

GIIDS: Generalized Intelligent Intrusion Detection System for Heterogeneous UAVs in UAM

Fahmina Kabir

*Division of Artificial Intelligence and Software
Ewha Womans University
Seoul, Korea
kabirfahmina@ewhain.net*

Nishat I Mowla

*Department of Industrial Systems
RISE Research Institutes of Sweden
Sundsvall, Sweden
nishat.mowla@ri.se*

Inshil Doh

*Department of Cyber Security
Ewha Womans University
Seoul, Korea
isdoh1@ewha.ac.kr*

Abstract—Unmanned Aerial Vehicles (UAVs) are increasingly integral in various sectors, simultaneously encountering rising security threats as UAV and Urban Air Mobility (UAM) networks continue to expand. This paper addresses the challenge of securing UAM networks while also emphasizing generalizability of the security solution to protect heterogeneous UAVs against threats that compromise their stability, reliability and can cause catastrophic failures such as a crash landing. The deployment of traditional machine learning (ML) based intrusion detection systems (IDSs) is often hampered in real-world applications due to a lack of generalizability of the security solution. As a result, the system fails to provide adequate security across the varying models and platforms of UAVs, each with its unique statistical properties and data distributions. To address these challenges, we focus on employing a comprehensive set of UAV sensor parameters, tailored feature engineering and selection to develop multi-stage cross-validated ensemble learning systems to facilitate generalized detection of attack and non-attack cases. For additional analysis, we cross-validate the models using two different cross-validation techniques. The proposed stacking ensemble systems provide the overall best performance, with AUC within the range of 92% to 100% across different cross-validations.

Index Terms—Urban air mobility, machine learning, generalizability, intrusion detection system, unmanned aerial vehicle.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) in Urban Air Mobility (UAM) systems have become increasingly essential across sectors such as agriculture, logistics, disaster management, and public safety, offering benefits like real-time data acquisition, infrastructure inspection, search and rescue operations [18], [19]. UAV communication links range from UAV to UAV, UAV to GCS, to UAV to satellite communication. However, the open and wireless nature of the communication links of these systems makes them vulnerable to various cyberattacks, such as jamming, spoofing, and DoS which can result in unauthorized access, control, and crash landings or arrival at misled destination. As UAVs integrate more deeply into mission-critical operations, ensuring their security and privacy has become a crucial concern.

Intrusion Detection System (IDS) is an essential security solution among all other alternatives to mitigate UAV cyberattacks. Intrusion detection means monitoring events within a computer system or network with an analysis for intrusion signs [18]. An IDS automates to detect different types of

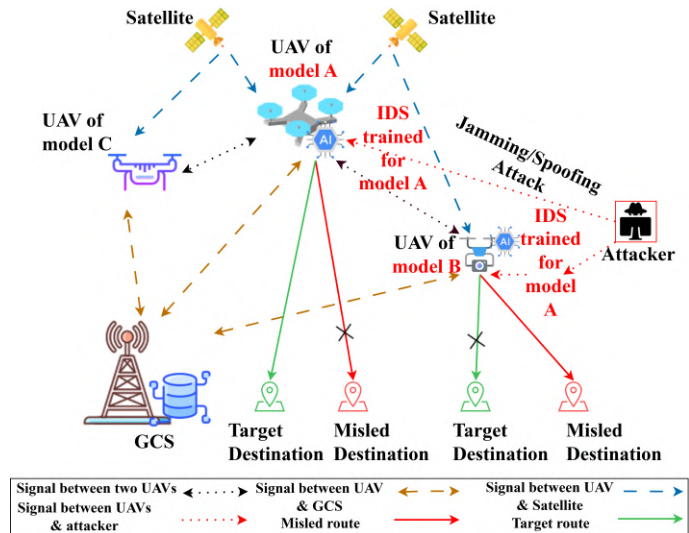


Figure 1. ML-based IDS generalizability challenges for UAVs in UAM. The IDS trained for model A cannot secure UAV model B, leading to misdirection.

anomalies and threats, such as path and message forgeries, malware, UAV capture, routing attacks, GPS spoofing, GPS jamming, and DoS attacks. Using machine learning (ML) techniques in IDSs to deal with such vulnerabilities and threats has gained widespread acceptance in the last few years [3], [4], [5], [12]. While developing ML-based systems, it is assumed that the data for training and testing do not overlap; the data is independent and identically distributed (IID) and has enough variance for good generalization to unseen data. However, in real-world applications, these assumptions are often violated, which degrades the model's performance as shown in Figure 1.

