Article Type: Description (see below for more detail)

Threatening or Not? Early Prediction of Threat Posed by Drone Trajectories

Tonmoay Deb, *Northwestern University, Evanston, IL, 60201, USA*

Sven de Laaf, *Netherlands Police, The Hague, Netherlands*

Valerio La Gatta, *Northwestern University, Evanston, IL, 60201, USA*

Odette Lemmens, *Netherlands Police, The Hague, Netherlands*

Roy Lindelauf, *Netherlands Defence Academy, Breda, 3509 AA, Netherlands*

Max van Meerten, *Municipality of The Hague, Netherlands*

Herwin Meerveld, *Netherlands Defence Academy, Breda, 3509 AA, Netherlands*

Afke Neeleman, *Municipality of The Hague, Netherlands*

Marco Postiglione, *Northwestern University, Evanston, IL, 60201, USA*

V.S. Subrahmanian, *Northwestern University, Evanston, IL, 60201, USA*

Abstract—Over the last few years, there has been increasing use of drones by terror groups and in armed conflict. Several technologies have been developed to detect drone flights. However, much less work has been done on the Drone Threat Prediction Problem (DTPP): predicting which drone trajectories are threatening and which ones are not. We propose DEWS (Drone Early Warning System), a framework to solve this problem. Solving DTPP early is key. Once a drone starts on its trajectory, we show that DEWS can make accurate predictions within 20-30 seconds of the flight with an F1-score of over 80% on data about a major European city. We study the tradeoff between earliness of predictions and accuracy. We identify the key features that ensure good predictions.

Terror groups such as ISIS [1], the PKK [2], Lashkar-e-Taiba[1], and others are increasingly using drones in various operations. Drones are also becoming a preferred instrument of nation state warfare as evidenced by the war in Ukraine. There is now deep concern that cities will be targeted by drone attacks [3].

However, the skies over a city are traversed by numerous drones. Realtors use drones to get aerial shots of properties for sale[4], insurance companies use drones to look for undeclared pools and property damage [5], sports arenas use drones to capture crowd pictures and game plays [6], and more. A major problem for police and security organizations around the world is to distinguish the few drones that pose a threat from the many that are benign. And we need to do this as early as possible. As stated by defense experts at the Modern War Institute at West Point[2], "The earlier you detect a threat (drone, rocket, missile, or artillery), the sooner you can alert the force to seek shelter while the air defense operators work to employ their systems to defeat the threat".

This is the problem that we address in this paper: developing a machine learning model that takes an *initial part* (e.g. the first 5, 10, 20, 30 seconds, ...) of a drone trajectory as input and predicts if it is threatening

Digital Object Identifier 10.1109/XXX.0000.0000000

Authors are listed in alphabetical order, not order of contribution. Work partly funded by ARO grant W911NF2320240.

[1] https://www.indiatoday.in/india/story/drone-attack-initial-probe-lashkar-role-jammu-and-kashmir-police-chief-1820679-2021-06-29

[2] https://mwi.westpoint.edu/understanding-the-counterdrone-fight-insights-from-combat-in-iraq-and-syria/

or not. The smaller the "initial" part, the earlier we can bring a potentially threatening trajectory to the attention of security agencies. But a small initial part might be too small to make a good prediction.

Though there has been a great deal of work on predicting trajectories of moving objects (e.g. mobile phones [7], drones [8]), there has been relatively little work on quantifying the *threat* posed to a city or geographic area by a drone. To quantify this threat, we must not only understand the drone's trajectory, but also the drone's capabilities (e.g. payload, battery life, max speed) and the value of the assets on the ground that the drone is flying over.

Our DEWS Drone Early Warning System predicts whether a drone trajectory is threatening or not. DEWS tries to understand how long we need to observe a drone flight in order to predict whether the drone poses a threat or not.

DEWS is novel in several respects. (i) As far as we know, DEWS is the first framework to predict the *threat* a drone flight poses to a city. (ii) It is the first framework to understand the tradeoff between the time for which a drone trajectory is observed (the "observation window") and threat prediction accuracy. (iii) In addition to the trajectory, DEWS looks at features about the drone's capabilities, violations of no fly zones, assets on the ground, and more. (iv) DEWS identifies the key features linked to accurate predictions. We find that the values of assets on the ground that a trajectory flies over constitute the single most important feature in assessing the threat of the trajectory. (v) DEWS can make predictions with an F1 score exceeding 0.8 in 3 seconds in operational use (after training), suggesting that it can be used for real-time predictions. (vi) DEWS has been tested by Dutch police, municipal, and security officials on 8 months of real trajectories over The Hague and the results show an F1-score over 0.85.

