathbf{x}}_0, \tilde{y}_0), \ldots, (\tilde{\mathbf{x}}_{N_l}, \tilde{y}_{N_l})\}$ and an unlabeled dataset $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N_u}\}$ are used. Here, $\mathbf{x}_i(\text{or } \tilde{\mathbf{x}}_i) = [\bar{\mathbf{s}}_{i-1:i+K-1}, \mathbf{o}^a_{i:i+K}, \hat{\mathbf{s}}_{i:i+K}, \mathbf{r}_{i:i+K}]^\intercal$, where K is the constant length of the observation window. We do not include control inputs to our input features based on the model's performance during the hyperparameter tuning process. The labels $\tilde{y}_i \in \{1, -1\}$ indicate whether the sample is nominal ($\tilde{y}_i = 1$) or anomaly ($\tilde{y}_i = -1$). The majority of the dataset is assumed to be unlabeled samples.

The detailed structures of our PIPDall and PIPDres models are shown in Fig. 1. The two models share the same detection branch which examines each sample with a hypersphere in the latent space with a centroid $\mathbf{c} \neq \mathbf{0}$ and a decision boundary of radius R. The training objective of this branch is to embed all unlabeled and nominal samples as close to the centroid as possible while

Authorized licensed use limited to: Purdue University. Downloaded on September 29,2025 at 02:28:00 UTC from IEEE Xplore. Restrictions apply.

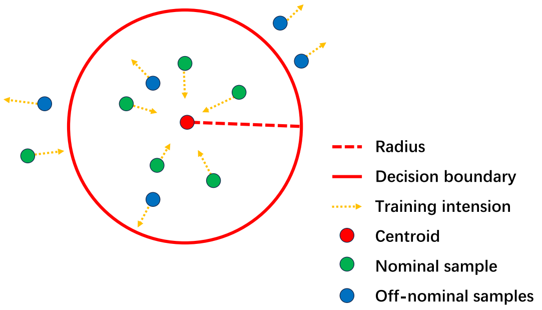


Fig. 2. Demonstration of the detection and the training process. Detection is based on the hypersphere defined by the centroid and the radius. Nominal samples are embedded closer to the centroid, while anomaly samples are pushed away.

pushing all anomaly samples away from the centroid, as shown in Fig. 2. The detection branch must have no bias terms and only use bounded activation functions to avoid a trivial solution, where all samples are embedded to the centroid [4]. The training objective of the detection branch is:

$$L_d = \frac{1}{N_u + N_l}\sum_{i=1}^{N} \|\phi(\mathbf{x}_i;\mathbf{w}_d) - \mathbf{c}\|^2 + \frac{\lambda}{2}\|\mathbf{w}_d\|_2^2 + \frac{\eta}{N_u + N_l}\sum_{j=1}^{M}\left(\|\phi(\tilde{\mathbf{x}}_j;\mathbf{w}_d) - \mathbf{c}\|^2\right)^{\tilde{y}_j}. \tag{6}$$

Here, λ and η control the weight of regularization and expert labels, respectively. N_u and N_l represent the numbers of unlabeled and labeled samples in the training set, respectively. The decision function $\xi(\mathbf{z}_i)$ is in the form of:

$$\xi(\mathbf{z}_i) = \mathbb{1}(\|\mathbf{z}_i - \mathbf{c}\| < R), \tag{7}$$

where $\mathbb{1}$ is the indicator function whose output follows our definition of $\tilde{y}$. We calculate $\mathbf{c}$ by averaging all the $\mathbf{z_i}$ for the training set before training.R is the largest radius with the false positive rate 0 on the ROC curve of the training set.

The PIPDall model considers an "augmented dynamics", which means that the prediction branch not only forecasts the future of the UAV's state, but also the measurements and reference trajectory at the next time step. That is, the predictor $\psi(\mathbf{z}_i, \mathbf{w}_p)$ predicts $\mathbf{x}_{i+1}^0 := [\bar{\mathbf{s}}_{i+K}, \mathbf{o}_{i+K+1}, \hat{\mathbf{s}}_{i+K+1}, \mathbf{r}_{i+K+1}]^\intercal$, i.e., the first element in the next sample. This requires $\phi(\cdot)$ and $\psi(\cdot)$ to jointly model the controller onboard, the dynamics of the UAV, the state estimator and the observation dynamics. The corresponding loss function is

$$L_p = \mathbf{MSE}(\mathbf{x}_{i+1}^0, \psi(\mathbf{z}_i;\mathbf{w}_p)). \tag{8}$$

