Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2023)
19-20 July 2023, Tenerife, Canary Islands, Spain
DOI: 10.1109/ICECCME57830.2023.10252859

Hybrid Deep Neural Networks for Drone High Level Intent Classification using Non-Cooperative Radar Data

Benjamin Fraser
Sch. of Aerospace Transport and Manufacturing
Cranfield University
Bedford, UK
b.fraser@cranfield.ac.uk

Adolfo Perrusquía
Centre for Autonomous and Cyberphysical Systems
Cranfield University
Bedford, UK
adolfo.perrusquia-guzman@cranfield.ac.uk

Dimitrios Panagiotakopoulos
Centre for Autonomous and Cyberphysical Systems
Cranfield University
Bedford, UK
d.panagiotakopoulos@cranfield.ac.uk

Weisi Guo
Centre for Autonomous and Cyberphysical Systems
Cranfield University
Bedford, UK
weisi.guo@cranfield.ac.uk

***Abstract*—The proliferation of drones has brought many benefits in different industrial and government sectors due to their low cost and potential applications. Nevertheless, the security and air space can be compromised due to anomalous performances derived to negligence or intentional malicious activities. Thus, identify the hidden intentions of drones' mission profiles is paramount to execute adequate countermeasures. In this paper, an hybrid deep neural network architecture is proposed to classify the high level intent of drones' mission profiles using non-cooperative radar. Radar measurements are created synthetically using open access telemetry data of flight trajectories. The proposed architecture exploits the classification and reconstruction capabilities of deep neural models to classify the drones hidden high-level intent. Several experiments and comparisons are carried out to verify the effectiveness of the proposed approach.**

***Index Terms*—drones, high-level intent, classification, novelty detection, deep neural networks**

I. Introduction

Research in detection and classification of drones is mainly focused on different detection methods for counter drone applications [1]. The most used methods encompass visual-based methods based on infrared (IR), acoustic, radar, and radio frequency (RF) [2]. In this paper, we focus on radar detection methods since it provides useful information regarded to the drone intent in comparison to visual methods.

Radar technologies have been used for drone classification [3] due to its high accuracy, range and classification capabilities. The exploitation of Doppler spectra is crucial in modern radar technologies to detect and classify objects. The key idea is to use micro-Doppler signatures to recognize profiles of objects and their moving components, e.g., drone propellers, limb motions of humans, or flapping wings from birds.

This work was supported by the Royal Academy of Engineering and the Office of the Chief Science Adviser for National Security under the UK Intelligence Community Postdoctoral Research Fellowship programme.

The spectrogram [4]–[8] is used as a feature representation in classification tasks. This generates a visual representation that can be exploited by machine learning or deep learning models, e.g, convolutional neural networks (CNNs) to detect spatial patterns associated to the profile [8]–[10], or recurrent neural networks (RNNs) [11], [12] to model time-dependencies from consecutive visual frames. Other visual representations that can be further used includes: the range Doppler matrix [13], the cepstrogram [5], [14], or the Cadence velocity diagram (CVD) [14] which can enhance current drone detection and classification systems.

Some additional features that can enhance the classification accuracy encompass the payload type and weight [6], [14]. For instance, these additional information provide prior knowledge of the potential flight purpose (e.g., delivery or inspection) and the possible actions that the drone could take [15]. Furthermore, distinguish between different types of drones such as fixed or rotary-wing [16] can help in the design of accurate inference models [17] for trajectory tracking of any mission profile, e.g., Kalman filter and their variants.

Most of the aforementioned works focus on drone detection and classification (e.g., drones vs birds or fixed-wing vs rotary-wing, etc.). However, the prediction of the hidden intent of the drone mission profile is an open gap that is currently taking attention for several research communities. In this paper, intent refers to the main objective that the drone aims to achieve based on observations of its flight profile. Drones intent includes: i) higher-level flight purpose [18] that defines the task of the mission profile (e.g., surveillance, delivery, inspection), ii) trajectory intent which defines the final goal destination (e.g., the location of an airport), and iii) objective function inference [19] associated to set of actions that the drone needs to perform to achieve a desired task.