This paper is organized as follows. The "Related Work" Section discusses related work. Next, Section "DTPP: Drone Threat Prediction Problem" formalizes the problem studied. Our "DEWS Architecture" Section provides a detailed description of our architecture, including its features and training process. Section "Experiments" presents the predictive performance of 11 ML models and a late fusion classifier) as the observation (i.e. training) window increases. After this, a "Limitations and Future Work" section describe limitations of the framework.

Related Work

Predicting the future location of a moving object has been explored in various domains [9], [10]. Vision-based object tracking methods [11] predict the future location of moving objects. This work has been used in self-driving cars [10] to create plans based on predicted future locations of humans and nearby moving objects. Other research uses historical GPS data to predict mobility of devices [7].

Numerous papers predict vehicle trajectories by learning models from historical driving data [12]. Temporal models such as LSTMs with attention networks [13], [14], [15] have been proposed for trajectory prediction. Recent advances incorporate trajectories of nearby vehicles to reduce accidents [16]. Drone trajectory prediction has been widely studied across various applications, including autonomous aerial cinematography [17], delivery [8], and search and rescue [18].

There is also work on predicting a mobile phone's next location based on historical movement data [19], [20]. These approaches include sequential pattern learning techniques to predict a phone's future location and/or human movements.

DEWS differs from past efforts in two respects. First, it predicts if a drone trajectory is *threatening* or not, which past works don't do. Second, DEWS is the first to study how early in a trajectory we can make a good prediction. This is particularly important because timeliness is key in mitigating drone threats. The identification of a threat is crucial input for the subsequent command and control process resulting in some kind of intervention. DEWS not only obtains features from the drone trajectory, but also from assets on the ground and the drone's capabilities. Past work doesn't consider assets on the ground.

DTPP: Drone Threat Prediction Problem

Suppose C is a city to be protected. We obtain a map of C containing locations of important national buildings, security installations (e.g., police stations, military bases), government buildings, hospitals, tourist attractions, entertainment venues, homes, parks, roads, bridges, utilities, etc.[3] Once the city C is selected, we define an asset valuation map $Val(C)$, which assigns a value to every point within the city. High $Val(C)$ values corresponds to important locations.

Consider a drone d flying over C. Its *trajectory* τ_d is a finite sequence $(\ell_1^d, t_1), \ldots, (\ell_n^d, t_n)$ where each

[3]We used OpenStreetMap https://www.openstreetmap.org.

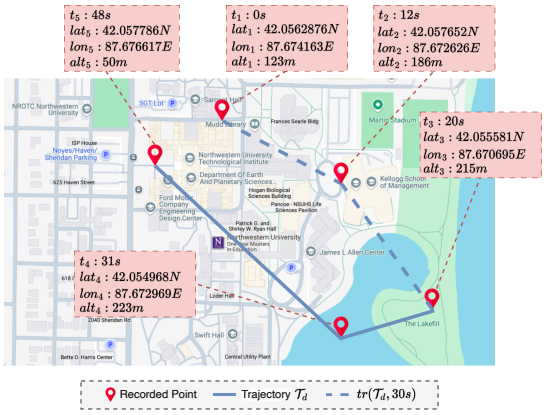

FIGURE 1: Sample drone trajectory with its 30-second restriction. The trajectory data is from a real drone, but the city was altered for security reasons.

$\ell_i^d = (lat_i, long_i, alt_i)$ is d's location at time t_i in terms of latitude, longitude and altitude, respectively. The *temporal restriction* of a trajectory τ_d to time j, denoted $tr(\tau_d, j)$, is the set $\{(\ell_i^d, t_i) \mid (\ell_i^d, t_i) \in t \wedge t_i \leq j\}$. We use $\mathcal{T}$ to denote a given set of trajectories and we use $tr(\mathcal{T}, j) = \{tr(\tau, j) \mid t \in \mathcal{T}\}$ to be the restriction of the trajectories in $\mathcal{T}$ to the first j timepoints. Figure 1 shows a drone's trajectory τ_d and its restriction $tr(\tau_d, 30)$ to 30 seconds.[4]. As an example, we may wish to predict the level of threat posed by $tr(\tau_d, 30)$ after the 30 seconds of the flight. The threat score is given by $y(\tau_d) \in [1, 10]$. The higher the threat score, the more threatening the drone's trajectory.