In the PIPDres model, the prediction branch focuses on the UAV dynamics, predicting estimated states $\hat{\mathbf{s}}_{i+K+1}$. The prediction branch is also trained using the mean square error (MSE) loss function:

$$L_p = \mathbf{MSE}(\hat{\mathbf{s}}_{i+K+1}, \psi(\mathbf{z}_i;\mathbf{w}_p)). \tag{9}$$

Next, we present the simulation results to demonstrate how the knowledge of system dynamics improve the performance of the anomaly detector.

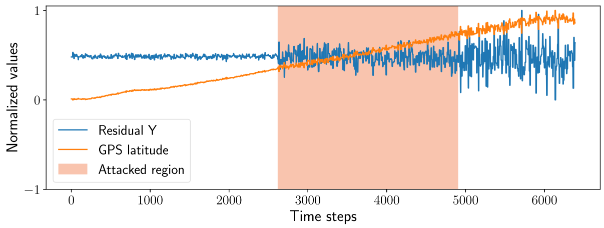


Fig. 3. The log of a simulated flight. We only present the residual on the Y direction and GNSS latitude for clarity of visualization. Magnitudes of the residual and the GNSS measurements are normalized.

IV. Simulation Results

In this section, we validate the effectiveness of our PIPD models with emulated multi-noise GNSS spoofing attacks in Gazebo simulations. GNSS spoofing attacks involve manipulating a victim's received GNSS signals to cause errors in position calculation or system time [27]. GNSS spoofing is a representative case of UAV anomaly since it is the most common cyberattack against UAVs and can be designed in a stealthy manner [3].

A. Simulation Setting

In this section, we introduce the specific simulation setting. Generating actual spoofing signals presents several logistical challenges, such as having a facility with anechoic chamber, specialized radio hardware, and permission from the authorities. In this work, we instead emulate the effect of actual GNSS spoofing attacks on GNSS measurements using a sensor emulation framework developed in [28]. Following the real GNSS spoofing attack experiment in [29], we abstract two representative profiles of the attack signal, namely linear attacks $a_t = \alpha(t - t_s)$ and sinusoid attacks $a_t = A\sin(\omega(t - t_s))$, where t is current time, t_s is the start time, α and A control the magnitude of attacks, and ω is the frequency of sinusoid attack. The attack patterns proposed here reflect realistic GNSS spoofing attack intentions, such as gradually deviating the victim's direction (linear attacks) and making the victim swing in narrow environments (sinusoid attacks). We sample an extensive dataset from Gazebo [6] simulations of a quadrotor flying in an urban environment. A simulated test flight with a linear GNSS spoofing attack is shown in Fig. 3, where the drone flies along the Y direction of the world frame. The blue curve is the onboard residuals in the Y direction, the orange curve is the GNSS latitude measurements, and the red region is the attacked region. The magnitudes of the residual and GNSS measurements are normalized. We plot only one channel of residuals and GNSS measurements for the sake of clarity. Similar patterns can be observed in other channels.

The total dataset contains 30% attacked data points. Each data point is 12 dimensional and contains the reference GNSS coordinates, onboard residuals, estimated position, and GNSS position measurements. We reorganize the dataset by concatenating all $K = 100$ consecutive data points to construct 1200 dimensional samples, where K is defined in Section III. A 1200-dimensional sample is an anomaly if any 12-dimensional point in it is an anomaly. To compare models' performance in different training

Authorized licensed use limited to: Purdue University. Downloaded on September 29,2025 at 02:28:00 UTC from IEEE Xplore. Restrictions apply.