While current ML-based IDSs perform well in controlled environments, they struggle in dynamic, real-world scenarios with diverse data and evolving threat. Thus, the lack of generalizability of the security model is one of the main issues preventing ML-based IDS from being widely adopted [9], [10], [11]. Generalizability refers to an IDS's ability to perform effectively across heterogeneous and unseen datasets, ensuring stability and robustness on new data across various platforms and evolving environments. The generalizability of a model is an important factor in supporting real-world applications

to determine the performance of the model on heterogeneous and unseen UAV data that was not used during the training. This capability is crucial for addressing sophisticated cyber threats in diverse operational settings [12]. To enhance the generalizability of IDS applications, this paper proposes adapting generalizability techniques for a more robust and reliable security system.

A. Summary of Contributions

In this paper, we address the above mentioned limitations found in existing ML-based IDSs by developing a more generalized solution to UAM security. Our approach enhances reliability and adaptability of IDS dedicated to UAM security through a comprehensive generalized IDS framework designed for dynamic environments. Key contributions of our work include:

- We introduce novel feature engineering techniques for enhancing security model's generalization. We identify and construct model agnostic enhanced features from the existing 'Timestamp' information to capture deeper patterns in data which helps the system to distinguish between attack and non-attack instances. It provides the system with the ability to generalize to unseen data other than training data, which makes the system more effective in real-world applications.
- We propose ensemble learning techniques to improve generalizability and detection rates. Different algorithms in the ensemble can specialize in detecting different intrusions or normal patterns, leading to a more robust system capable of handling a wide variety of unseen situations.
- To further analyze our system's generalization power and robustness, we use multi-stage cross validation techniques that test our model on multiple types of data and distributions. By training and validating across diverse datasets, our system becomes resilient and capable of handling the complexity and diversity of real-world networks, ensuring consistent protection even in constantly changing environments.

The rest of the paper is organized as follows: related works is discussed in Section II, the proposed mechanisms are described in Section III and evaluated in Section IV while Section V contains our conclusions.

II. RELATED WORKS

A. UAV Security & IDS

Intrusion detection techniques as a UAV security solution include signature-based, specification-based, statistical-based, anomaly-based, and hybrid approaches [6] which have been enhanced through ML and DL techniques in recent years. These ML-based systems may also use different feature engineering techniques for improved effectiveness and optimization methods like hyper-parameter tuning [12] and genetic algorithms [3].

B. Generalizability

A generalized model balances underfitting, caused by low variance, and overfitting, which occurs when IID is violated, ensuring sufficient generalization to new data [1].

1) *Generalizability Techniques*: There are various types of generalizability techniques, such as:

- *Cross Data Techniques*: In [5] K-fold cross-validation and in [10] cross-data analysis are applied to generalize ML models demonstrating robustness.
- *Ensemble Learning*: Ensemble techniques like boosting, bagging, and stacking improve generalization. In [14] 'Domain Experts' trained over related domains showed strong out-of-distribution performance enhancing robustness across varying datasets.
- *Avoiding Methodological Pitfalls*: In [9], the importance of avoiding data leakage is highlighted by applying oversampling, data augmentation, and feature selection before data splitting. Using external datasets and multiple performance metrics also helps to accurately estimate generalization errors.
- *Data Augmentation*: Data augmentation enhances generalization, as seen in [15] and [16]. TabPTM, which standardizes datasets and builds meta-representations, improved generalization on untrained tabular data without additional training [13].