The Drone Threat Prediction Problem (DTPP[lev]) is to learn a function $f_{lev} : (d, tr(\tau_d, j)) \rightarrow \{0, 1\}$, such that $f(d, tr(\tau_d, j)) = 1$ if the threat posed by $\tau_j \geq lev$, where $lev \in [1, 10]$.

DTPP can work after any observation window $j > 0$ after the drone flight begins. This is critical for security. The earlier predictions are made about the threat level of trajectories, the earlier security officials can prioritize their responses.[2]. Earliness of prediction must be balanced against accuracy of prediction. Understanding this balance is a major goal of this paper.

DEWS Architecture

Figure 2 shows the DEWS architecture. DEWS uses a dataset of drone trajectories annotated by Dutch police and municipality — Table 1 presents a brief overivew.

The *Feature Extraction* module extracts key features that characterize a drone trajectory. The *Threat Classification* module combines the predictions of 11 classifiers to provide a final classification.

TABLE 1: DEWS Dataset Statistics

Statistic	Threat Score [1, 3]	[4, 7]	[8, 10]
Number of Drones	18		
Number of Trajectories	213	94	42
Avg. Duration (s)	265	298	286
Avg. Distance (m)	435.1	988.2	752.1
Avg. Altitude (m)	62.69	115.9	100.7
Avg. Speed (km/h)	7.088	14.58	10.41

Trajectory Training Data

We collected a dataset of 349 drone trajectories to train DEWS. These trajectories represent all known recorded drone flights over The Hague captured by Dutch police and municipality over a period of eight months. The threat of each trajectory was assessed on a 1-10 scale by at least one police official. **Fifty trajectories were annotated independently by 2 or more police and municipal officers. To assess agreement amongst the officials, we computed the inter-annotation weighted Cohen's kappa coefficient of 0.772, indicating substantial agreement amongst annotators.**.

Police officials then categorized trajectories as *low* threat (score < 4), *medium* threat (score $\in [4, 8)$), and *high* (score ≥ 8) threat.

Feature Extraction

This module extracts 110 features for each trajectory.[5]

Basic features offer an initial summary of each trajectory. They include the number of observations, duration of the flight, distance traveled, and communication channel used (e.g. radio-frequency, Wi-Fi).

Capability features include physical attributes (e.g., weight, dimensions) and performance specifications (e.g., maximum payload, battery capacity). These features are critical for assessing the drone's operational limits and the potential threat it may pose.

Altitude features,(e.g., the mean altitude above takeoff) and *speed features* (e.g., minimum/maximum speeds) provide insight into the dynamics of each trajectory. They are essential for detecting suspicious

[4]Drone locations may be acquired at irregular intervals.

[5]Our Online Appendix describes all the features.

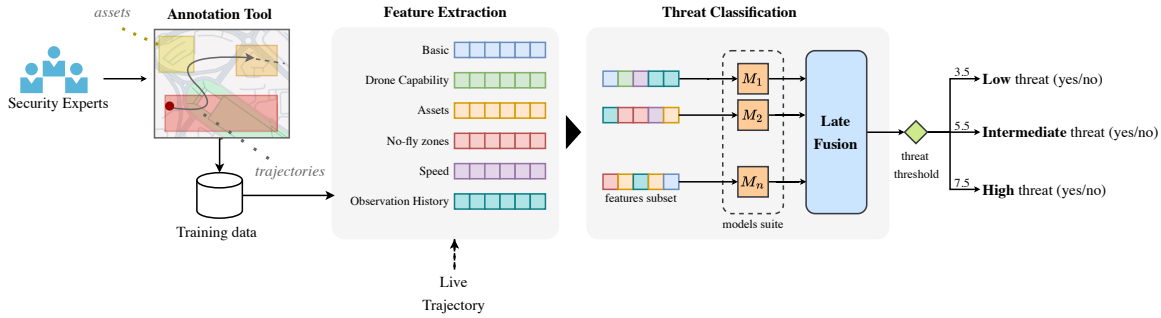

FIGURE 2: DEWS Architecture. Data set preparation involves annotating *asset values* and *drone trajectories* by *police*. Subsequently, DEWS extracts *features* and trains 11 classifiers $M_1, \cdots, M_{11}$ to yield 11 predictions which are integrated using *late fusion* to predict the final *threat level*. During operational use (after training), an initial part of a *live trajectory* is processed to extract features, and the combination of single predictors and late fusion produces the final threat score.

activities and in ensuring regulatory compliance as drones may have altitude or speed restrictions.

No-fly Zone features capture the behavior of trajectories in terms of their respect for the law. We used no-fly zone data[6], and defined six features to quantify the proximity of the trajectory to a no-fly zone, e.g., whether the drone entered a no-fly zone, the percentage of time the trajectory was within a no-fly zone.