TABLE I
HYPERPARAMETERS

	DeepSAD [15]	PIPDall	PIPDres
# node of $\phi(\cdot)$	(512, 512)	(512, 512)	(256, 256)
# node of $\psi(\cdot)$	/	(512, 512)	(256, 256)
Latent dimension	1024	1024	512
η	5	6.93	9.65
λ	5×10^{-7}	5×10^{-7}	5×10^{-7}
γ	/	9.11	7.34

settings, we construct smaller datasets with various percentages of anomaly and labeled samples by sampling from the overall dataset. Specifically, we determine the number of samples of the constructed dataset as follows:

$$
\begin{aligned}
&\text{Given } p_{la}, p_a, p_{ln}, N_n, \text{ solve } N_{un}, N_{ua}, N_{ln}, N_{la} \\
&\text{s.t.} \quad 0 = (1 - p_{ln})N_{ln} - p_{la}N_{un} - p_{la}N_{ua} + (1 - p_{la})N_{la}, \\
&\qquad 0 = -p_{la}N_{ln} - p_{la}N_{un} - p_{la}N_{ua} + (1 - p_{la})N_{la}, \\
&\qquad 0 = -p_a N_{un} + (1 - p_a)N_{ua}, \\
&\quad N_n = N_{ln} + N_{un}, \qquad (10)
\end{aligned}
$$

where subscript l, u, n, and a stand for "labeled", "unlabeled", "nominal", and "anomaly". p represents "percentage" and N is the number of samples.

We set the variance of nominal GNSS measurement noise to 1, 5, and 10 meters during different stages of flying to imitate the blocking or multipath effects of high-rise buildings in urban environments. This further challenges our model to distinguish the high noise level from attacks. To highlight our approach of informing the deep anomaly detector with the physical dynamics, we choose the DeepSAD model [15] as our main baseline since the structures of the detection branch of our framework is the same as the DeepSAD model. This excludes the performance difference caused by using different models in the detection branch. Comparison with this baseline automatically serves as an ablation study. The encoder $\phi(\cdot)$ and the predictor $\psi(\cdot)$ are multilayer perceptron (MLP) with two hidden layers. Hyperparameters are tuned based on models' ROC-AUC score on a dataset whose $p_a = 0.1$ and $p_{la} = 0.03$. The learning rate starts at 1×10^{-4} and decays logarithmically every 50 epoch. We train the overall model for 300 epochs. The attack parameters are $\alpha = 0.001$, $A = 1$, $\omega = 1$. Other hyperparameters are summarized in Table I. To exclude the factor that different algorithms may have different optimal hyperparameters, all models run with their own empirically optimal hyperparameters. The radius of the decision hypersphere is chosen to be the largest radius with zero false-positive ratio on the ROC curve of the training set. We also choose two other state-of-the-art anomaly detection algorithms—TimesNet [30] and RoSAS [31]—as our baselines for performance comparison. The detailed structures of these two models follow their original papers.

B. Numerical Evaluations

In this section, we compare the performance of the baselines and the proposed PIPD models in various settings.

TABLE II
ROC-AUC SCORES OF MODELS WHEN FACING GNSS SPOOFING ATTACKS

		p_{la}				Gap%
Model	p_a	0	0.1%	0.2%	0.3%	
All	0	0.99/0.92	0.99/0.95	0.99/0.95	0.99/0.96	0.00/1.61
Res	0	1.00/0.94	0.99/0.96	0.99/0.97	0.99/0.96	0.25/2.96
D	0	0.99/0.92	1.00/0.93	0.99/0.94	1.00/0.93	0.51/0.00
T	0	/	/	/	/	/
R	0	/	1.00/0.95	0.98/0.94	0.99/0.97	0.00/2.51
All	2%	0.99/0.88	0.98/0.91	0.99/0.93	1.00/0.93	11.24/2.82
Res	2%	0.98/0.89	0.99/0.94	1.00/0.96	1.00/0.95	11.52/5.35
D	2%	0.98/0.86	0.99/0.89	0.99/0.89	0.95/0.91	9.83/0.00
T	2%	0.89/0.93	/	/	/	0.00/4.79
R	2%	/	0.98/0.94	1.00/0.95	0.99/0.96	11.23/7.04
All	4%	0.97/0.83	0.98/0.86	0.99/0.89	1.00/0.92	10.67/6.71
Res	4%	0.97/0.84	0.99/0.89	0.99/0.96	1.00/0.95	10.96/10.96
D	4%	0.96/0.79	0.98/0.78	0.99/0.84	0.99/0.87	10.11/0.00
T	4%	0.89/0.88	/	/	/	0.00/7.32
R	4%	/	0.99/0.95	0.98/0.93	0.98/0.95	10.49/15.04
All	6%	0.95/0.76	0.97/0.87	0.98/0.88	0.99/0.90	15.77/10.71
Res	6%	0.95/0.82	0.99/0.89	0.99/0.93	0.99/0.90	16.67/14.94
D	6%	0.96/0.72	0.96/0.73	0.97/0.82	1.00/0.81	15.77/0.00
T	6%	0.84/0.84	/	/	/	0.00/9.09
R	6%	/	0.95/0.97	0.99/0.94	0.99/0.97	16.2/24.67
All	8%	0.88/0.68	0.97/0.75	0.97/0.83	0.99/0.88	14.75/0.64
Res	8%	0.96/0.72	0.97/0.84	0.99/0.92	0.99/0.93	17.77/9.29
D	8%	0.89/0.72	0.93/0.77	0.96/0.72	0.98/0.91	13.25/0.00
T	8%	0.83/0.80	/	/	/	0.00/2.56
R	8%	/	0.99/0.93	1.00/0.95	0.98/0.94	19.27/20.51