2) *Generalizability & IDS*: Modern techniques in generalizability research are being applied to IDSs without direct associations with the term "generalization" recently. The work in [8] proposes an ensemble model for eavesdropping attack detection in UAVs using both supervised and unsupervised learners. Similarly, the authors in [7] have proposed a model for detecting man-in-the-middle (MITM) attack using decision trees, K-nearest neighbor, logistic regression, random forest, artificial neural network, and SVM. Works using K-fold cross validation are discussed in [5] and [10], which points to the increased importance of generalization in the specific domain. The study in [12] investigates the integration of physical and cyber aspects into the generalization and detection performance of IDS. The results indicate that the greater the complexity and the range of the attack data, the better the detection, and performances are impressively improved.

III. PROPOSED GENERALIZED INTELLIGENT IDS FOR UAM

The proposed generalized intelligent IDS for UAM framework has three main parts: a) threat model, b) system overview, and c) proposed generalization approach.

A. Threat Model

The UAV network can have multiple components, like UAVs, ground control station, satellites, and target destinations as shown in Figure 2a. In a benign scenario, the UAVs are supposed to reach a specific target destination while maintaining proper communications with other UAVs, GCS, and satellites. On the other hand, when the UAV is under attack by an attacker, it might not be able to maintain proper

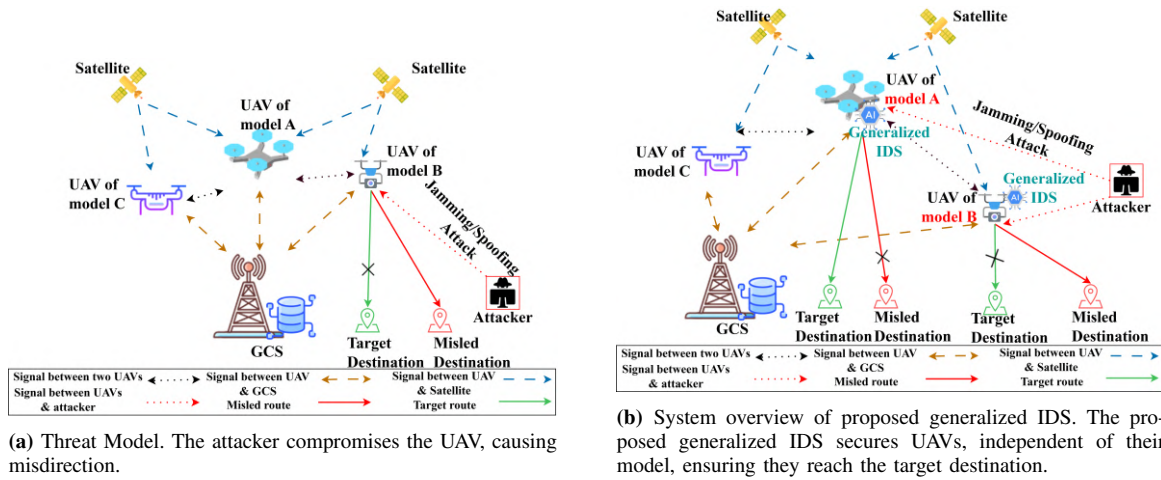


Figure 2. Approach to generalized intelligent IDS for UAM.

communication with the GCS or satellite or receive spoofed GPS information from the attacker, which will lead the UAV to a misled destination. As a result, the UAV enters a state of vulnerability and is open to further, more sophisticated attacks.

B. System Overview

We consider a complex scenario where the UAVs are from different models/platforms with various configurations. As shown in Figure 1, the network includes UAVs deploying models A, B, and C, ground control station, satellites, and target destinations. An ML-based IDS trained to only secure model A UAVs will only secure UAVs of model A and will fail to secure UAVs of model B when under attack. As a result, UAV deploying model B will be led to an incorrect destination because of poor model generalizability. Thus to solve this issue, as shown in Figure 2b, we propose a generalized IDS that can provide security to UAVs from different models.

