# Detecting Radio Frequency Jamming Attacks on Unmanned Aerial Vehicles Using Machine Learning

Garrison Gralike
*Department of Electrical and Computer Engineering*
*Florida State University*
Tallahassee, Florida
gdg22a@fsu.edu

***Abstract*—This project will explore the use of supervised machine learning to detect malicious radio frequency jamming attacks on unmanned aerial vehicles. Drones rely heavily on wireless communication for control and feedback. This means that interference in the RF channel can cause a lot of issues such as loss of navigation, mission failure, or physical instability. To help address these risks, multiple machine learning models were created. These binary classification models were trained on real RF spectral data. First, the data was preprocessed and standardized. Then two models were built, which this data was then tested on. These models were a Support Vector Machine and a Random Forest model. Each of these were successfully trained to distinguish between normal RF communication and malicious jamming events. Both of the models achieved accuracy values above 0.97, which demonstrates strong detection capability. The Random Forest reduced false alarms but missed more attacks overall. The Support Vector Machine produced a balanced tradeoff between false positives and missed detections. These results show that machine learning is an effective detection mechanism. It also shows that the detection can be tailored to mission requirements by altering the machine learning technique. This allows for prioritization of either sensitivity to attacks or avoidance of false alerts.**

***Index Terms*—UAV Security, Radio Frequency Jamming, Supervised Machine Learning, Intrusion Detection, Support Vector Machine, Random Forest**

## I. Overview and Introduction

### A. Project Overview and Expected Outcomes

The goal of this project is to develop a machine learning detection model that is able to distinguish between benign and malicious RF communications. For this project we will specifically focus on communications directed toward unmanned aerial vehicles (UAV's). Our system should be able to thoroughly analyze RF signal data, and then from there be able to identify any jamming activity. This will allow for early detection and create an opportunity for the drone to respond. The expected outcome of this project is a trained model that can accurately classify signal activity as either normal or malicious. With this, we should be able to improve operational reliability and security in drone communication systems.

### B. Motivations

Drones are one of the best examples of a modern day cyber physical system. They are used in sectors ranging from surveillance, agriculture, logistics, disaster response, and defense [1]. The downside of drones is that their heavy reliance on wireless communication makes them vulnerable to jamming attacks. These attacks can disrupt command-and-control links and also cause a full mission failure [3], [10]. There have been real-world incidents that have demonstrated how low-cost jamming devices can disable or take over drones. This makes jamming a growing security concern for every sector. Detecting these attacks in real time could help mitigate these risks, as well as protect sensitive missions and maintain airspace safety [7].

## II. The Cyber Physical System

### A. Overall Description of the System

The cyber physical system that will be targeted for this project is an unmanned aerial vehicle (UAV). UAV's are good examples of CPS because they combine cyber and physical components into a single unified structure. These drones rely very heavily on radio frequency (RF) communication in order to transmit the navigation commands. RF is also used for telemetry data and control signals between the drone and its controller. Since the cyber and physical components of drones are so closely related any interference could have huge consequences [2]. An attack at the communication layer could have immediate physical consequences. These include flight path deviation, mission interruption, or loss of control. This makes the communication layer both a very essential component and a key vulnerability [3]. The relationship between the cyber layer and the physical layer can be seen in Figure 1.

### B. Physical Components (Physical Layer)

The physical layer of UAVs involves the physical components needed to maintain stable flight and execute the drone's missions. This layer mostly consists of the drone airframe, propulsion system, flight controller, onboard sensors, communication modules, and finally power supply. The airframe is important to providing structural integrity. Other components such as the propulsion and flight controllers work together to control the drone's movement. The onboard sensors provide all the data used for navigation and stabilization. The RF modules maintain the communication link with the ground control station. Finally, the power system makes sure the drone remains operational. Together, these components form the drone's physical layer.

### C. Cyber Components (Cyber Layer)

The drone's cyber layer is the part that contains all the software, communication, and control logic. This is the part that allows the drone to process data, follow commands, and execute different flight operations. Some other important parts of this layer include the onboard flight control software, communication protocols, and encryption mechanisms that are used to secure data transmission. One of the most important jobs of the cyber layer is to interpret RF signals and translate them to usable instructions. These flight instructions are what allow the drone to send commands to the physical components while also maintaining communication with the operator. A successful attack on this layer could cause significant disruptions to the physical behavior of the drone, which is why its integrity is so important. This shows the critical role that having secure communication plays in the proper operation of the entire system [1].