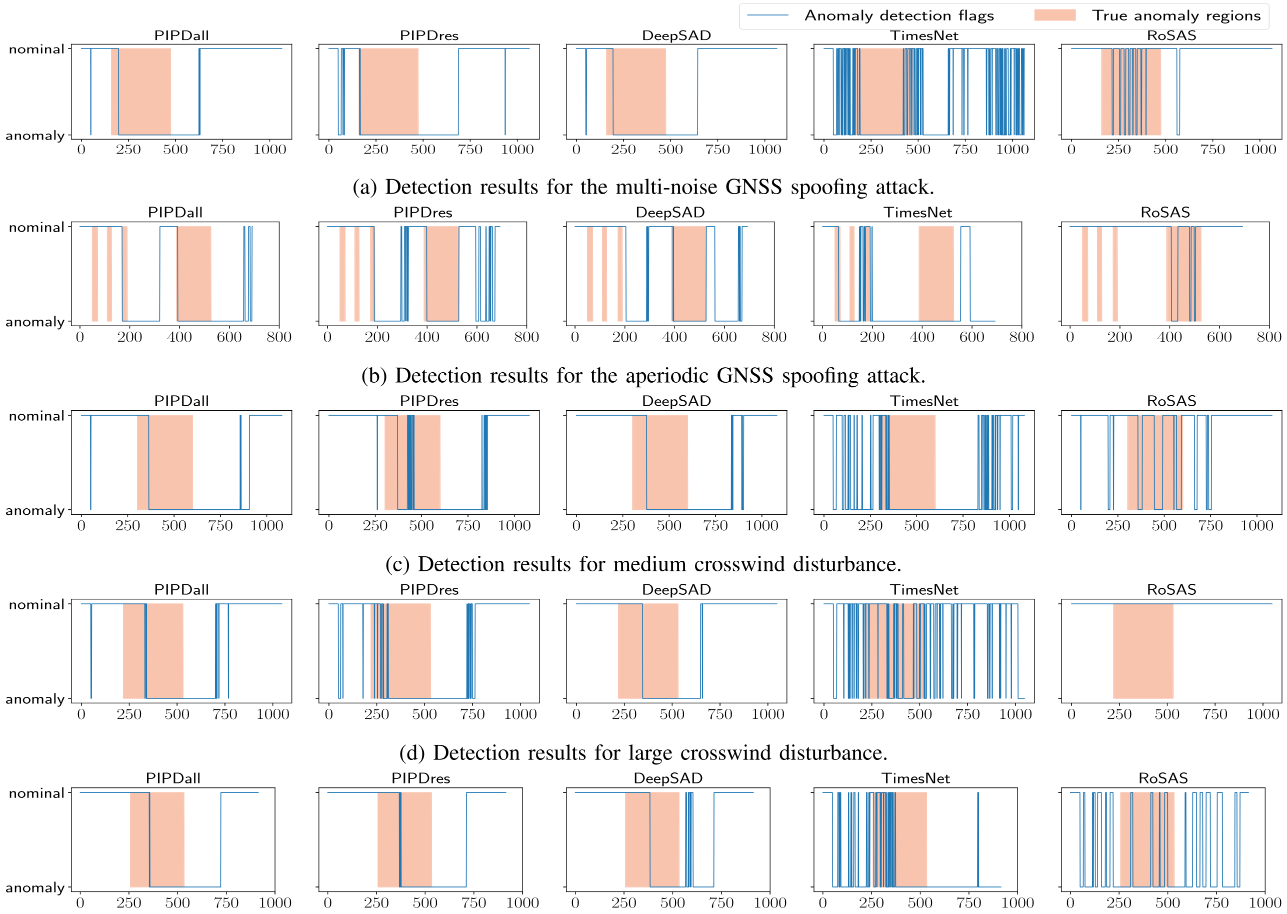
Models are all trained with data containing a linear attack. Each entry is in the form of "score when tested with linear attack/score when tested with sinusoid attack". In the model column, All=PIPDall (ours), Res=PIPDres (ours), D=DeepSAD [15], T=TimesNet [30], R=RoSAS [31].