In the literature, several works have been developed to classify intent in the field of robotics, autonomy and smart

navigations. This includes predicting pedestrians intent or lane changing for the design of safe self-driving car systems [20]–[22]. However, any of the aforementioned methods focused on drone intent classification based on sequential data. To best of our knowledge, classifying drone mission profiles into their high-level intent represents a novel approach using either supervised or unsupervised learning methodologies. In this paper, we adopt a supervised learning strategy to address the above gaps through the design of trajectory classification models based on deep neural architectures. The proposed approach consists in two deep neural structures for trajectory classification and intent anomaly detection. Time-series data obtained from diverse high-level intents are used and preprocessed to feed the neural models for different time-window sizes. The results of the approach are compared against different supervised learning classifiers of time-series trajectories.

The contributions of this work in comparison to previous approaches are:

1) A data generation architecture that converts telemetry data to simulated radar data under radar location and noise assumptions.
2) A deep neural classifier which predicts the high-level intent class of drones trajectories under different time-windows length.
3) A novelty detector architecture based on an encoder-decoder deep neural model that determines if a trajectory is normal or anomalous in terms of the well-identified drones' high-level intent.

II. Data Preparation and Radar Simulation

In this paper, drone high-level intent is referred to the purpose of use of the drone [23], e.g., surveillance, inspection, delivery, etc. We focus on four high-level intent classes due to the low number of open-access datasets of drones mission profiles, however the approach can easily be extended for n distinct classes. The four high-level intent classes used in this paper cover: i) perimeter inspection or surveillance, ii) point-to-point transit flights, iii) package delivery, and iv) area mapping and surveying. These classes are obtained from telemetry data of open access sources available in [24]–[27]. The collected telemetry data [28] obtained from GPS and inertial navigation system (INS) measurements are transformed into simulated radar data and track estimations to emulate real radar measurements and trajectory inference [17]. This step is crucial to demonstrate the capability of the approach to be integrated in real-world radar systems.

The simulated radar data depend on two factors: 1) the assumptions of the noise distribution and 2) the proposed radar location relative to the mission profile. By modifying these elements, we are able to generate a pool of heterogeneous simulated radar data from a single mission profile. This serves as a data generation process to create a large enough dataset to train and test the intent classifier.

The transformation of the telemetry data is performed with less than 400 real drones flights with predefined high-level intent classes. The real data encompasses several types of drones, tasks, and different parameters [29]. Moreover, additional flights are carry out using a DJI Mavic Air 2 to complement the perimeter and mapping intent classes. The telemetry data of each flight include longitude, latitude and altitude, which are converted into Cartesian coordinates [23] to avoid location bias. Furthermore, each flight class has different sampling time and hence, each dataset were down/up-sampled as required to standardised the sample time to 1 Hz.

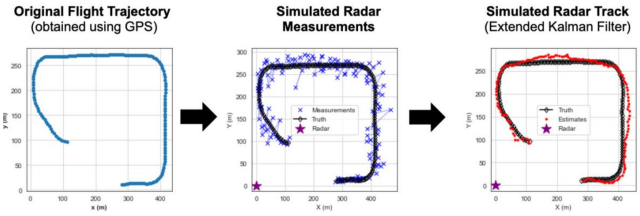


Fig. 1. Simulated radar data and trajectory inference

Two noise intensities are used to generate different tracking results [16]. The noise intensities are set to 1 and 3. Seven different locations are proposed for the radar setting which produce different simulated radar results despite of using the same mission profile. After the simulated radar data is obtained, a trajectory inference algorithm is used to estimate the trajectory from the simulated radar measurements. An extended Kalman Filter (EKF) is used as trajectory inference algorithm. The likelihood function of the EKF is a nearly-constant velocity transition model [30] modelled as a linear time-variant (LTV) Gaussian model with constant velocity per dimension. The model for the x-axis is given by

$$\boldsymbol{x}_t = \boldsymbol{F}_t \boldsymbol{x}_{t-1} + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_t), \tag{1}$$

with

$$\boldsymbol{x} = \begin{bmatrix} x_{pos} \\ x_{vel} \end{bmatrix}, \quad \boldsymbol{F}_t = \begin{bmatrix} 1 & d_t \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{Q}_t = \begin{bmatrix} \frac{dt^3}{3} & \frac{dt^3}{2} \\ \frac{dt^3}{2} & dt \end{bmatrix} q, \tag{2}$$

where x_{pos} is the x-axis Cartesian position, x_{vel} is the x-axis linear velocity, d_t is the sampling time, and q is the velocity noise diffusion constant. The same model holds for the other dimensions. Fig. 1 depicts an example of the tracked trajectory obtained from the simulated radar measurements.