Asset-based features (e.g. dams, utilities, government buildings, defense sites) also need to be considered when assessing the threat of a trajectory. We defined features about the proximity of the trajectory to these assets (e.g. the maximum/mean asset values overflown). Asset values were provided by Dutch police.

Observation history features capture the similarity between the current trajectory and historical trajectories. *Self-similarity features* refer to the similarity between the current trajectory and past trajectories of the same drone, which can help detect recurring flight patterns or behaviors that may indicate potentially benign operations. *Cross-similarity features* capture the similarity between the current trajectory and past trajectories of other drones. This may be useful for identifying anomalous behavior by comparing it to known suspicious or dangerous flight patterns exhibited by other devices. Cosine similarity is used in both.

Threat Classification

The Threat Classification module predicts the threat level (low, medium, high) of a trajectory based on its extracted features. To accomplish this, we trained a suite of 11 well-known machine learning classifiers, encompassing both traditional and neural network models.[7]

For each classifier, we did hyper-parameter optimization and applied feature selection to identify the most relevant subset of features. The feature selection process consists of three main steps: (1) removing constant columns, (2) retaining only one feature from pairs with a Pearson correlation greater than 0.95, and (3) selecting the top-k features based on their Mutual Information (MI) scores, where k is a user-defined parameter. This approach allowed us to develop specialized models that leverage distinct subsets of features for the same trajectory, thereby enhancing model diversity within the suite.

After individually training each model M_i, we used late fusion to combine their predictions. The final threat score for a trajectory t is computed as a weighted sum of the probability estimates produced by each model: $y(t) = \sum_{i=1}^{11} M_i(t) \cdot w_i$, where $M_i(t)$ represents the probability prediction of model M_i for trajectory t, and w_i denotes the weight assigned to model M_i. The weights w_i were optimized through grid search to identify the combination of weights that maximized overall classification performance. This fusion process enables DEWS to integrate the strengths of multiple models, ensuring robust and accurate threat classification.

[6] https://www.godrone.nl

[7] The classifiers used in DEWS are: Logistic Regression, k-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Trees, Random Forest, Gradient Boosting, Naive Bayes, AdaBoost, Extra Trees, a Multi-layer Perceptron (MLP), and a wide-and-deep neural network.

Experiments

All experiments were conducted on a computational platform having a 9th Gen Intel i9-10980XE processor, 256 GB of RAM, and an NVIDIA RTX A6000. The codebase involved approximately 2000 lines of code in Python 3.10. All classification models were implemented using the Scikit-learn library, expect for the wide and deep classifier for which we used the Tensorflow 2 library.

Data Collection

Data about 349 drone trajectories over a Dutch city was systematically collected by the Dutch police using the Senhive[8] commercial drone tracking system. This system tracks drones by monitoring their communication frequencies with drone operators, allowing for the detection and recording of their trajectories within a radius of 25 km. An anonymized version of this dataset was provided to the academic part of our team, with sensitive information such as device IDs replaced with anonymized IDs. Summary statistics for the dataset are provided in Appendix.

We developed our own GUIs for annotating asset values and threat scores associated with the drone trajectories. **When this paper is published, we will release anonymized versions of the DEWS data.**

Experimental Protocol

In our experiments, we address the DTPP problem at three distinct levels: 3, 5 and 7. This corresponds to the scenarios detailed as follows:

(i) *Low-Threat Prediction* (LTP): trajectories with a threat score in the $[3, 10)$ range (i.e. greater than or equal to 3 and strictly less than 10) are considered low threats. The LTP problems predicts no-threat (score less than 3) and low threat (score greater than 3).

(ii) *Medium-Threat Prediction* (MTP): trajectories with a threat score in the $[5, 10)$ are considered medium threat trajectories. So MTP distinguishes between medium threats (score of 5 or more) and other trajectories.

(iii) *High-Threat Prediction* (HTP): trajectories with a threat score greater than or equal to 8 are classified as threatening, while trajectories with a score of 7 or less are classified as non-threatening.

By applying the learned predictive models for a given trajectory, we can uniquely classify a trajectory into one of the four threat levels (no threat, low, medium, high threat).

These classification tasks are increasingly difficult due to the skewed distribution of threat labels, with the HTP setting containing significantly fewer threatening trajectories compared to MTP and LTP.