The training set only contains linear GNSS spoofing attacks, and the test set contains linear attacks or sinusoid attacks. The results are summarized in Table II. For each entry, the number before the slash is the ROC-AUC score when the model is tested with linear attacks, and the number behind is the score when the model is tested with sinusoid attacks. Gaps are between the current model and the worst performance in each setting. The comparison to the DeepSAD [15] is equivalent to an ablation study since the PIPD models are the same as the DeepSAD model except for the physics-informed component. The TimesNet [30] only works in an unsupervised setting, and the RoSAS [31] requires some label anomaly samples to work. That is the reason for the missing entries. When tested with linear attacks, all the models produce satisfactory performance, which gradually degrades as the ratio of pollution increases. However, gaps become significant when the models only observe the linear attack during training and are tested with the sinusoid attack. Our PIPD models almost always produce the best performance until the ratio of pollution reaches 8%. Note that the performance of our PIPD models is better than the ablated baseline DeepSAD [15] in almost all settings. This demonstrates that the injected dynamics knowledge helps anomaly detectors to generalize to unseen attacks.

V. LOSS LANDSCAPE

In this section, we plot the landscape of the proposed models and the DeepSAD [15] baseline, showing that the way we

Authorized licensed use limited to: Purdue University. Downloaded on September 29,2025 at 02:28:00 UTC from IEEE Xplore. Restrictions apply.



(a) Detection results for the multi-noise GNSS spoofing attack.

(b) Detection results for the aperiodic GNSS spoofing attack.

(c) Detection results for medium crosswind disturbance.

(d) Detection results for large crosswind disturbance.

(e) Detection results for large front wind disturbance.

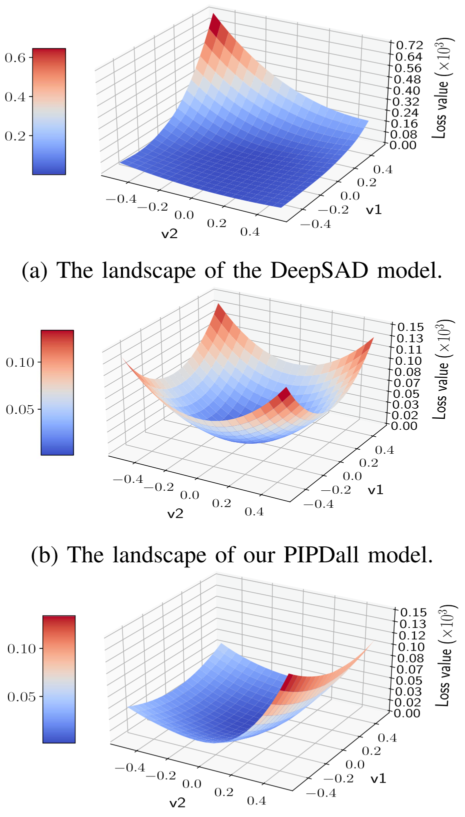
Fig. 4. The detection results of real-world Crazyfile experiments. From top down, each row is the detection results for the multi-noise GNSS spoofing attack, the aperiodic attack, and wind disturbance (with different settings), respectively. The red regions correspond to the true anomaly regions, and the blue curves are the detection flags.