### D. Context: Implementing the Machine Learning Model

Implementing the proposed machine learning model will help keep the drones RF communication secure. It would most likely be included either at the ground level or at an edge processing node directly on the drone. During normal

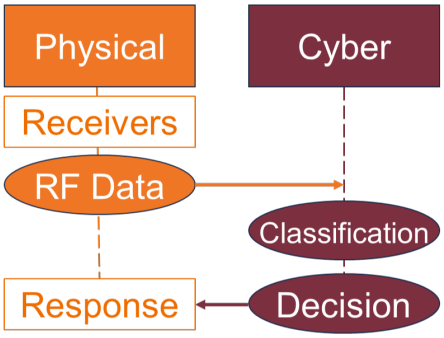


Fig. 1. Interaction Between Physical and Cyber Layers

operation, the ground control station continuously receives radio frequency signals from the drone and vice versa. When ML is included, these signals will be analyzed and then classified as either benign or malicious. If an attack is detected, the system can issue alerts or in a real scenario activate countermeasures such as changing frequencies or switching to a backup communication channel [9]. Keeping the ML model at the ground level would be advantageous because it would avoid overloading the drone with computational tasks.

### E. Visualization

The best way to understand the interaction of the cyber and physical layers is to conceptualize it as a continuous feedback loop. The drone's sensors and flight controller receive and execute commands via RF. This communication link is the primary attack surface for a malicious person to interfere. In our scenario, these signals move to and from the ground control station, where the machine learning detection system is being used to analyze the communication.

## III. Process and Problem

### A. Process to be Enhanced

The main process targeted in this study is the quality of RF communication in UAV systems. These systems rely on continuous communication between the drone and its ground controller and are unable to operate properly if there is any interference. When the normal signals are jammed or interfered with, the drone can lose navigation capabilities or even entirely lose its link to the operator [5]. The drone's communication channel is obviously essential for successful mission execution, but it's also paramount to flight safety. Current drone systems lack any type of detection mechanisms to distinguish between normal signal fluctuations and deliberate malicious interference. This project seeks to address that gap. The introduction of machine learning should be able to detect and classify these attacks before they cause any critical failures.

### B. Formulation of Research Problem

For this project, the research problem can be defined as the following: To develop a reliable method to identify malicious jamming signals within the RF communication stream of a drone in real time. The traditional rule-based detection methods are not able to adapt to subtle or evolving RF interference. The biggest challenge here is differentiating between benign signal variations caused by normal natural environmental factors. Some variations could also be expected as normal communication behavior. Our ML algorithm should be able to identify between these and an attacker. A big issue with the traditional detection methods is that they are set and structured, meaning they are unable to adapt to complex or subtle attack patterns. This is why we need a smart solution. Supervised machine learning is an attractive solution because it can learn from real data and detect non-obvious signatures of interference [4]. Applying the ML model to incoming RF signals will allow the system to make rapid, accurate classifications of the communication signal.

### C. Anticipated Final Outcome

The anticipated result of this work is to develop a machine learning model that is capable of classifying incoming RF communications as either benign or malicious with both high accuracy and low latency. The ML model should be able to serve as an intrusion detection layer for the drone communication system. The model should be easily able to classify attacks as either benign or malicious. It should also be accurate enough that small RF variations due to environmental factors are ignored, yet small RF variations due to malicious attacks are detected. This project hopes to strengthen UAV communications.

## IV. Solution - The Dataset

### A. Dataset Description

The dataset used in this project is the RF Jamming Dataset (Dania Herzalla et al., 2024) [6]. This dataset was comprised of real data with about 2000 samples, 8 features, 2 classes, and about 1000 per class. Each row of this dataset represents one spectral scan sample. There are numeric RF features for each sample, and then they are classified with a binary label to indicate whether the signal is benign or malicious. This is to separate normal communication or attack jamming activity. Each feature of this dataset can be seen in Table 1, along with a short description. This data set will be useful since it contains real RF signal measurements rather than synthetic data. This means it provides a more realistic foundation for developing a machine learning model. It also will help better generalize to real-world attack scenarios.

