Drone External and Internal Fault Detection Based on ROS topics; Final Report

Azin Shamshirgaran, Akshay Bhatia†

Abstract--- **Drones are effective for reducing human activity and interactions by performing tasks such as exploring and inspecting new environments, monitoring resources and delivering packages. During the current COVID-19 pandemic, and any similar outbreaks in the future, drones can be arranged and set up to help and improve the everyday lives of people by delivering packages and taking samples to hospitals. One important issue of utilizing drones is tackling fault of the system. In this project, we plan to build and optimize the performance of a network which will give feedback to the drone in the presence of external and internal fault to maintain a safe operation for it. We frame our problem as a binary classification task where the label ”0” would correspond to a ”not faulty” system and a label ”1” corresponds to a ”faulty” system and plan to apply deep learning methods to achieve this.**

Keywords: **Drone, Fault Detection, Sensory Data, Machine Learning, optimization.**

I. INTRODUCTION [1]

In the last decay, drone and quadcopters which has been considered in Unmanned Aerial Vehicle (UAV) family have drawn attraction to themselves due to their benefits. The main benefits of them are increasing robustness, reliability and stability, resource consumption saving and time saving.

The main application of drones are inspection and exploring new environment, resource management, computational biology, space discoveries, grid computing and manufacturing applications [2]. During the COVID-19 pandemic and similar outbreaks in the future, drones can be set up to improve the everyday lives of people. Drones are effective at reducing human interaction, which is crucial in times of pandemic. To reduce the risk of coronavirus infection, governments have asked and encouraged people to remain in their homes. But then, there should be a way to provide services and support for people in their homes. Drones can be used for that purpose by facilitating contact-free interactions with healthcare professionals, such as transporting blood or urine samples, and delivering medical supplies like medicine or healthcare devices. During a pandemic, hospitals are potential vectors of contamination, so drones provide an efficient contact-free way to transport critical and necessary medical supplies [3]. Although medical supply delivery has been achieved by the commercial company DJI [4], there are still many challenges, some of which we focus on in this project. To reach a goal and accomplish the task, drone needs to decide based on its states and status which can be considered like linear and angular position and velocity of the drone, its battery usage, heat and some other criteria.

Despite the progress in the area of control, navigation and target detection, there has been still challenges like fault detection during the process of accomplishing the task which is the focus of this paper.

A fault is an unexpected change in the system of the drone causing unacceptable deviation from the normal operation and behavior. In such event, the robot may become uncontrollable, can not accomplish the defined task and cause damage to its infrastructure as well as to humans [5].

II. RELATED WORK [1]

In this section we discuss some prior work for the Drone Fault Detection task and a brief overview of how our contribution is different from theirs.

Previous work in Drone Fault Detection task has utilized classical machine learning algorithms and statistical modelling techniques. For example, [6] uses GPS data with Generalized Linear Mixed Model (GLMM)to predict stress from GPS traces via the GStress model. In [7], the authors used sensory data and clustering method to identify activities of a flying drone like; moving forward, backward, flying down, up and etc. In [8], they used a classification method to predict the fault in drones. In [9] and [10], they just used IMU data to find the fault in the system.

Our work is different from these in two ways: 1) We apply a concept borrowed from Multi-task learning(MTL), known as ordering techniques, to enhance the performance of Convolutional Neural Networks(CNNs) that outperforms classical machine learning algorithms and 2) To train these models, we gather and label the sensory data using a Parrot Bebop 2 drone and also include some novel features for the same as discussed in the sections below.

III. DATASET EXPLORATION AND OVERVIEW

In this section, we describe some details about our dataset such the ratio of the faulty v/s not faulty labels, the correlation of various features with the true label(faulty) and the distribution of the features.