physically inform the anomaly detector smoothens the landscape. We perturb the trained parameters along the directions of the dominant eigenvectors of the Hessian matrix using PyHessian [32]. Both the DeepSAD model and the PIPD models are trained with the linear attack dataset with $p_a = 12\%$ and $p_{la} = 0.03\%$. As shown in Fig. 5, the local loss landscape is almost convex with the physics-informed term in the loss function. In addition, the magnitudes on the z-axis for the PIPD models are more than 60% smaller than those for the DeepSAD model, making our PIPD model more stable under data distribution shifts. This conclusion is also supported by the eigenvalues of their Hessians. The ratio between the largest eigenvalue (in magnitude) and the 100th eigenvalue for the DeepSAD model is -208.9, while this value for the PIPDall method is 31.5, and for the PIPDres model is 45.5. These ratios are lower bounds for the condition numbers of the Hessians. We can also observe that the Hessian of the DeepSAD model has a negative dominant eigenvalue, indicating that the model is stuck at a saddle point. This gap in the model's stability becomes even more significant with more anomaly data in the training set. For example, when we set $p_a = 20\%$ and $p_{la} = 0.5\%$, the maximum magnitude on the z-axis of the DeepSAD plot is more than $10\times$ larger than that of the PIPD models. This again supports the claim that our proposed PIPD models are more robust to unlabeled anomalies in training data and can more efficiently exploit labels.

VI. Crazyflie Experiments

In this section, we validate the effectiveness of the proposed algorithm using Crazyflie [7], an open-source nano quadrotor. In the experiments, the local positions of the Crazyflie quadrotor are measured by the Qualisys motion capture system [33] with a sampling rate of 30 Hz. To enhance the performance of our proposed models in real-world experiments, we fine-tune all the models with a small dataset collected from the Crazyflie flight logs. The variance of GNSS measurement noise is always 1 m during the data collection. This adjusts the models to account for changes in the dynamics and operating environment. We set $p_a = 5\%$ and $p_{la} = 0.3\%$ for retraining. The radii of decision R are chosen in the same way as introduced in Section III or follow the baselines' original papers. During the runtime, the samples are stored in a buffer that can contain 100 samples. With every new sample coming, the buffer pops out the oldest sample in its memory and concatenates the current 100 samples into a 1200

Authorized licensed use limited to: Purdue University. Downloaded on September 29,2025 at 02:28:00 UTC from IEEE Xplore. Restrictions apply.



(a) The landscape of the DeepSAD model.

(b) The landscape of our PIPDall model.

(c) The landscape of our PIPDres model.

Fig. 5. The landscape plots. The plots are generated by perturbing the trained parameters along the directions of the dominant eigenvectors of their Hessian matrix using the method proposed in [32].

dimensional vector as input to the models, enabling real-time anomaly detection after the first 99 samples fill the buffer. We will first introduce our experiments with GNSS spoofing attacks, followed by experiments with wind disturbance.

A. GNSS Spoofing Attack Experiments

In the GNSS spoofing attack experiments, the local position measurements are utilized to emulate GNSS measurements [28]. Position measurements are perturbed by Gaussian noise $\mathcal{N}(0, \sigma^2)$ with zero mean and σ^2 as variance. In the multi-noise attack scenario, the general flight includes 3 phases, namely low-noise phase, medium-noise phase, and high-noise phase, with the corresponding σ^2 equal to 0.01, 0.05, and 0.08, respectively. Flights start with a low-noise phase to mimic the nominal behavior of GNSS signals in an open sky. A linear GNSS spoofing attack is launched during the medium-noise phase with $\alpha = 0.001$, imitating an adversary's intention of drifting the UAV to a desired location. During the medium-noise and high-noise phases, a UAV is more vulnerable to attacks because it is much easier for the attack signals to hide beneath the noise floor and remain undetected. Finally, a high-noise phase is engaged after the attack to show that the proposed method not only detects the presence of a GNSS spoofing attack but can also differentiate it from large non-adversarial noise. The performance of the baselines and our PIPD models during the multi-noise attack experiment is shown in the first row of Fig. 4. The proposed models detect the attacks shortly after the attack engages and always give clear detection flags. The DeepSAD [15] baseline produces equally satisfactory detection results. However, we manually adjust the decision radius of the DeepSAD model to 1.5-times its original value so that the detector gives negative flags during the initial phase. Otherwise, the DeepSAD model tends to always give anomaly flags. This indicates that the physics-informed components in our PIPD framework enhance the generalization ability of the anomaly detector. The TimesNet [30] model produces clear anomaly flags during the attack, but the flags are noisy during the nominal periods. The detection from the RoSAS [31] is too aggressive and ignores many anomalies. Note that most anomaly models raise anomaly flags during the nominal operation right after the attack is disengaged. This is because the UAV takes aggressive actions to compensate for position errors that accumulate during the attack, leading to an unstable state.