### B. Data Preprocessing

In order to use the dataset, some preprocessing was needed. A subset was taken since the original dataset was too large to use on a local device. The original dataset also included scans from different locations, so the dataset was filtered to only include scans from a singular location for consistency. This

helped reduce environmental variability and avoid location specific patterns which could have inflated model performance. Next, the dataset was filtered to include only active scan samples since these would better represent realistic communication behavior. In the dataset documentation, active scans are described as "spectral scans obtained through actively transmitting signals into the RF spectrum and observing the resulting reflections, as would occur in real-world scenarios where devices are actively communicating. During an active scan, the channels are sequentially scanned in a predefined order, returning to the device's current channel before scanning the next channel in sequence." [6]

By this preprocessing was completed, the new subset was at a device-friendly size of approximately 2,000 samples, evenly split between benign and malicious. This subset was now ready to be fed to the model. The model input consists of eight numeric features, all of which can be shown in Table 1. the dataset preview below. Since this is a binary model the target label is the Output column, where the output can either be 1 or 0. This is because the dataset produces two classes total, where 0 represents benign and 1 represents malicious jamming. The dataset also contains a Source file field allowing for easy traceability back to the original CSV files. This field is metadata and was not treated as a numeric RF feature for training.

## V. Solution - The Model

### A. Machine Learning Type and Task

In this project, supervised machine learning is used. Supervised learning is used because each sample includes either a 1 or 0 label in the Output field. This output indicates whether the observation is benign communication or malicious interference. This also means that the task for this project is binary classification. Here the models will map an input RF feature vector to one of two discrete classes, 0 or 1.This matches the goal of creating a system that can automatically detect unusual RF signals.

### B. Dataset Split and Training Procedure

The processed dataset was randomized and separated into an eighty percent training set and a twenty percent testing set. The testing set was only used for the final evaluation since we wanted to test on unseen data. During the hyperparameter tuning, cross validation was used on the training set to help choose the best model. The performance of each model was quantified using accuracy, precision, recall, and F1 score, and error behavior. This information was then interpreted using the confusion matrix to distinguish false alarms from missed detections.

### C. Support Vector Machine Classifier

Two different models were tested in this project. The first was a Support Vector Machine. This model was chosen mainly because margin-based classifiers usually work well when the data is not separated by a simple straight line. The features in this dataset are structured numeric values, meaning that an SVM makes sense for the task [8]. The goal of the SVM is to draw a clear boundary that separates normal signals from jamming signals.A nonlinear kernel was used because real RF jamming does not usually behave in a perfectly linear way across each feature.

One of the main strengths of the SVM is that it can handle complicated feature spaces and balance its complexity using regularization. The downside is that its performance depends a lot on feature scaling and choosing the right hyperparameters.On top of this, training time can also take longer as the dataset grows. Because of this, the features were standardized before training so that no single measurement could dominate the model.

### D. Random Forest Classifier

The second model used in this study was a Random Forest classifier.The Random Forest model works by using a lot of different trees that each make their own prediction, and then the final result comes from combining them. This model was selected as the comparison model because jamming behavior can involve nonlinear relationships between the diffrent features [8]. This means that tree-based models can learn these patterns without needing feature scaling.

Using a Random Forest model has several advantages. One of which is that it can handle mixed relationships well and gives very useful information about feature importance. However, one of the drawbacks is that very large forests can be slow during prediction. They can also overfit if the trees are allowed to grow without limits, which is especially true when the dataset is smaller.

### E. Hyperparameters and Grid Search Tuning

To improve both models, grid search was used to tune several hyperparameters. For the SVM the kernel type, regularization value C, and the gamma value for the kernel were all adjusted. For the Random Forest the number of trees, the maximum depth of each tree, and the minimum number of samples required to split a node were adjusted. Each hyperparameter was tested with several possible values. The best setup was selected by using cross validation on the training set. This helped avoid picking hyperparameters by luck and made the comparison between the two models more fair.

## VI. Experimental Results

### A. Evaluation Metrics

To evaluate each model's performance three metrics were used. Each models accuracy, detection rate, and the F one score were analyzed. Accuracy represented the percentage of correct predictions out of all total predictions made. It gives a general sense of how well the classifier performed overall. Detection rate measured how many jamming events were correctly identified. This is equivalent to recall for the malicious class. Detection rate is especially important in this type of problem because if the model fails to detect malicious interference it can be more harmful than raising a false alarm.