We conducted three experiments:

- *Early Threat Prediction Evaluation*: We assess DEWS's capability for early threat prediction by varying the observation window for each trajectory. Specifically, we analyze each trajectory t using the first i seconds of a flight, where $i \in \{1, 5, 10, 20, 30, 60, 180, 360, 720\}$. This assesses how early accurate predictions about the potential threat can be made.
- *Ablation Study*: To determine the relative importance of different feature types, we systematically remove each feature type from the model and retrain the DEWS[lf] late fusion predictor. Performance is then evaluated based on recall, precision, and F1-scores to identify which features contribute most significantly to predictive accuracy.
- *Feature Relevance Analysis*: Assuming that features selected for classification are the most relevant for solving the task, we analyze the features chosen by each classifier during the feature selection process. For each observation window, we count how often each attribute is selected for classification across all classifiers in the model suite. These counts are then normalized to compute the relative frequency of each feature category. Specifically, let w denote an observation window, $\mathcal{A} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n\}$ represent the set of features and $M_1, M_2, \ldots, M_{11}$ be the classifiers in the model suite. We define:
  - $N_{ij}^{(w)}$ as the count of how often the feature $\mathcal{F}_i$ is selected for classifier M_j during M_j's features optimisation process for classification, within window w;
  - $N_i^{(w)}$ as the total count of how often the feature $\mathcal{F}_i$ is selected across all classifiers, i.e.
$$N_i^{(w)} = \sum_{j=1}^{11} N_{ij}^{(w)}.$$

  To compute the relative frequency $f_i^{(w)}$ of the feature $\mathcal{F}_i$ for the observation window w, we normalize $N_i^{(w)}$ by the total counts for all attributes:
$$f_i^{(w)} = \frac{N_i^{(w)}}{\sum_{k=1}^{n} N_k^{(w)}}.$$

[8] https://senhive.com/sen-id-1

- *Runtime*: We measure DEWS's runtime for feature extraction and classification with late fusion in operational use (after training).

All experiments were conducted using time series cross-validation, i.e. we learned a model from an early set of trajectories and then used them to predict on later sets of trajectories.

Results

Early Threat Prediction Evaluation

Figure 3 illustrates DEWS's performance under the HTP setting. Specifically, Figures 3a, 3b, and 3c depict precision, recall, and F1-score, respectively, for all 11 classifiers as well as the DEWS late fusion classifier, DEWS[lf]. These metrics are analyzed by varying the observation window. Performance comparisons under MTP and LTP settings are reported in the Appendix.

Finding 1: Late Fusion is the Best Predictor. Late fusion consistently outperforms the 11 classifiers across all observation windows, achieving the highest results in terms of precision, recall, and F1-score. As shown in Figure 3, within an observation window of 1-30 seconds, DEWS[lf] stabilizes at an F1-score of approximately 80%, with precision exceeding 90% and recall around 75%. As the observation window increases, performance shows an upward trend, with the most substantial improvement occurring between one minute and three minutes. The best performance is achieved at the six-minute threshold, where precision reaches 0.967 and recall 0.869.

Finding 2: Increasing the Observation Window may Not Improve Performance. Interestingly, increasing the observation window does not always lead to improved performance. For example, Figure 3 shows that the highest recall of 0.789 for shorter observation windows occurs with 5 seconds of observation, when precision is 0.934 (using our DEWS[lf] classifier). Both metrics show a slight decline when the window is extended up to 30 seconds. Moreover, beyond six minutes, performance deteriorates across all models and metrics.

Finding 3: Precision is always higher than recall. Figure 3 shows that the same time thresholds yield higher performance in terms of precision compared to recall. For instance, with a short observation window of 5 seconds, precision reaches 0.934, while recall is comparatively lower at 0.789. This trend is consistently observed across all observation windows. This is due to the imbalance of the data considered for the HTP problem which causes DEWS to be more conservative when predicting the the highly threatening (minority) class. This suggests that DEWS is very accurate at detecting highly threatening trajectories with a very low false positive rate.

This is extremely valuable for police for two critical reasons: First, it enhances trust in the system, as the low false positive rate minimizes the likelihood of unnecessary interventions. Second, in resource-constrained environments, human assessment of predicted high threat can be costly and inefficient. High precision ensures that humans don't get frustrated with false positives.

Ablation Study

Figures 4a, 4b and 4c show the F1-scores obtained when removing individual feature categories under the LTP, MTP, and HTP settings, respectively.

Finding 4: Asset-related features are the most critical for threat prediction. We see from Figure 4 that with a 5-second observation window, the F1-score with all features included is 0.723 in the HTP setting, but decreases to 0.586 when asset features are excluded, representing a 19% reduction in performance.