C. Proposed Generalization Approach

By assessing sensor data from the IMU sub-system of UAVs, our framework aims to detect attacks and prompt the ground station to take the appropriate safety precautions for heterogeneous UAVs that are being attacked. This system makes use of state-of-the-art machine learning algorithms, such as K-nearest neighbors (KNN), random forest, and deep neural networks. Attacks using GPS spoofing, GPS jamming, and DoS are our primary concern. There can be several situations in a wireless network-based UAM context: 1) Best scenario: There is no attack on the UAV. Here, the UAV uses accurate GPS coordinates from satellites to travel to its destination without running into any obstacles, 2) An attack on a specific model of UAV without a generalized IDS framework: In this instance, an attacker uses spoofed GPS signals to take the UAV to an unsuitable and possibly hazardous location or use jamming signals to confuse the UAV and stray from its path, 3) UAV is under attack with generalized IDS framework: The generalized framework shown in Figure 2b recognizes attacks on different models of UAVs, allowing it to react

appropriately, even in the face of a threat effort to use fake GPS coordinates or jamming, overpowering signals. In the event of a lost connection, the UAV may use a specified function resembling the 'failsafe' function on the on-board computer of the DJI Phantom 4, which would cause the drone to hover, automatically land, or return to a predefined ground station [2].

1) *Data Pre-processing for Generalization:* We first perform initial cleaning of the datasets, followed by sorting them based on 'Timestamp' information. We then label the data frames as 'attack' or 'non-attack' accordingly. The subsequent phase involves feature engineering, which will be detailed in the next section. To prevent data leakage, we remove the 'Timestamp' information from the data after feature engineering. Finally, we sample and mix the dataset thoroughly to eliminate any time-related biases, ensuring a more robust model development process (lines 2 to 9 in Alg. 1.).

2) *Feature Engineering for Generalization:* We extract periodic features, such as cosine and sine transformations of the minute information, date, and time-related features from the existing 'Timestamp' information. The periodic features capture the cyclic nature of time [20] which helps the model to learn cyclic patterns that could be linked to benign vs. malicious instances. The periodic features are formed using Eq. 1 and Eq. 2:

$$\text{Cosine Transformation } (m) = \cos\left(\frac{2\pi \cdot m}{60}\right) \quad (1)$$

$$\text{Sine Transformation } (m) = \sin\left(\frac{2\pi \cdot m}{60}\right) \quad (2)$$

where m is the time unit (i.e., in minutes).

The time difference feature calculates the time difference between consecutive data instances. Unexpected gaps or spikes in time intervals can suggest an attack where the attacker is hampering proper communication or sending a sudden burst of transmission. The time difference can be described as:

$$\text{Time Difference}_n = \text{Time}_n - \text{Time}_{n-1} \quad (3)$$

where Time_n is current timestamp and Time_{n-1} is the previous timestamp.

Elapsed time helps to identify the patterns of benign and malicious events over the course of the entire UAV session. Elapsed time can be measured by the following equation:

$$\text{Elapsed Time}_n = \text{Time}_n - \text{Time}_{\text{start}} \quad (4)$$

where Time_n is current timestamp and $\text{Time}_{\text{start}}$ is the starting time of the flight session (lines 10 to 15 in Alg. 1.). The features are described in Table I.

TABLE I. Dataset Features.

Features	Description	Feature Category
gyro_rad[0]	Angular velocity around x-axis (roll)	Raw
gyro_rad[1]	Angular velocity around y-axis (pitch)	Raw
gyro_rad[2]	Angular velocity around z-axis (yaw)	Raw
gyro_integral_dt	Time interval for gyroscope integration	Raw
accelerometer_timestamp_relative	Relative timestamp for accelerometer data	Raw
accelerometer_m_s2[0]	Acceleration along x-axis (m/s^2)	Raw
accelerometer_m_s2[1]	Acceleration along y-axis (m/s^2)	Raw
accelerometer_m_s2[2]	Acceleration along z-axis (m/s^2)	Raw
accelerometer_integral_dt	Time interval for accelerometer integration	Raw
accelerometer_clipping	Indicates if readings exceeded range	Raw
year	Year of data entry	Engineered
month	Month of data entry	Engineered
day	Day of data entry	Engineered
day_of_week	Day of the week	Engineered
hour	Hour of data entry	Engineered
minute	Minute of data entry	Engineered
second	Second of data entry	Engineered
minute_sin	Sine of the minute (cyclical pattern)	Engineered
minute_cos	Cosine of the minute (cyclical pattern)	Engineered
time_diff	Time difference from a reference	Engineered
elapsed_time	Total time elapsed since start	Engineered