III. Proposed Methodology

Fig. 2 depicts the general scheme of the proposed high-level intent classifier. The scheme is composed of two main phases: i) data generation and preprocessing to create suitable training and testing sets to feed the intent classifier (see Fig. 1), and ii) an hybrid intent classifier composed of two deep neural networks schemes for classification and novelty detection. The hybrid intent classifier is discussed in the next sections.

A. Trajectory Intent Classification

We design several deep neural networks (DNN) classifiers to exploit their capabilities to extract high-dimensional features

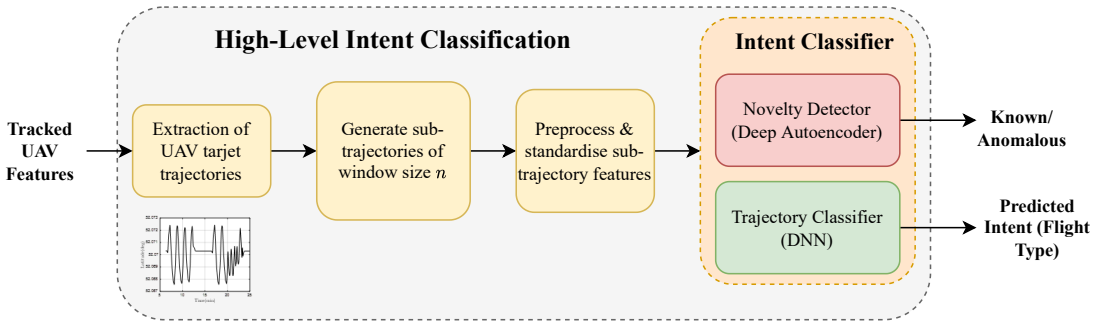


Fig. 2. High-level intent classification scheme

associated to each class. In addition, we design a random forest classifier as a baseline to compare the performance of each DNN. Here, it is important to mention that random forests cannot work directly with multi-dimensional time-series sequences. Hence, a set of summary features are constructed from the means and standard deviations of each sub-trajectory to train this classifier. Seven neural classifiers are considered in this paper, which includes: long-short term memory (LSTM), gate recurrent units (GRU), Convolutional bidirectional-LSTM (CNN-BLSTM), Convolutional bidirectional-LSTM with attention (CNN-BLSTMA), CNN, CNN with attention (CNNA).

Each neural classifier is trained under the categorical cross-entropy loss. Regularisation is applied to avoid overfitting using dense and recurrent dropout layers. In addition, in the training phase we add an early stopper to stop the learning phase when the validation loss does not show any improvement after several number of epochs [31]. This process helps to recognize the best weights that reduce the classification error without overfitting issues and, in consequence, to improve the generalization capabilities of the classifier.

B. Autoencoder Novelty Detector

The second element of the hybrid classifier is given by a deep autoencoder architecture. Here, the main goal of this network is to minimize the reconstruction error between the input sequences of the radar trajectories and the predicted output of the autoencoder. This task is performed by minimizing the following mean squared reconstruction error across all the radar simulated data and the autoencoder output predictions

$$\mathcal{L}_{ae}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) = \frac{1}{NMT} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{t=1}^{T} (x_{ijt} - \widehat{x}_{ijt})^2, \quad (3)$$

subject to $\boldsymbol{X} \in \mathbb{R}^{N \times M \times T}$ and $\widehat{\boldsymbol{X}} \in \mathbb{R}^{N \times M \times T}$, where N is the number of mission profiles, M is the number of features, T is the length of the time sequence, $\boldsymbol{X}$ is the input trajectory tensor, $\widehat{\boldsymbol{X}}$ is the reconstructed trajectory tensor, and x_{ijt} is the trajectory value for the i-th mission profile, j-th feature, and t-th time step.