Our dataset consists of 10813 data points out of which 24.4%(2640) are faulty and 75.6%(8173) are not faulty as shown in Figure 2. In other words, the dataset is highly imbalanced i.e the number of not faulty samples is almost three times than the number of faulty samples. To show the robustness of our models, we will compute different metrics including Precision, Recall, F1 and AUC-ROC scores in addition to the standard accuracy scores. The updated version

*The authors are with the Department of Computer Science and Engineering, University of California, Merced, CA, USA. †Corresponding author e-mail address: ashamshirgaran@ucmerced.edu, abhatia8@ucmerced.edu

Fig. 1: Parrot Bebop 2 Drone used to collect the dataset)

of our dataset now has the following 18 features(Previously we just had 13 features):

Time interval
Normalized-X-position
Normalized-y-position
Normalized-z-position
Normalized-phi-angles
Normalized-theta-angles
Normalized-psai-angles
Normalized-x-velocity
Normalized-y-velocity
Normalized-z-velocity
Normalized-phi-velocity
Normalized-theta-velocity
Normalized-psai-velocity
Normalized-Temperature
Normalized-wind-speed
Normalized-Bent-propeller-degree
Normalized-Battery-Percentage
Normalized-Path-Percentage-traverse

Features in bold are the novel features we add to our dataset in the new version. Inclusion of these features in our proposed method improves the performance as discussed in the results section later.

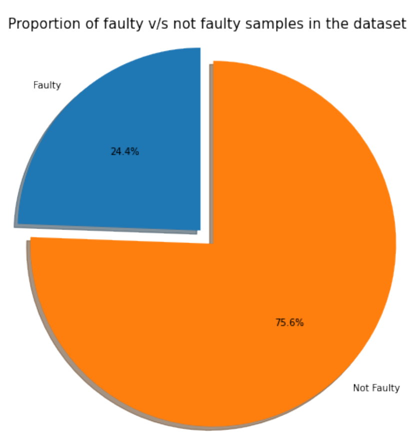

Fig. 2: A pie-chart showing the ratio of faulty v/s not faulty samples in our dataset

Below we discuss some statistics extracted from the dataset which may provide a much better understanding of our collected data and its distribution.

Figure 3 shows the correlation of each feature with the target label(faulty). Values greater than 0 show a positive correlation and values less than 0 show a negative correlation between the feature and the target label.

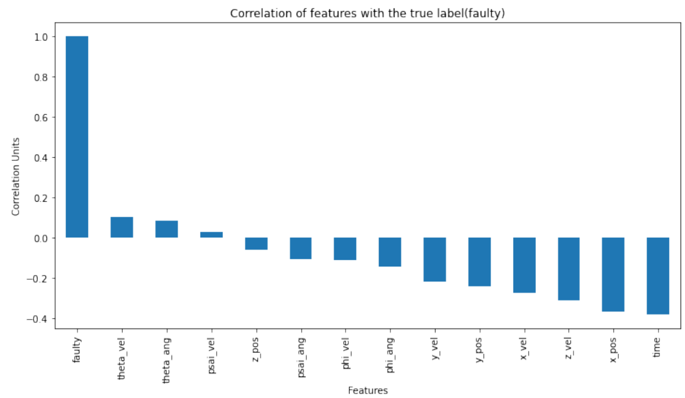

Fig. 3: Correlation of features with the target label. Here values $>$ 0 signifies +ve correlation and values $<$ 0 signifies -ve correlation

In many instances, it is also important to understand the distribution of the data we wish to model. This helps us interpret a machine learning models' prediction and also enables us to debug and validate its performance. Figure 4, explains the distribution of a subset of features. The top row shows the distribution of features with positive correlation and bottom row shows the distribution of features with negative correlation. We can observe a clear distinction between the distributions of each feature when categorized as faulty v/s not faulty samples.

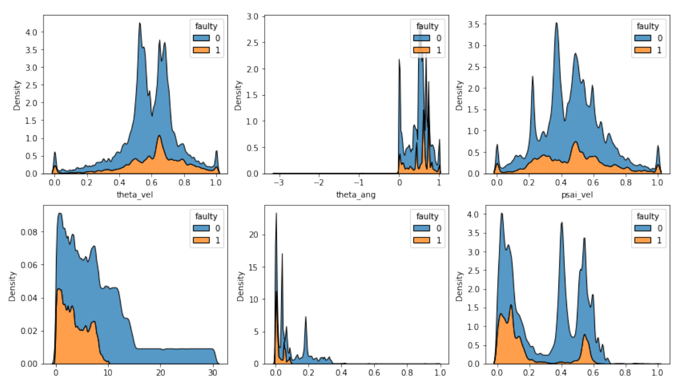

Fig. 4: Distribution of features faulty v/s not faulty: Top row is features with +ve correlation and bottom row are features with -ve correlation

IV. Technical Challenges with Convolutional Neural Networks

In our milestone report [11] we explored why deep learning techniques have be shown to outperform many classical machine learning algorithms and other statistical modeling approaches and the reasons why they are difficult to apply to the Drone Fault Detection task. We also looked at why it is difficult to collect data for the Drone Fault Detection task and how existing labeling tools and services are not the best solutions.