TABLE I
RF FEATURES USED IN THIS STUDY

| Feature name | Feature description |
|---|---|
| freq1 | Center frequency of the channel being monitored. |
| noise | Background noise power measured in the band. Elevated noise values indicate wideband interference or active jamming. |
| max magnitude | Peak magnitude of the received spectrum during the sample. A sudden jump in this value is a strong indicator of a jammer turning on. |
| total gain db | Total receiver gain in decibels applied by the front end and amplifier chain. Needed to interpret power levels consistently across samples. |
| base pwr db | Baseline received power in decibels under nominal conditions. Serves as a reference for detecting power increases caused by jamming. |
| rssi | Received signal strength indicator reported by the hardware. Higher than expected RSSI values often correspond to strong interference sources. |
| relpwr db | Received power in decibels relative to the baseline or reference level. Captures how much stronger the current signal environment is compared with normal. |
| avgpwr db | Average received power in decibels over the observation window. Smooths short spikes and highlights sustained jamming activity. |

The F one score balances both precision and recall. It does this by representing how well the model avoids false positives while also correctly identifying true malicious events. In this project we were able to see that both the Support Vector Machine and the Random Forest achieved strong values across all three metrics, which suggests that the models were well suited to this task.

### B. Confusion Matrix

To better understand how each classifier made its decisions a confusion matrix was used. This helped by breaking down correct and incorrect predictions into true positives, true negatives, false positives, and false negatives. The confusion matrix showed us that the Random Forest Model achieved 199 true negatives and 194 true positives. It also had only 1 false positive and 6 false negatives. These results, shown in Figure 3, show that the Random Forest was extremely effective at avoiding false alarms. It rarely labeled a normal signal as malicious. However, its most common mistake was missing malicious jamming events. In contrast, the Support vector Machine, shown in Figure 2, produced 197 true negatives and 197 true positives. It also had 3 false positives and 3 false negatives. These results indicate that the SVM was more aggressive in detecting malicious activity. It correctly identifying slightly more jamming events than the random forest, but at the cost of producing more false alarms.

### C. Evaluation

When comparing the two models directly, we can see that both produced a strong classification performance. The metrics recorded for each model were accuracy, precision, recall, and F1 score, and each model performed well in

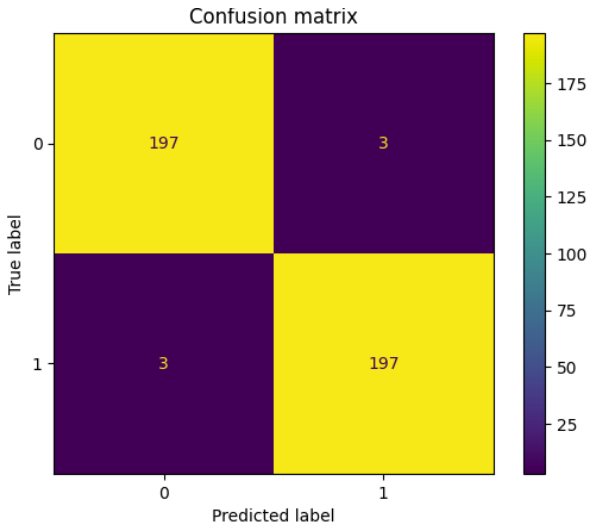


Fig. 2. Support Vector Machine Confusion Matrix

every category. These results can be seen in Figure 4 and Figure 5. An analysis of these results also shows that these model have slightly different tradeoffs. The Random Forest achieved an accuracy of 0.9825 and demonstrated a preference for conservative decision boundaries. This led to fewer false positives than the SVM. This would make the Random Forest Model potentially more suitable in cases where false alarms could be disruptive. However, its most common mistake was failing to detect attacks. This would make it less useful for cases where missed detection is detrimental but false alarms are acceptable. The SVM had an accuracy of 0.985 and was able to detect more malicious signals. This suggests a stronger sensitivity to identifying attacks. This model did generate more false alarms, but this shows a slightly better balance between detection and overall robustness. This makes it a strong candidate for systems where missing an attack is more costly than responding to a false alert.