Trajectories that do not match with the predefined classes may exhibit a large MSE when they pass through the autoencoder architecture. This is because the network was trained to effectively reconstruct trajectories with specific low and high dimensional patterns, but when a novel trajectory with different patterns is given, then the reconstructed trajectory will be poor and, in consequence, a large MSE is obtained. Therefore, the reconstruction error can be used as a proxy indicator of a potential malicious drone. To ensure that most of the training data is considered as non-anomalous, we set a threshold in terms of the reconstruction errors, where trajectories with MSEs above this value are considered as anomalous. The threshold is computed using

$$\varepsilon = \mu_{train} + \alpha(\sigma_{train}), \quad (4)$$

where α is a threshold factor, μ_{train} and σ_{train} are the mean and standard deviation of the training data. Here, α serves as a weighting factor to determine how much an input trajectory can vary with respect to the training data to be considered non-anomalous, otherwise the trajectory will be anomalous since its variations are high. In this paper, we set $\alpha = 3$ to capture approximately the 99.9% of the training data as non-anomalous.

IV. Results and Comparisons

Each network is trained to ensure high prediction accuracy but without falling into the overfitting problem. Here, the core idea is to learn high-dimensional patterns associated to each flight class to enable generalization to unseen trajectories with similar patterns. To ensure generalization without overfitting, we use the recurrent and dense dropout layers for regularisation. In addition, we carefully observed the training and validation losses and apply the early stopping method to obtain the best model associated to each network.

Each model is trained under four different windows lengths of the input sequence. This allow to observe the prediction capabilities of each network under input trajectories of variable length. The proposed window lengths are set to 8, 16, 32, and 64, which can be increased depending of the length of the input trajectories. The hyperparameters are tuned using grid search until the best validation performance is obtained. The classification results across all input-window lengths are averaged and summarized in Table I.

The results show that all DNN methods outperform the random forest classifier by exploiting the high-dimensional encoding capabilities of deep neural architectures. The basic

TABLE I
CLASSIFICATION RESULTS AVERAGED ACROSS ALL WINDOW SIZES. BEST RESULTS ARE IN BOLD

Model	Validation				Test			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
Random Forest	0.8756	0.8033	0.7492	0.7590	0.9205	0.9261	0.8842	0.8933
LSTM	0.9105	0.8623	0.8611	0.8599	0.9591	0.9550	0.9630	0.9588
GRU	0.9299	0.9041	0.8868	0.8938	0.9554	0.9567	0.9380	0.9417
CNN-BLSTM	0.9420	0.9110	0.9099	0.9101	0.9775	0.9737	0.9832	0.9782
CNN-BLSTMA	**0.9467**	**0.9205**	**0.9230**	**0.9213**	**0.9795**	**0.9761**	**0.9844**	**0.9801**
CNN	0.9369	0.9009	0.9127	0.9063	0.9690	0.9633	0.9752	0.9689
CNNA	0.9437	0.9137	0.9148	0.9140	0.9756	0.9714	0.9805	0.9757

recurrent units (LSTM, GRU) show poor performances for small window lengths (8 and 16) but high performance for long window lengths (32 and 64) because more information is given to train the networks. The F1-score metric is used as the main important metric to evaluate the performance of each model across all window lengths. In addition, it is helpful to deal with class imbalance throughout the data partitions. An important aspect to take into account is the inference time that each model requires to predict the mission profile task. This is highly important since early predictions are required in real-time implementations to apply adequate countermeasures before the drone acts. The average of the training and prediction times are summarized in Table II.

TABLE II
COMPARISON OF CLASSIFIER TRAINING AND PREDICTION TIMES. BEST RESULTS IN BOLD.

Model	Training Time (s)	Mean Prediction Time(s)
Random Forest	**6.66**	**2.80e-6**
LSTM	191.34	3.63e-5
GRU	103.1	3.38e-5
CNN-BLSTM	111.29	3.35e-5
CNN-BLSTMA	242.91	3.36e-5
CNN	78.89	**1.69e-5**
CNNA	**56.99**	1.80e-5

The inference time results show that (without considering the random forest classifier) the CNN and CNNA require less computation time in comparison with the other neural models. This is due to the use of 1D convolutional layers that encode the most important regions of each sequence of the trajectory to feed the fully connected layers of the classifier. The CNN-BLSTM and CNN-BLSTMA have these 1D convolutional layers inside of their architectures, however the incorporation of the LSTM network increases the computation time whilst improving the prediction accuracy (see Table I). The time inference results show that the prediction times are small for each classifier and hence, they will not compromise the real-time implementation. In conclusion, the best model in terms of performance is the CNN-BLSTMA. It shows impressive results for all time-windows in the validation and test datasets. In addition, it has the best results in all metrics for the test dataset. The main drawback is the training time which is considerably bigger than CNN or CNNA models. However, this can be viewed as a compromise between accuracy and time which is highly acceptable in these experiments. Furthermore, the incorporation of an attention layer to each output of the LSTM layers enhance the performance of the CNN-BLSTMA and CNNA classifiers.