In this section we explore why vanilla Convolutional Neural Networks do not perform well for the Fault Detection task and our approach to fix it.

Convolutional Neural Networks differ from Feedforward Neural Networks in 2 ways: 1) They utilize kernels/filters which are only connected to a small local region in the previous feature representation rather than being connected to all of them. This lets them encode certain properties specific to a particular region in the input feature. 2) This also makes them space efficient and reduces the number of parameters in the networks thus making inference predictions faster compared to feedforward neural networks.

CNN transforms a N-dimensional tensor to another K-dimensional tensor where N and K depend on the filter/kernel, pooling and padding layer size(these are all hyperparameters of the model and can affect the model performance in different ways). These operations are called convolutions(thus the name CNN) that transform one set of features to another through a differentiable approximation function or non-linear activation functions. CNNs have enabled tremendous progress in many computer vision tasks such as object detection, image segmentation, image classification [12] etc.

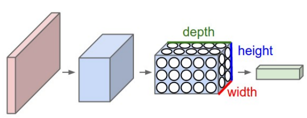

Fig. 5: Convolutional Neural Network

In spite of this progress, CNN do not perform well on tabular datasets i.e. 1-dimension data. This is due to the fact that CNNs expect the input features to be spatially correlated. In simpler terms, they can only extract features from the input data when the columns are contiguous and there is a local relation between the features. For example, consider an image of a dog. If we are trying to identify the nose of a dog, then the pixels adjacent to the body of the dog do not matter. If one were to replace the pixels near the nose with some arbitrary pixels from the anywhere in the image, we would not be able to identify the nose since the nose in the image would cease to exist. On the other hand, consider the set of features for the fault detection task which is a 1-Dimensional vector. If we were to switch the position of the velocity feature with the position feature, the data would still remain the same since the relative position of columns does not matter. In such a case there is no correlation between the feature set and hence CNN cannot extract any distinguishable features from the dataset.

V. Contribution and Results

We now present our proposed method which improves the performance of vanilla CNN for the faulty detection task. In general, we hypothesize that this method should work for any tabular dataset i.e 1-Dimensional dataset. Our approach is based on a concept known as Ordering, derived from the multi-task learning domain in deep learning. In the previous section we discussed that CNNs do not perform well on 1D data due to the absence of spatial locality. We fix this with a technique we introduce as ”Upsampled Ordering”. The basic idea of our solution is to reorder the set of input features in a way such that they transform as contiguous features. This is achieved by first upsampling(or increasing) the set of input features to a fixed integer through a fully-connected feedforward layer. We then reshape the features we obtained from the previous step to some fixed number of vectors where each vector consists of a contiguous set of K-dimension tensors. Repeating this process x times(where x is a hyperparameter of the network), the original set of 1-Dimensional data is transformed to a K-dimension contiguous tensor which can be furthered classified using a softmax layer for our binary classification task.

Below we present the results of our model and compare it with the baselines machine learning models based on accuracy, precision, recall, F1-score and most importantly, AUC-ROC score which explain model performance at various thresholds. We compare our proposed method Convolutional Neural Network with Upsampled Ordering(CNN-UO) with 4 classical machine learning algorithms - Naive Bayes(NB), Support Vector Machines(SVM), Logistic Regression(LR), Logistic Regression with Polynomial Kernel(LR-PK) and a vanilla Convolutional Neural Network(CNN).