3) *Feature Importance Analysis for Generalization:* We measure the feature importance of the features in the new dataset that contains both original and feature-engineered features. To select the most important features for training, we calculate the feature importance based on random forest classifier and gradient boosting classifier. We select the most important features that are present in both of the feature importance lists and remove the zero-importance features from the feature-enhanced dataset (lines 17 to 25 in Alg. 1.).

4) Model Development:

- **Stage 1: Selecting Best Learners:** We train a set of shallow individual learners; random forest, logistic regression, KNN, decision tree, extra tree, and GBM are chosen as the shallow learners and another set of deep individual learners which are 1D-CNN, MLP, LSTM, TabNet, and Transformer. To enhance generalization, we cross-validate the models by simulating different data distributions using various scaling techniques. We trained each model with multiple scaling combinations and selected the best-performing model. Then we select the top shallow learners of each model type for shallow ensemble models and deep learners of each model type for deep ensemble models (lines 1 to 9 in Alg. 2.).
- **Stage 2: Ensemble Learning Stage:** From the previous stage, we have a set of best suitable shallow learners which are logistic regression, KNN, decision tree, and random forest, and a set of best suitable deep learners which are TabNet, MLP, and 1-dimensional CNN. We trained a weighted voting and a stacking ensemble classifier for each set. For the weighted voting ensemble, we used soft voting with the F1-score as the weight criterion. Assuming i to N classifiers in the ensemble,

Algorithm 1. Data Augmentation

```

1: LoadAndSortData();
2: for each sensor dataset  $d \in \{\text{benign\_data}, \text{attack\_data\_1},$ 
    $\text{attack\_data\_2}, \dots, \text{attack\_data\_n}\}$  do
3:   Sort  $d$  by timestamp
4: end for
5: LabelData( $d$ ):
6: for each sample  $s$  in dataset  $d$  do
7:   Assign  $\text{attack} = 0$  if  $s$  is benign; otherwise, assign
      $\text{attack} = 1$ 
8: end for
9: Combine and shuffle datasets
10: FeatureEngineering( $d$ ):
11: for each dataset  $d \in \{\text{benign\_data}, \text{attack\_data\_1},$ 
     $\text{attack\_data\_2}, \dots, \text{attack\_data\_n}\}$  do
12:   Convert timestamp to datetime
13:   Extract time & date-related features and compute
      $\text{minute\_sin}$  and  $\text{minute\_cos}$ 
14:   Calculate  $\text{time\_diff}$  and  $\text{elapsed\_time}$ 
15: end for
16: Drop timestamp
17: FeatureImportanceAnalysis( $df$ ):
18: Split  $df$  into train and test
19: for each model  $m \in \{\text{GB}, \text{RF}\}$  do
20:   Train  $m$  on train
21:   Compute feature importance
22: end for
23: Compare the feature importance lists from GB and RF
24: Identify common zero-importance features in both lists
25: Drop the common zero-importance features

```

each classifier's weight, $W(\text{model}_i)$, is calculated using the total F1-score, Total_F1 (lines 10 to 15 in Alg. 2.).

$$\text{Total_F1} = \sum_{i=1}^N F1(\text{model}_i) \quad (5)$$

$$W(\text{model}_i) = \frac{F1(\text{model}_i)}{\text{Total_F1}} \quad (6)$$

We stacked the chosen classifiers with the logistic regression classifier as the meta-learner for the stacking ensemble classifier (lines 17 to 22 in Alg. 2.).