A. Hybrid Intent Classifier

The proposed high-level intent classifier is discussed in this section. The algorithm is a multi-modal architecture composed by a deep sequential model for trajectory classification and a novelty detector based on a deep autoencoder model.

Whilst standard classifiers only predicts the class of the flight profile within the set of classes, the proposed approach additionally provides an indicator to determine if the input trajectories matches with the learned patterns of the training data. We use the CNN-BLSTMA as the classifier of the hybrid architecture since it provides the best prediction accuracy (see Table I). An additional output is incorporated to this architecture to perform input reconstruction by using a LSTM decoder network. The hyperparameters of the classifier remain the same. The network is optimized using a composite multi-modal loss, which consists in a weighted average between the categorical cross-entropy loss (for classification) and the MSE loss (for novelty detection). The new loss is given by

$$\begin{aligned} \mathcal{L}_{hybrid} &= \alpha_{ae}\mathcal{L}_{ae}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) - \alpha_{clf}\mathcal{L}_{clf}(\boldsymbol{y}, \widehat{\boldsymbol{y}}), \\ \mathcal{L}_{clf}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) &= -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{C} y_i \log(\widehat{y}_i), \end{aligned} \tag{5}$$

where y_i is the true flight class, $\widehat{y}_i$ is the prediction of the flight class, $\alpha_{ae} \in (0, 1)$ and $\alpha_{clf} \in (0, 1)$ are the weighting factors for each loss component. In addition, the following constraint must hold

$$\alpha_{ae} + \alpha_{clf} = 1. \tag{6}$$

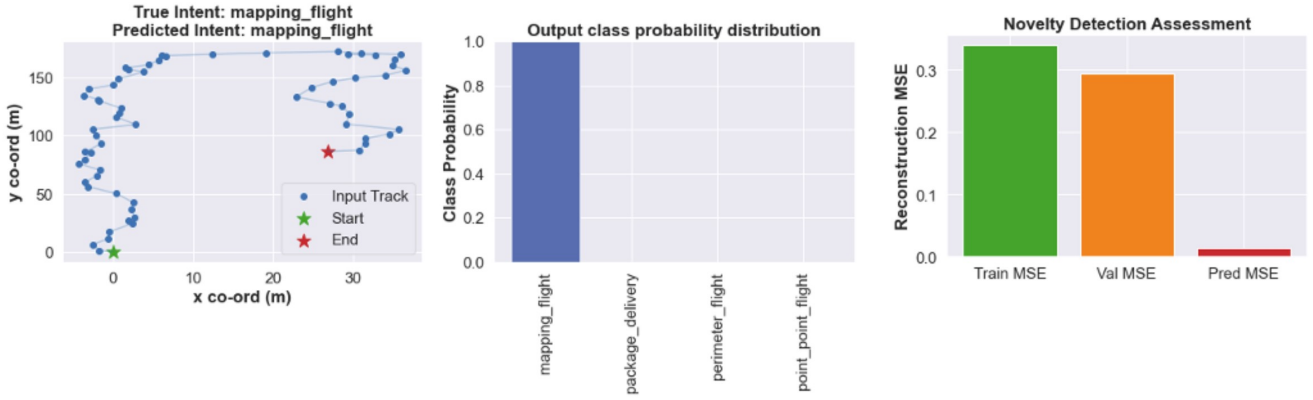
The validation loss of the classification model is monitored carefully to apply early stopping if the loss does not show any improvement after 20 epochs. Then the final classification model is saved for testing purposes. Notice that the classification loss of the validation set is used as criterion to stop the model instead of the reconstruction MSE. Here, the classification task is deemed as the most important task to optimize in the hybrid architecture.

Table III shows the results of the proposed hybrid classifier for different time-windows under the validation and test data sets. Here we can observe that competitive results are obtained in comparison to the results of Table I despite of the incorporation of the deep autoencoder model. Here, the hybrid model performance is close to the CNN-BLSTMA model performance which is relatively evident because the hybrid model classifier is based on the CNN-BLSTMA model. In terms of F1 score, the proposed model ranks third overall on the test data behind the CNN-BLSTMA and CNN-BLSTM, respectively. Furthermore, the proposed model ranks second overall the validation data (in terms of the F1 score). The main reason of the relatively degraded performance is due to the incorporation of the novelty detection architecture which enables to determine if an input trajectory is anomalous or not. Therefore, this trade-off between accuracy and anomaly detection is acceptable.