TABLE I: Comparison of various methods with the proposed method

Model	Precision	Recall	F1 Score	Accuracy	AUC-ROC
NB	0.806	0.810	0.808	0.859	0.810
SVM	**0.946**	0.854	0.889	0.927	0.854
LR	0.916	0.905	0.911	0.936	0.905
LR-PK	0.918	0.906	0.912	0.937	0.906
CNN	0.926	0.914	0.919	0.927	0.914
CNN-UO	**0.946**	**0.943**	**0.944**	**0.959**	**0.943**

All models are trained with the full set of 18 features. The table above shows that our proposed method - CNN with upsampled ordering outperform all models including vanilla CNN mode. Figure 6 shows the Receiver Operating Curve(ROC) at various threshold values for all models.

VI. Conclusion and Future Work

In this report we discussed prior work for the Drone Fault Detection task, some details about our technical contributions including our proposed method and manually collected dataset and also why applying deep learning techniques to this task is still a challenge. We also explained various details about our proposed method and how it enhances a vanilla Convolutional Neural Network based on F1-scores and ROC graphs. In the future work we would like to frame the fault detection task as a time-series classification/prediction where, in addition to using the current sample as input to the proposed method,

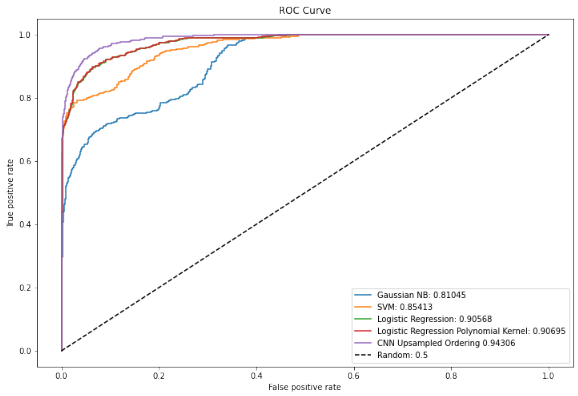

Fig. 6: Comparison of ROC curves of various models

we also incorporate previous information and apply recurrent neural networks to this task.

References

[1] A. Shamshirgaran and A. Bhatia, ‘‘Drone external and internal fault detection based on ros topics;theory, method and experimental result,’’ in Final project report 1, 2021.

[2] A. Shamshirgaran and F. Abdollahi, ‘‘Dynamic coverage control via underactuated autonomous underwater vehicles in unknown 3D environment,’’ in 4th International Conference on Control, Instrumentation, and Automation (ICCIA), pp. 333--338, 2016.

[3] A. Shamshirgaran, H. Javidi, and D. Simon, ‘‘Evolutionary algorithms for multi-objective optimization of drone controller parameters,’’ arXiv preprint arXiv:2105.08650, 2021.

[4] ‘‘Innovating to combat COVID-19 by DJI”,’’ 2017. Last accessed 21 September 2020.

[5] R. Isermann, Fault-diagnosis systems: an introduction from fault detection to fault tolerance. Springer Science & Business Media, 2006.

[6] S. Vhaduri, A. Ali, M. Sharmin, K. Hovsepian, and S. Kumar, ‘‘Estimating drivers’ stress from gps traces,’’ in Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, pp. 1--8, ACM, 2014.

[7] R. Barták and M. Vomlelová, ‘‘Using machine learning to identify activities of a flying drone from sensor readings,’’ in The Thirtieth International Flairs Conference, 2017.

[8] A. Manukyan, M. A. Olivares-Mendez, T. F. Bissyandé, H. Voos, and Y. Le Traon, ‘‘Uav degradation identification for pilot notification using machine learning techniques,’’ in 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1--8, IEEE, 2016.

[9] V. Sadhu, S. Zonouz, and D. Pompili, ‘‘On-board deep-learning-based unmanned aerial vehicle fault cause detection and identification,’’ in 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 5255--5261, IEEE, 2020.

[10] M. Bronz, E. Baskaya, D. Delahaye, and S. Puechmore, ‘‘Real-time fault detection on small fixed-wing uavs using machine learning,’’ in 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), pp. 1--10, IEEE, 2020.

[11] A. Shamshirgaran and A. Bhatia, ‘‘Drone external and internal fault detection based on ros topics; milestone report,’’ in Final project report 2, 2021.

[12] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ 2015.