### D. Evaluation Analysis

The differences seen between the two models can be explained by how each algorithm learns decision boundaries and handles the different feature relationships. The Support Vector Machine produced a nearly symmetric confusion matrix. It had three false alarms and three missed jamming events. This indicates that it learned a balanced boundary between the two classes. This type of result is expected from an SVM, since it works by trying to maximize the margin rather than aggressively fit patterns in the data. Even after tuning, the SVM still favored a middle ground solution. It left a slightly wider separation between classes, which reduced the chance of overfitting. However, this also prevented it from fully separating borderline cases. As a result, the model is very balanced. It does not prioritize either minimizing false positives or maximizing detections.

In contrast, the Random Forest displayed a more uneven tradeoff. It produced only one false positive but six missed

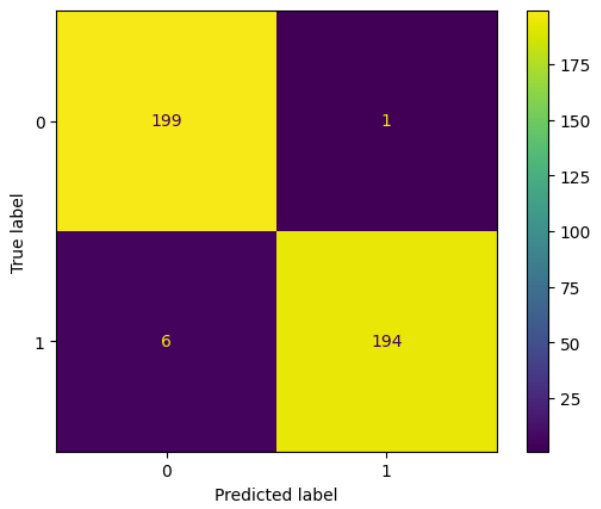


Fig. 3. Random Forest Confusion Matrix

attacks. This suggests that the forest learned a tighter set of rules for identifying benign signals. Also, it shows that it was slightly more conservative about labeling something as malicious. The model likely learned very specific patterns in the dataset that helped it confidently avoid false alarms. Unfortunately, this made it less responsive to malicious cases if they did not strongly match those patterns. Decision trees naturally specialize in localized feature splits. This means that when combined in a forest, they can only trigger a malicious classification when multiple trees agree. Clearly, this same mechanism can increase false negatives when an attack does not clearly appear in enough trees. This all leads to a mild tendency toward under detection in exchange for lower false alarm rates.

Overall, the SVM behaved like a generally cautious and balanced classifier. The Random Forest behaved more selectively and leaned toward protecting against false alerts. These outcomes align with the design philosophy of each algorithm and provide two different operational tradeoffs for RF jamming detection. Each has its respective use depending on whether the system is more concerned with avoiding false alarms or avoiding missed attacks.

## VII. Conclusion

### A. Project Conclusions

This project was able to demonstrate how supervised machine learning can be used to detect malicious jamming activity in UAVs. The system was able to detect benign and malicious signals by training models on real spectral data. Both the Support Vector Machine and the Random Forest model were able to do this with high accuracy. They were able to study and observe nonobvious distinctions in feature behavior that are characteristics of jamming activity. This success proves that machine learning can serve as a reliable defense mechanism for drone communication systems [11].

```
Accuracy: 0.9825
Precision: [0.97073171 0.99487179]
Recall: [0.995 0.97 ]
F1: [0.98271605 0.98227848]
```

Fig. 4. Random Forest Metrics

```
Accuracy: 0.985
Precision: [0.985 0.985]
Recall: [0.985 0.985]
F1: [0.985 0.985]
```

Fig. 5. Support Vector Machine Metrics

Each model had its strengths and weaknesses. The Random Forest showed more conservative decision behavior. It was able to nearly eliminate false alarms, at the cost of missing a number of attacks. The Support Vector Machine offered a more balanced tradeoff. It had three false alarms and three missed attacks. This means it had a higher false alarm rate than the Random Forest model, but was more reliable at detecting attacks. These results show that ML detection can be used to meet specific goals. This can include things such as prioritizing sensitivity to attacks or minimizing false alerts. Overall, the outcomes of this study show that machine learning provides a meaningful way to strengthen UAV communication. It also shows that different ML models excel at different things, depending on mission requirements.