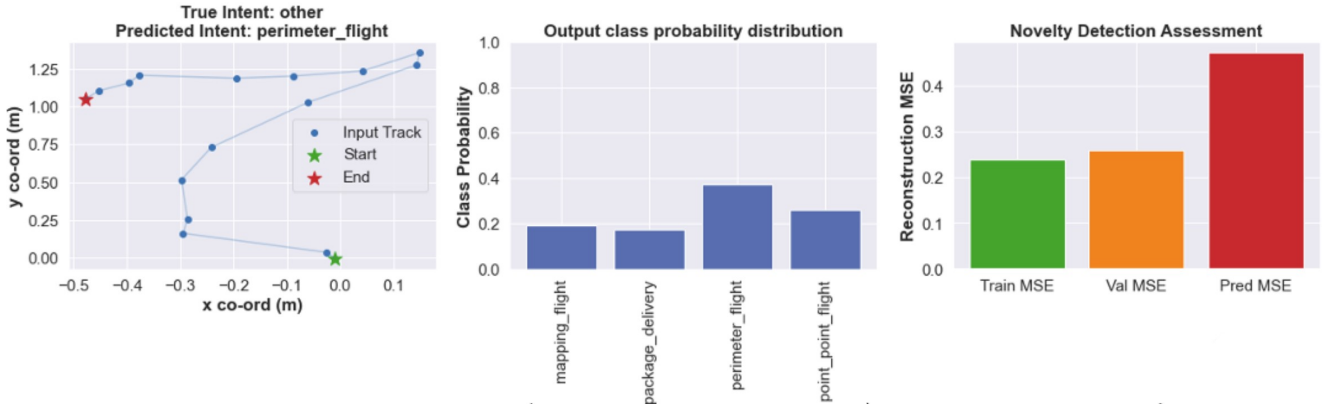
The final reconstruction errors are also provided for the training, validation, and testing data splits (see Table III). We

TABLE III
HYBRID CLASSIFIER & ANOMALY DETECTOR FINAL RESULTS

Model	Window Length	Training	Validation					Test					Other (anomalous flights)
		Recon MSE	Accuracy	Precision	Recall	F1 Score	Recon MSE	Accuracy	Precision	Recall	F1 Score	Recon MSE	Recon MSE
Hybrid Classifier & Novelty Detector	8	0.1801	0.9217	0.8867	0.8953	0.8906	1.0160	0.9591	0.9523	0.9637	0.9577	0.4308	4748758.0000
	16	0.1909	0.9359	0.9014	0.9005	0.9009	1.4313	0.9757	0.9714	0.9808	0.9759	0.4841	2368608.0000
	32	0.1993	0.9566	0.9258	0.9442	0.9344	1.7888	0.9851	0.9823	0.9896	0.9858	0.5145	1068317.7500
	64	0.2272	0.9664	0.9666	0.9301	0.9456	1.5193	0.9902	0.9881	0.9948	0.9913	0.3749	556330.9375
	Mean	**0.1994**	**0.9451**	**0.9201**	**0.9175**	**0.9179**	**1.4388**	**0.9775**	**0.9735**	**0.9822**	**0.9777**	**0.4511**	**2185503.6719**



(a) Known flight profile



(b) Unknown/Novel flight profile

Fig. 3. Intent classifier predictions on unseen test data examples

additionally provide new trajectories with different patterns to the training data denoted as "*other*" to test the approach to unseen flight classes. The results show that these new trajectories have high reconstruction error since they do not match with the learned patterns of the training data. This highlights the main advantage of the approach, that is, the reconstruction error serves as a proxy indicator to determine if a trajectory is anomalous or not.

It can be observed that the reconstruction errors for the anomalous trajectories decreases as the sequence length increases. This is caused due to the reduced number of data samples for large time windows length (e.g, 64). Then, we remove flight profiles with short length to avoid this issue.