Equally surprising are the features that proved to be less important than we had expected. For example, we initially hypothesized that no-fly zone features would play a significant role in threat prediction, yet they had a relatively minor impact on the model's performance. Similarly, we expected the type of drone (e.g., fast drones with large payloads) to be a key predictor, but their importance for prediction was small. Additionally, speed-related features, which we assumed would be important, turned out to have limited significance in our experiments.

Overall, these findings support our preliminary hypothesis that the geographical region, represented by asset-related features, is a key determinant in assessing the threat level of a trajectory, independent of the drone's intrinsic characteristics or the specific properties of the trajectory itself.

Feature Relevance Analysis

The results in Figure 5 (HTP problem) indicate that as the observation window increases, the importance of asset-related features becomes more pronounced. For instance, after a 180 second observation window, over 60% of the features used for classification belong to the asset category. Interestingly, within the first 1-5 seconds of observation, capability-related features show relatively high importance. The importance of these features decreases sharply with longer observation windows. This may be due to the limited information available in short observation windows, where the drone's capabilities alone serve as a strong indicator of potential threat.

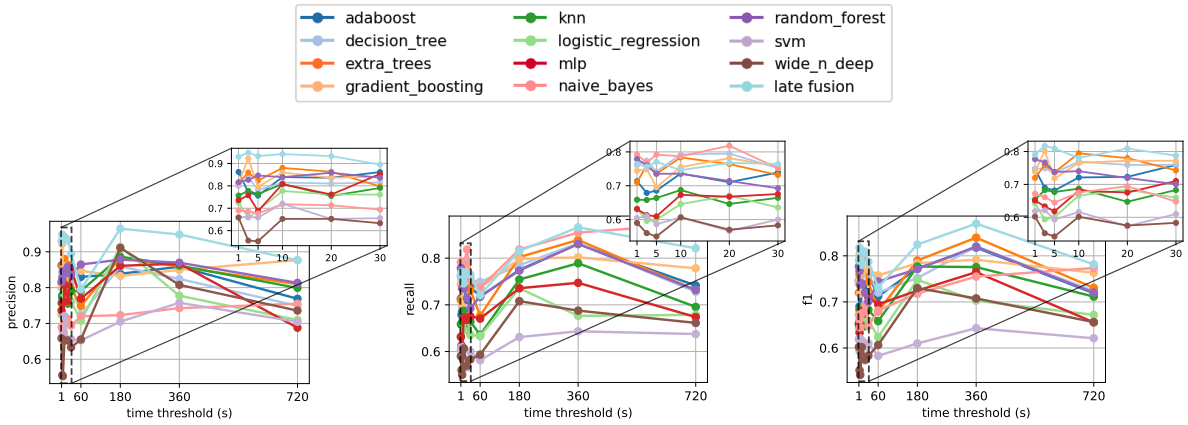

FIGURE 3: High-Threat Prediction (HTP) settings: Precision (a), Recall (b), and F1-score (c) metrics are shown as functions of varying temporal restrictions on the trajectories. The top row provides a zoomed-in view of the results for shorter time windows (less than 30 seconds), while the bottom row displays the complete range of observation windows.

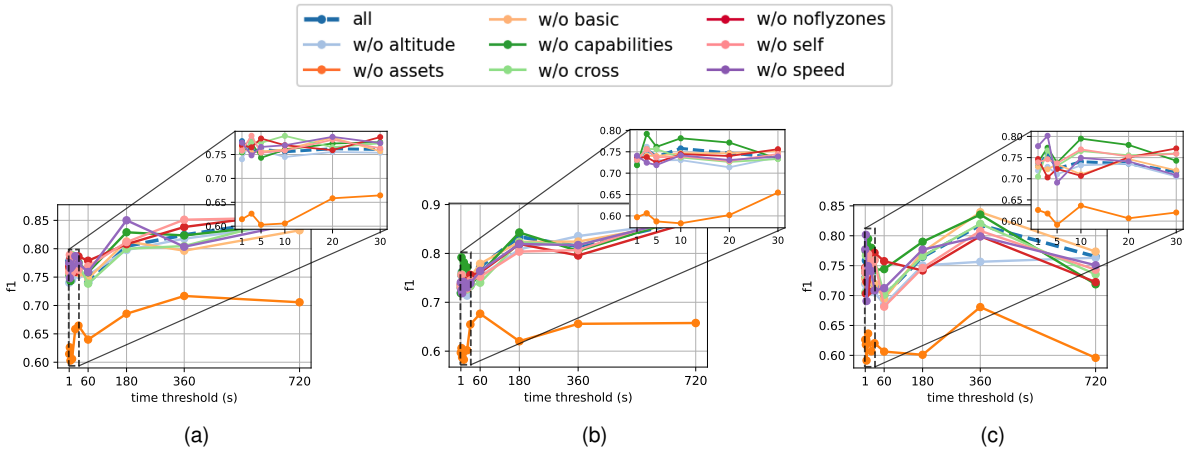

FIGURE 4: Ablation Study: F1-score under LTP (a), MTP (b) and HTP (c) settings when removing one feature category. The dashed line represents the scenario with all features.