In the aperiodic GNSS spoofing case, the attack starts with three short impulses, following a longer and more influential attack. The GNSS measurement noise is always 1 m and $\alpha = 0.002$. As shown in the second row in Fig. 4. Both PIPD models detect the attack during the third impulse. Although the three impulses are of the same duration and magnitude, their effect on the drone is cumulative because of the post-attack instability, as mentioned in the previous paragraph. Thus, the latter impulse is easier to detect compared with the earlier ones. The DeepSAD baseline detects the impulses slightly later than the proposed models and also handles the long attack accurately. However, this level of performance still relies on the manually tuned detection threshold. The TimesNet has a high false-positive ratio, while the RoSAS has a high false-negative ratio in the aperiodic GNSS attack scenario.

B. Wind Disturbance Experiments

Now we present our setting and results for the wind disturbance experiment. We create the wind disturbance with a commercial air blower with a medium wind speed of 5 m/s and a large wind speed of 7.5 m/s. Although the wind speed is not extreme in outdoor operation environments, it can seriously impact the flight of nano-sized UAVs like Crazyflie. In the first two experiments, the UAV follows a straight reference trajectory along the y-axis, and the air blower is located on the x-axis, generating constant speed winds (medium or large) which directly point to the UAV. The detection results are shown in the third and fourth rows in Fig. 4. The final row presents the detection result when the large wind comes from the heading direction of the UAV. The proposed PIPD models and the DeepSAD models (with a manually chosen threshold) can detect the wind disturbance shortly after the wind appears. The TimesNet model again produces noisy detection flags, although the anomaly flags in the windy region are clear. The RoSAS exhibits inconsistent performance. Except for the RoSAS detector, all other models are less sensitive to the front wind disturbance due to the intrinsic aerodynamics of the UAV.

VII. Conclusion and Future Work

In this letter, we proposed a PIPD framework to softly introduce physics knowledge to deep anomaly detectors. Both instantiated PIPD models present significant advantages over the ablated baseline model in generalization ability, resiliency against unlabeled anomalies in training data, and efficiency of exploiting labels. We validated our proposed framework via simulations and real-world experiments. Furthermore, the proposed methodology of physically informing the deep anomaly detector smoothens the loss landscape instead of sharpening it, making our work unique among PIML frameworks. Although the proposed method could be extended to any anomalies that cause discrepancies in the correlation between the control inputs, measurements, and state estimations of a dynamic system, the experimental validations in this letter are limited to GNSS spoofing attacks and wind disturbance. We will explore more types of anomalies in our future work.

References

[1] C. Zhang and J. M. Kovacs, “The application of small unmanned aerial systems for precision agriculture: A review,” *Precis. Agriculture*, vol. 13, pp. 693–712, 2012.

[2] X. Li and A. V. Savkin, “Networked unmanned aerial vehicles for surveillance and monitoring: A survey,” *Future Internet*, vol. 13, no. 7, 2021, Art. no. 174.

[3] C. L. Krishna and R. R. Murphy, “A review on cybersecurity vulnerabilities for unmanned aerial vehicles,” in *Proc. Int. Symp. Saf., Secur. Rescue Robot.*, 2017, pp. 194–199.

[4] L. Ruff et al., “Deep one-class classification,” in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.

[5] Y. Wu, B. Sicard, and S. A. Gadsden, “Physics-informed machine learning: A comprehensive review on applications in anomaly detection and condition monitoring,” *Expert Syst. Appl.*, vol. 255, 2024, Art. no. 124678.

[6] N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2004, pp. 2149–2154.

[7] W. Giernacki, L. Ambroziak, and M. Becker, “Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering,” in *Proc. Int. Conf. Methods Models Automat. Robot.*, 2017, pp. 37–42.

[8] Y. Luo, Y. Xiao, L. Cheng, G. Peng, and D. Yao, “Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities,” *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–36, 2021.

[9] R. T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, “Unmasking the abnormal events in video,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2895–2903.

[10] W. Lu et al., “Unsupervised sequential outlier detection with deep architectures,” *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4321–4330, Sep. 2017.

[11] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 146–157.

[12] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, “f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks,” *Med. Image Anal.*, vol. 54, pp. 30–44, 2019.