B. Predictions on unknown trajectories

The trained hybrid network is tested with different drones trajectories under hidden mission profiles, which can either exhibit similar performances to the proposed classes or completely unknown trajectories that are regarded as anomalous. Fig. 3 shows the two possible scenarios.

On the one hand, when the trajectory poses a similar performance to the proposed high-level classes, then high probability is obtained from the classifier and the reconstruction error is small (see Fig. 3(a)). On the other hand, if the trajectories do not match with any of the proposed classes, then the classifier exhibits an approximately uniform distribution between all classes. This means that the classifier is not confident in the prediction and, in consequence, the reconstruction error is high. This result suggests that the input trajectory is anomalous and require further analysis (see Fig.

3(b)) and apply appropriate countermeasures.

V. Conclusions

This paper proposed an hybrid deep neural network model for high-level drone intent classification. Telemetry data obtained from open-access datasets are preprocessed to model real radar measurements under different locations. These data are used to create a large pool of heterogeneous high-level trajectories to train and test the network. The hybrid network is composed of: i) a sequential network classifier based on convolutional and recurrent neural networks with attention, and ii) a novelty detector based on a deep autoencoder network. The prediction results are promising to predict known flight profiles and recognize unknown flight profiles for further analysis. Future work encompasses the incorporation of additional features provided from radar technologies, e.g., doppler spectra, signal-to-noise ratios, RCS signatures, etc., to increase the prediction capabilities of the proposed architecture.

References

[1] S. Samaras, E. Diamantidou, D. Ataloglou, N. Sakellariou, A. Vafeiadis, V. Magoulianitis, A. Lalas, A. Dimou, D. Zarpalas, K. Votis *et al.*, "Deep learning on multi sensor data for counter uav applications—a systematic review," *Sensors*, vol. 19, no. 22, p. 4837, 2019.

[2] Y. Xiao and X. Zhang, "Micro-uav detection and identification based on radio frequency signature," in *2019 6th International Conference on Systems and Informatics (ICSAI)*. IEEE, 2019, pp. 1056–1062.

[3] J. S. Patel, F. Fioranelli, and D. Anderson, "Review of radar classification and rcs characterisation techniques for small uavs or drones," *IET Radar, Sonar & Navigation*, vol. 12, no. 9, pp. 911–919, 2018.

[4] F. Hoffmann, M. Ritchie, F. Fioranelli, A. Charlish, and H. Griffiths, "Micro-doppler based detection and tracking of uavs with multistatic radar," in *2016 IEEE radar conference (RadarConf)*. IEEE, 2016, pp. 1–6.

[5] R. Harmanny, J. De Wit, and G. P. Cabic, "Radar micro-doppler feature extraction using the spectrogram and the cepstrogram," in *2014 11th European Radar Conference*. IEEE, 2014, pp. 165–168.

[6] D. Dhulashia, N. Peters, C. Horne, P. Beasley, and M. Ritchie, "Multi-frequency radar micro-doppler based classification of micro-drone payload weight," *Frontiers in Signal Processing*, vol. 1, p. 14, 2021.

[7] M. Ritchie, F. Fioranelli, H. Borrion, and H. Griffiths, "Multistatic micro-doppler radar feature extraction for classification of unloaded/loaded micro-drones," *IET Radar, Sonar & Navigation*, vol. 11, no. 1, pp. 116–124, 2017.

[8] D. Park, S. Lee, S. Park, and N. Kwak, "Radar-spectrogram-based uav classification using convolutional neural networks," *Sensors*, vol. 21, no. 1, p. 210, 2020.

[9] S. Rahman and D. A. Robertson, "Classification of drones and birds using convolutional neural networks applied to radar micro-doppler spectrogram images," *IET radar, sonar & navigation*, vol. 14, no. 5, pp. 653–661, 2020.

[10] T. Kacker, A. Perrusquia, and W. Guo, "Multi-spectral fusion using generative adversarial networks for uav detection of wild fires," in *2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2023, pp. 182–187.

[11] W. Yu and A. Perrusquia, *Human-Robot Interaction Control Using Reinforcement Learning*. John Wiley & Sons, 2021.

[12] A. Perrusquía and W. Yu, "Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview," *Neurocomputing*, vol. 438, pp. 145–154, 2021.