Runtime

Figure 6 shows the mean DEWS runtime for feature extraction and prediction (with late fusion) under the HTP setting in operational use. The feature extraction time shows a slight increase with a larger observation window, reflecting the additional computational load due to the increased number of trajectory points. In contrast, the prediction time is not affected by the length of the trajectory. With an overall classification time of approximately 3 seconds, the DEWS system demonstrates its potential for real-time predictions, enabling trajectory classification after just 3-5 seconds of observation. This highlights the system's suitability for applications requiring prompt decision-making.

Limitations and Future Work

Like all studies, our study can be improved in many ways. First, we note that we looked at all trajectories over a city that Dutch police tracked over an 8-month period. But these may not reflect *all* possible flights because of limitations in tracking technology. Second, once adversaries know about DEWS, they may take evasive actions to prevent their intentions being predicted. To some extent, this is mitigated by our finding that asset value is far and away, the important feature in assessing threat - and adversaries cannot manipulate that. But the development of ML models that are more robust to an adversary's evasion attempts need to be studied as a next step. Third, there is

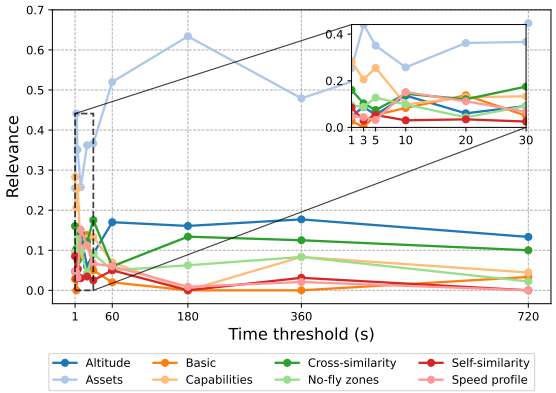

FIGURE 5: Feature Relevance Analysis (HTP problem): relative frequency of feature categories selected by classifiers across different temporal restriction windows.

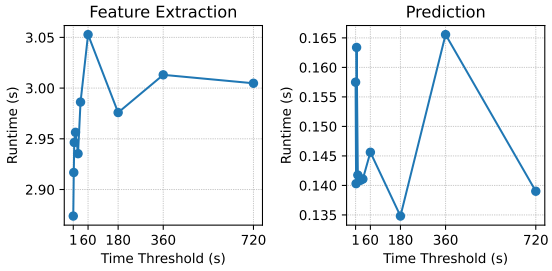

FIGURE 6: Runtime Analysis (HTP problem): Time (in seconds) for feature extraction (left) and prediction using late fusion (right).

the possibility of collusion. Two or more trajectories that individually seem non-threatening might collude to pose a significantly higher threat.

Conclusion

To the best of our knowledge, this is the first paper to explicitly study the problem of how threatening a drone flight is to a city or geographic region. We propose a repertoire of features for quantifying the threat of a drone flight, build out the first drone threat dataset that was assessed by police and security officials and will be made publicly available (with some anonymization to ensure security), and build the first predictive models to assess the threat level posed by a trajectory. We are also the first to show that we can predict threat levels *early*, when a trajectory is just underway. With just 5-10 seconds of trajectory data, DEWS is able to make predictions of high threat levels with an F1-score over 0.8. And these predictions take only a few seconds to make. Predictive accuracy goes up till about 5-6 minutes of the trajectory is observed. This enables DEWS to continuously provide forecasts to security officials after 5 seconds of the flight is observed and they can decide on their response depending on their own judgement and knowledge of context. DEWS also allows predictions to be tailored to a specific context and threat assessment. In other words, given a specific threat assessment, particular assets (on the ground) or capabilities (of drones) may be valued differently — and DEWS will still work.

Our biggest new finding is that the key determinant of the danger posed by a trajectory is not the trajectory itself, but the values of the assets on the ground that a trajectory flies over.