5) *Additional Cross Data Analysis:* During cross-data validation, we identified significant differences in data patterns between live and simulated data. As a result, models trained on live data struggled to generalize to simulated data, and vice versa. To address this, we combined both live and simulated data for training to introduce similar patterns. To mitigate potential data leakage when sampling from the same dataset, we first sampled the training data and then excluded these instances before selecting test data. We employed two approaches for cross-data analysis: CV1, where we trained on the combined dataset and tested on live data, and CV2, where we trained on data from three UAV models and tested on data from five different UAV models.

Algorithm 2. Model Training Pipeline

```

1: IndividualLearnerTraining( $df$ ,  $scalers$ ):
2: for each scaling method combination  $s$  do
3:   Apply  $s$  to train and test
4:   for each model  $m \in \{RF, LR, KNN, DT, ET\}$  do
5:     Train  $m$  on train
6:     Evaluate  $m$  using accuracy, precision, recall, F1, and AUC
7:   end for
8: end for
9: Save the best performing models,  $m \in \{model\_1, model\_2, \dots, model\_n\}$ 
10: VotingEnsemble():
11: for each model  $m \in \{model\_1, model\_2, \dots, model\_n\}$  do
12:   Compute F1 score for  $m$ 
13:   Assign weight to  $m$  based on its F1 score
14: end for
15: Train weighted voting classifier with soft voting
16: Evaluate model  $m$  on cross validation data
17: StackingEnsemble():
18: for each model  $m \in \{model\_1, model\_2, \dots, model\_n\}$  do
19:   Train base model  $m$  on train data
20: end for
21: Use Logistic Regression as the meta-learner
22: Train stacking ensemble using base models on train data
23: Evaluate model  $m$  on cross validation data

```

IV. PERFORMANCE EVALUATION

A. Dataset Description & Experimental Setup:

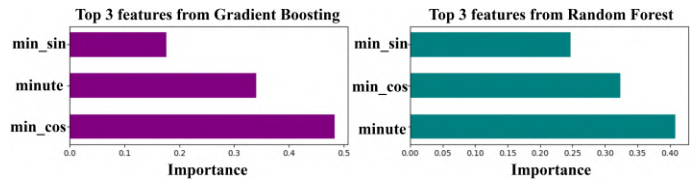
The dataset [17] has both live and simulated data from 5 different models of UAVs (H480 airframe, Fixed-wing plane, Tail-sitter VTOL, Standard VTOL, Quadcopters) for benign, GPS jamming, spoofing and Ping DoS attack environments. The proposed system's performance was evaluated on a machine with 32.0 GB of RAM, an Intel(R) Core(TM) i5-10400 CPU featuring 6 cores and 12 logical processors. We have also used Python and the TensorFlow library for our evaluations.

B. Results of Feature Importance Analysis

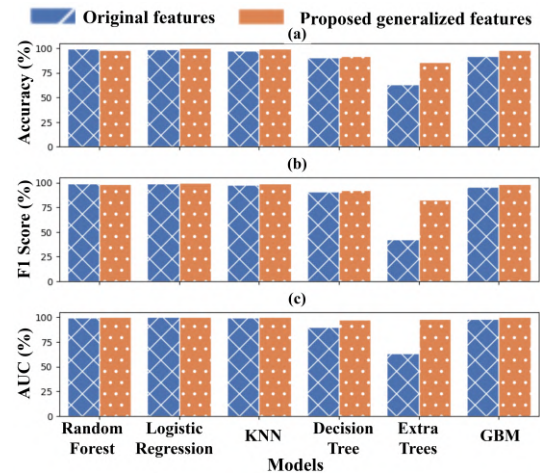
The results of the feature importance analysis reveal that the newly created feature-engineered features are among the most important features for detection for both algorithms (i.e., random forest and gradient boosting), used for the feature importance calculation. The top 3 important features are shown in Figure 3, where all three features (i.e., minute, minute_sin, minute_cos) are from the engineered features.