[13] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, “Latent space autoregression for novelty detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 481–490.

[14] W. Liu, W. Luo, D. Lian, and S. Gao, “Future frame prediction for anomaly detection–A new baseline,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6536–6545.

[15] L. Ruff et al., “Deep semi-supervised anomaly detection,” in *Int. Conf. Learn. Representations*, 2019.

[16] C. Guille-Escuret, P. Rodriguez, D. Vazquez, I. Mitliagkas, and J. Monteiro, “CADET: Fully self-supervised out-of-distribution detection with contrastive learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, vol. 36, pp. 7361–7376.

[17] G. Pang, C. Shen, H. Jin, and A. V. D. Hengel, “Deep weakly-supervised anomaly detection,” in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2023, pp. 1795–1807.

[18] T. Yang, Y. Lu, H. Deng, and C. Tang, “A hybrid multimodal neural network-based anomaly detection model for UAVs,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 61, no. 4, pp. 10273–10291, Aug. 2025.

[19] H. Qiao, Q. Wen, X. Li, E.-P. Lim, and G. Pang, “Generative semi-supervised graph anomaly detection,” in *Proc. 38th Int. Conf. Neural Inf. Process. Syst.*, 2024, pp. 4660–4688.

[20] Y. Zhou, P. Yang, Y. Qu, X. Xu, Z. Sun, and A. Cichocki, “Anoonly: Semi-supervised anomaly detection with the only loss on anomalies,” *Expert Syst. Appl.*, vol. 262, 2025, Art. no. 125597.

[21] K. Nagaraj et al., “State estimator and machine learning analysis of residual differences to detect and identify FDI and parameter errors in smart grids,” in *Proc. North Amer. Power Symp.*, 2021, pp. 1–6.

[22] Z. Chen, J. Xu, T. Peng, and C. Yang, “Graph convolutional network-based method for fault diagnosis using a hybrid of measurement and prior knowledge,” *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9157–9169, Sep. 2022.

[23] A. Abbaspour, P. Aboutalebi, K. K. Yen, and A. Sargolzaei, “Neural adaptive observer-based sensor and actuator fault detection in nonlinear systems: Application in UAV,” *ISA Trans.*, vol. 67, pp. 317–329, 2017.

[24] A. Perrusquía, W. Guo, B. Fraser, and Z. Wei, “Uncovering drone intentions using control physics informed machine learning,” *Commun. Eng.*, vol. 3, no. 1, 2024, Art. no. 36.

[25] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, “Characterizing possible failure modes in physics-informed neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 26548–26560.

[26] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ, USA: Princeton Univ. Press, 2012.

[27] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, “On the requirements for successful GPS spoofing attacks,” in *Proc. 18th ACM Conf. Comput. Commun. Secur.*, 2011, pp. 75–86.

[28] K. A. Pant, L.-Y. Lin, J. Kim, W. Sribunma, J. M. Goppert, and I. Hwang, “MIXED-SENSE: A mixed reality sensor emulation framework for test and evaluation of UAVs against false data injection attacks,” in *Proc. 2024 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 12414–12419.

[29] H. Sathaye, M. Strohmeier, V. Lenders, and A. Ranganathan, “An experimental study of GPS spoofing and takeover attacks on UAVs,” in *Proc. USENIX Secur. Symp.*, 2022, pp. 3503–3520.

[30] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, “TimesNet: Temporal 2D-variation modeling for general time series analysis,” in *Eleventh Int. Conf. Learn. Representations*, 2022.

[31] H. Xu, Y. Wang, G. Pang, S. Jian, N. Liu, and Y. Wang, “Rosas: Deep semi-supervised anomaly detection with contamination-resilient continuous supervision,” *Inf. Process. Manage.*, vol. 60, no. 5, 2023, Art. no. 103459.

[32] Z. Yao, A. Gholami, K. Keutzer, and M. W. Mahoney, “Pyhessian: Neural networks through the lens of the Hessian,” in *Proc. Int. Conf. Big Data*, 2020, pp. 581–590.

[33] Qualisys AB, “Qualisys motion capture system,” 2024. [Online]. Available: https://www.qualisys.com

Authorized licensed use limited to: Purdue University. Downloaded on September 29,2025 at 02:28:00 UTC from IEEE Xplore. Restrictions apply.