REFERENCES

1. A. Almohammad and A. Speckhard, "Isis drones: Evolution, leadership, bases, operations and logistics," *The International Center for the Study of Violent Extremism*, vol. 5, 2017.
2. O. ŞEN and H. AKARSLAN, "Terrorist use of unmanned aerial vehicles: Turkey's example." *Defence Against Terrorism Review*, vol. 13, 2020.
3. K. L. Best, J. Schmid, S. Tierney, J. Awan, N. M. Beyene, M. A. Holliday, R. Khan, and K. Lee, *How to Analyze the Cyber Threat from Drones: Background, Analysis Frameworks, and Analysis Tools*. Santa Monica, CA: RAND Corporation, 2020.
4. F. Ullah, S. M. Sepasgozar, and C. Wang, "A systematic review of smart real estate technology: Drivers of, and barriers to, the use of digital disruptive technologies and online platforms," *Sustainability*, vol. 10, no. 9, p. 3142, 2018.
5. M. Thompson, A. A. Tarr, J.-A. Tarr, and S. Ritterband, "Unmanned aerial vehicles: Liability and insurance," in *The Global Insurance Market and Change*. Informa Law from Routledge, 2024, pp. 212–245.
6. C.-W. Wu, M.-D. Shieh, J.-J. J. Lien, J.-F. Yang, W.-T. Chu, T.-H. Huang, H.-C. Hsieh, H.-T. Chiu, K.-C. Tu, Y.-T. Chen, S.-Y. Lin, J.-J. Hu, C.-H. Lin, and C.-S. Jheng, "Enhancing fan engagement in a 5g stadium with ai-based technologies and live streaming," *IEEE Systems Journal*, vol. 16, no. 4, pp. 6590–6601, 2022.
7. V. Kulkarni, A. Moro, and B. Garbinato, "Mobidict: A mobility prediction system leveraging realtime location data streams," in *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*, 2016, pp. 1–10.
8. A. M. Raivi, S. A. Huda, M. M. Alam, and S. Moh, "Drone routing for drone-based delivery systems: A

review of trajectory planning, charging, and security," *Sensors*, vol. 23, no. 3, p. 1463, 2023.

9. H. Georgiou, S. Karagiorgou, Y. Kontoulis, N. Pelekis, P. Petrou, D. Scarlatti, and Y. Theodoridis, "Moving objects analytics: Survey on future location & trajectory prediction methods," *arXiv preprint arXiv:1807.04639*, 2018.
10. A. Poibrenski, M. Klusch, I. Vozniak, and C. Müller, "M2p3: multimodal multi-pedestrian path prediction by self-driving cars with egocentric vision," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 190–197.
11. F. Chen, X. Wang, Y. Zhao, S. Lv, and X. Niu, "Visual object tracking: A survey," *Computer Vision and Image Understanding*, vol. 222, p. 103508, 2022.
12. V. Bharilya and N. Kumar, "Machine learning for autonomous vehicle's trajectory prediction: A comprehensive survey, challenges, and future research directions," *Vehicular Communications*, p. 100733, 2024.
13. L. Lin, W. Li, H. Bi, and L. Qin, "Vehicle trajectory prediction using lstms with spatial–temporal attention mechanisms," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 2, pp. 197–208, 2021.
14. K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Attention based vehicle trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 175–185, 2020.
15. Y. Ren, Z. Lan, L. Liu, and H. Yu, "Emsin: enhanced multi-stream interaction network for vehicle trajectory prediction," *IEEE Transactions on Fuzzy Systems*, 2024.
16. J.-H. Kim and D.-S. Kum, "Threat prediction algorithm based on local path candidates and surrounding vehicle trajectory predictions for automated driving vehicles," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1220–1225.
17. R. Bonatti, W. Wang, C. Ho, A. Ahuja, M. Gschwindt, E. Camci, E. Kayacan, S. Choudhury, and S. Scherer, "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making," *Journal of Field Robotics*, vol. 37, no. 4, pp. 606–641, 2020.
18. V. Ajith and K. Jolly, "Unmanned aerial systems in search and rescue applications with their path planning: a review," in *Journal of Physics: Conference Series*, vol. 2115, no. 1. IOP Publishing, 2021, p. 012020.
19. M. Ozer, I. Keles, İ. H. Toroslu, P. Karagoz, and S. Ergut, "Predicting the next location change and time of change for mobile phone users," in *proceedings of the third ACM SIGSPATIAL international workshop on mobile geographic information systems*, 2014, pp. 51–59.
20. M. Ozer, I. Keles, I. H. Toroslu, and P. Karagoz, "Predicting the change of location of mobile phone users," in *Proceedings of the Second ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, 2013, pp. 43–50.