C. Performance Analysis

We analyzed the performance of individual learners and deep learners as discussed below:

**Figure 3.** Top 3 important features.

1) *Individual Learners*: For cross-validating the individual models, we applied different scaling techniques. To simulate data with different statistical properties, in some cases leading to various distributions, similar to real-world scenarios, we train and test the individual ML models while applying different combinations of scaling methods and compare the results for models trained on feature-engineered data and original data. Overall, we observed superior performance from the models trained on our proposed generalized features. For shallow learners, we achieved a maximum accuracy gain of 22.01% with the Extra Trees classifier, while for deep learners, the maximum gain was 18.18% with the 1D-CNN. The performance of the best shallow individual learners and deep individual learners are displayed in Figure 4 and 5, respectively.

**Figure 4.** Performance of shallow individual learners.

2) *Proposed Ensemble Learners*: We selected the best-performing learner from the trained individual learners and developed ensemble learners of 2 categories: voting and stacking. The performance of each ensemble learner is presented in Table II with different scaling. In the case of shallow learners, the voting ensemble provides better detection performance with perfect accuracy and F1-score. On the other hand, for the deep learners, the stacking ensemble provides an overall better detection performance than voting. We further cross validate the proposed ensemble learners with CV1 and CV2 techniques. Among both the shallow and deep ensembles, the stacking ensemble performs better with AUC within the range of 92% to 100%. The results are presented in Table III.

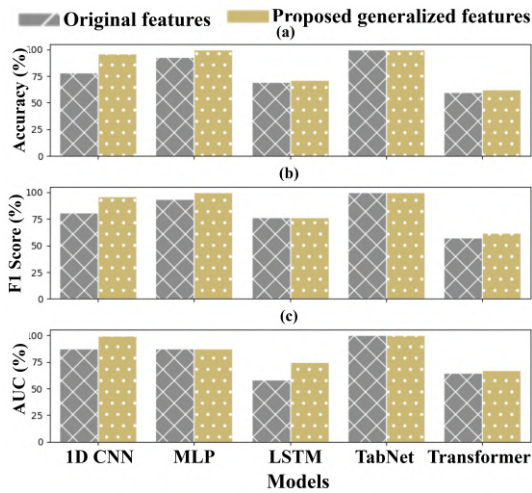


Figure 5. Performance of deep individual learners.

TABLE II. Performance Comparison of Proposed Intrusion Detection Systems.

Model	Accuracy	Precision	Recall	F1-Score	AUC
Shallow Learning Ensembles					
Voting ensemble	1.0000	1.0000	1.0000	1.0000	1.0000
Stacking ensemble	0.9960	0.9960	0.9960	0.9960	1.0000
Deep Learning Ensembles					
Voting ensemble	0.9919	0.9927	0.9910	0.9919	0.9993
Stacking ensemble	0.9926	0.9940	0.9913	0.9926	0.9995

V. CONCLUSION

In this work, we introduce a generalized intrusion detection system specifically designed for diverse UAVs for urban aerial mobility security. The proposed approach addresses the common generalization issues found in existing ML-based IDS solutions across various UAV models. By employing thorough data preprocessing, feature engineering, and feature importance analysis, we develop both shallow and deep learning ensemble IDS models. These models demonstrate significant performance gain, even under complex attack scenarios, and are capable of detecting intrusions across UAVs with varied configurations. Furthermore, the multi-stage cross-data analysis underscores the system's robustness, preparing it for real-world applications. Ultimately, this generalized IDS advances UAV network security by offering high performance, adaptability, and scalability, setting the stage for future enhancements and broader network applications. In the future, we will focus on enhancing our IDS with adversarial robustness and explainability.

TABLE III. AUC Comparison of Proposed Intrusion Detection Systems for Further Cross Data Analysis.

Models	CV1	CV2
Voting ensemble (Shallow Learners)	1.0000	0.8506
Stacking ensemble (Shallow Learners)	1.0000	0.9296
Voting ensemble (Deep Learners)	0.9998	0.8993
Stacking ensemble (Deep Learners)	0.9999	0.9165

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (NRF-2023R1A2C1005712). Inshil Doh is the corresponding author.