### B. Future Directions

While this work shows that machine learning is effective, there are several ways it could be extended. One improvement would be to involve testing the larger data set. It would be beneficial to test the model on data from different operational environments. Data collected from a range of environments would help make sure the model can be used in different signal conditions. Future models could also be deployed and tested in real life on the embedded or edge processing hardware onboard a drone [12]. This would allow real-time detection without relying on a ground station. Finally, in the future, adaptive learning or online retraining could be used to allow the system to update as attackers change techniques over time. [13] Expanding in these directions would help create a more resilient defense system.

## REFERENCES


[1] M. Haider, I. Ahmed, and D. B. Rawat, "Cyber Threats and Cybersecurity Reassessed in UAV-assisted Cyber Physical Systems," 2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN), Jul. 2022, doi: https://doi.org/10.1109/icufn55119.2022.9829584.

[2] G. Rong-xiao, T. Ji-wei, W. Bu-hong, and S. Fu-te, "Cyber-Physical Attack Threats Analysis for UAVs from CPS Perspective," IEEE Xplore, Mar. 01, 2020. https://ieeexplore.ieee.org/document/9103889.

[3] G. K. Pandey, D. S. Gurjar, H. H. Nguyen, and S. Yadav, "Security Threats and Mitigation Techniques in UAV Communications: A Comprehensive Survey," IEEE Access, vol. 10, pp. 112858–112897, 2022, doi: https://doi.org/10.1109/access.2022.3215975.

[4] Fadhila Tlili, Samiha Ayed, and Lamia Chaari Fourati, "Advancing UAV security with artificial intelligence: A comprehensive survey of techniques and future directions," Internet of things, pp. 101281–101281, Jul. 2024, doi: https://doi.org/10.1016/j.iot.2024.101281.

[5] O. Šimon and T. Götthans, "A Survey on the Use of Deep Learning Techniques for UAV Jamming and Deception," Electronics, vol. 11, no. 19, p. 3025, Sep. 2022, doi: https://doi.org/10.3390/electronics11193025.

[6] D. Herzalla, "RF Jamming Dataset," Kaggle.com, 2024, doi: https://doi.org/10.36227/techrxiv.20115980.v1

[7] J. Harshil et al., "AI-based Approach for Radio Frequency Jamming Attack Detection in Unmanned Aerial Vehicles," 2024 IEEE 29th Asia Pacific Conference on Communications (APCC), pp. 266–271, Nov. 2024, doi: https://doi.org/10.1109/apcc62576.2024.10768068.

[8] G. Kasturi, A. Jain, and J. Singh, "Detection and Classification of Radio Frequency Jamming Attacks using Machine learning," doi: https://doi.org/10.22667/JOWUA.2020.12.31.049.

[9] Marko Jacovic, X. R. Rey, G. Mainland, and K. R. Dandekar, "Mitigating RF jamming attacks at the physical layer with machine learning," IET Communications, vol. 17, no. 1, pp. 12–28, Oct. 2022, doi: https://doi.org/10.1049/cmu2.12461.

[10] Guo Rong-xiao, B. Wang, and J. Weng, "Vulnerabilities and Attacks of UAV Cyber Physical Systems," Apr. 2020, doi: https://doi.org/10.1145/3398329.3398331.

[11] Samuel Chase Hassler, Umair Ahmad Mughal, and M. Ismail, "Cyber-Physical Intrusion Detection System for Unmanned Aerial Vehicles," IEEE Transactions on Intelligent Transportation Systems, pp. 1–12, Jan. 2023, doi: https://doi.org/10.1109/tits.2023.3339728.

[12] F. Fei, Z. Tu, D. Xu, and X. Deng, "Learn-to-Recover: Retrofitting UAVs with Reinforcement Learning-Assisted Flight Control Under Cyber-Physical Attacks," 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 7358–7364, May 2020, doi: https://doi.org/10.1109/icra40945.2020.9196611.

[13] F. O. Olowononi, D. B. Rawat, and C. Liu, "Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS," IEEE Communications Surveys and Tutorials, pp. 1–1, 2020, doi: https://doi.org/10.1109/comst.2020.3036778.