[13] I. Roldan, C. R. del Blanco, Á. Duque de Quevedo, F. Ibañez Urzaiz, J. Gismero Menoyo, A. Asensio López, D. Berjón, F. Jaureguizar, and N. García, "Dopplernet: a convolutional neural network for recognising targets in real scenarios using a persistent range–doppler radar," *IET Radar, Sonar & Navigation*, vol. 14, no. 4, pp. 593–600, 2020.

[14] J. S. Patel, C. Al-Ameri, F. Fioranelli, and D. Anderson, "Multi-time frequency analysis and classification of a micro-drone carrying payloads using multistatic radar," *The Journal of Engineering*, vol. 2019, no. 20, pp. 7047–7051, 2019.

[15] A. Perrusquía and W. Guo, "Performance objective extraction of optimal controllers: A hippocampal learning approach," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2022, pp. 1545–1550.

[16] P. Sévigny, D. Kirkland, X. Li, and B. Balaji, "Unmanned aircraft (ua) telemetry data for track modelling and classification," in *STO Meeting Proceedings*, 2021.

[17] A. Perrusquía and W. Guo, "A closed-loop output error approach for physics-informed trajectory inference using online data," *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 1379–1391, 2023.

[18] J. Liang, B. I. Ahmad, M. Jahangir, and S. Godsill, "Detection of malicious intent in non-cooperative drone surveillance," in *2021 Sensor Signal Processing for Defence Conference (SSPD)*. IEEE, 2021, pp. 1–5.

[19] A. Perrusquía and W. Guo, "Reward inference of discrete-time expert's controllers: A complementary learning approach," *Information Sciences*, vol. 631, pp. 396–411, 2023.

[20] T. Su, Y. Meng, and Y. Xu, "Pedestrian trajectory prediction via spatial interaction transformer network," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE, 2021, pp. 154–159.

[21] K. Saleh, M. Hossny, and S. Nahavandi, "Intent prediction of pedestrians via motion trajectories using stacked recurrent neural networks," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 414–424, 2018.

[22] G. He, X. Li, Y. Lv, B. Gao, and H. Chen, "Probabilistic intention prediction and trajectory generation based on dynamic bayesian networks," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 2646–2651.

[23] A. Perrusquía and W. Guo, "Closed-loop output error approaches for drone's physics informed trajectory inference," *IEEE Transactions on Automatic Control*, 2023.

[24] J. Whelan, T. Sangarapillai, O. Minawi, A. Almehmadi, and K. El-Khatib, "Uav attack dataset," 2020. [Online]. Available: https://dx.doi.org/10.21227/00dg-0d12

[25] A. Keipour, M. Mousaei, and S. Scherer, "Alfa: A dataset for uav fault and anomaly detection," *The International Journal of Robotics Research*, vol. 40, no. 2-3, pp. 515–520, 2021.

[26] M. Street, "Drone identification and tracking," 2021. [Online]. Available: https://kaggle.com/competitions/icmcis-drone-tracking

[27] T. Rodrigues, J. Patrikar *et al.*, "Data collected with package delivery quadcopter drone," *Carnegie Mellon University*, pp. 1–15, 2020.

[28] A. Perrusquia, C. Tovar, A. Soria, and J. C. Martinez, "Robust controller for aircraft roll control system using data flight parameters," in *2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. IEEE, 2016, pp. 1–5.

[29] A. Perrusquía, R. Garrido, and W. Yu, "Stable robot manipulator parameter identification: A closed-loop input error approach," *Automatica*, vol. 141, p. 110294, 2022.

[30] J. A. Flores-Campos, A. Perrusquía, L. H. Hernández-Gómez, N. González, and A. Armenta-Molina, "Constant speed control of slider-crank mechanisms: A joint-task space hybrid control approach," *IEEE Access*, vol. 9, pp. 65 676–65 687, 2021.

[31] A. Maier, H. Köstler, M. Heisig, P. Krauss, and S. H. Yang, "Known operator learning and hybrid machine learning in medical imaging—a review of the past, the present, and the future," *Progress in Biomedical Engineering*, 2022.

Hybrid deep neural networks for drone high level intent classification using non-cooperative radar data

Fraser, Benjamin

2023-09-22

Fraser B, Perrusquía A, Panagiotakopoulous D, Guo W. (2023) Hybrid deep neural networks for drone high level intent classification using non-cooperative radar data. In: 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 19-21 July 2023, Santa Cruz de Tenerife, Spain

https://doi.org/10.1109/ICECCME57830.2023.10252859