REFERENCES

- [1] G. L. Peterson and B. T. McBride, "The importance of generalizability for anomaly detection," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 377-392, 2008.
- [2] A. C. Tang, "A review on cybersecurity vulnerabilities for urban air mobility," in *AIAA Scitech 2021 Forum*, p. 0773, 2021.
- [3] K. Cengiz, S. Lipsa, R. K. Dash, N. Ivković, and M. Konecki, "A novel intrusion detection system based on artificial neural network and genetic algorithm with a new dimensionality reduction technique for UAV communication," *IEEE Access*, 2024.
- [4] J. Whelan, A. Almeahmadi, and K. El-Khatib, "Artificial intelligence for intrusion detection systems in unmanned aerial vehicles," *Computers and Electrical Engineering*, vol. 99, p. 107784, 2022.
- [5] M. S. Munir, S. H. Dipro, K. Hasan, T. Islam, and S. Shetty, "Artificial intelligence-enabled exploratory cyber-physical safety analyzer framework for civilian urban air mobility," *Applied Sciences*, vol. 13, no. 2, p. 755, 2023.
- [6] J. Whelan, T. Sangarapillai, O. Minawi, A. Almeahmadi, and K. El-Khatib, "Novelty-based intrusion detection of sensor attacks on unmanned aerial vehicles," in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp. 23-28, 2020.
- [7] K. Das, R. Basu, and R. Karmakar, "Man-in-the-middle attack detection using ensemble learning," in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-6, 2022.
- [8] K. Das, C. Ghosh, and R. Karmakar, "Eavesdropping Attack Detection in UAVs using Ensemble Learning," in *2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, pp. 01-07, 2023.
- [9] F. Maleki, K. Ovens, R. Gupta, C. Reinhold, A. Spatz, and R. Forghani, "Generalizability of Machine Learning Models: Quantitative Evaluation of Three Methodological Pitfalls," *arXiv preprint arXiv:2202.01337*, 2022.
- [10] M. Benčekroun, P. E. Velmovitsky, D. Istrate, V. Zalc, P. P. Morita, and D. Lenne, "Cross dataset analysis for generalizability of HRV-based stress detection models," *Sensors*, vol. 23, no. 4, p. 1807, 2023.
- [11] V. Gohil, S. Dev, G. Upasani, D. Lo, P. Ranganathan, and C. Delimitrou, "The importance of generalizability in machine learning for systems," *IEEE Computer Architecture Letters*, vol. 24, no. 1, pp. 1-4, 2024.
- [12] 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. 24, no. 7, pp. 4123-4135, 2023.
- [13] H.-J. Ye, Q. Zhou, and D.-C. Zhan, "Training-free generalization on heterogeneous tabular data via meta-representation," *preprint*, 2023.
- [14] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Transactions on Image Processing*, vol. 30, pp. 8008-8018, 2021.
- [15] P. Machado, B. Fernandes, and P. Novais, "Benchmarking data augmentation techniques for tabular data," in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 104-112, Cham: Springer International Publishing, 2022.
- [16] S. Onishi and S. Meguro, "Rethinking data augmentation for tabular data in deep learning," *arXiv preprint arXiv:2305.10308*, 2023.
- [17] UAV Attack Dataset, [Online]. Available: <https://ieee-dataport.org/open-access/uav-attack-dataset>. [Accessed: 22-Jul-2024].
- [18] Z. Yu, Z. Wang, J. Yu, D. Liu, H. H. Song, and Z. Li, "Cybersecurity of unmanned aerial vehicles: A survey," in *Proc. IEEE Aerosp. Electron. Syst. Mag.*, vol. 39, no. 9, pp. 182-215, Sep. 2023.
- [19] B. S. Mohammed and K. K. Basha, "Unmanned aerial vehicles for search and rescue in natural disasters: A review," in *Proc. IEEE Access*, vol. 8, pp. 19073-19089, 2020.
- [20] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, 4th ed. New York: Springer, 2017.