A Multivariate time-series segmentation framework for Flight Condition Recognition

Francesco Zinnari[a], Giovanni Coral[b], Mara Tanelli[a,d], Gabriele Cazzulani[b], Andrea Baldi[c], Ugo Mariani[c], Daniele Mezzanzanica[c]

***Abstract*—Helicopters usage monitoring has gained significant attention in recent years, due to the safety and cost management implications. At its core there is the Flight Condition Recognition algorithm, which enables to detect the maneuvers carried out by the aircraft through on-board sensors measurements. In this work we propose a multi-variate time-series segmentation framework, which uses supervised learning algorithms, sliding windows and stacking ensembles to produce reliable estimates of the flown flight regimes. We validate the proposed approach on a large dataset of 460 labeled load flights from two distinct helicopter models, demonstrating its efficacy in predicting a range of 49 different maneuver types.**

***Index Terms*—Usage Monitoring, Flight Condition Recognition, Machine Learning, Time-series segmentation.**

I. Introduction

HELICOPTERS are extremely complex systems, with an high number of dynamic parts, especially when compared to fixed-wing aircrafts. Careful design of the critical components, regular maintenance and innovative methods to monitor the correct helicopters' operation have been employed in the recent years to guarantee the structural integrity of the rotorcraft and, thus, ensuring safety. Many different systems and processes have been developed to gather and analyse important information from the aircraft by using sensors, avionics equipment, and software. Notably, the first *Health and Usage Monitoring System* (HUMS) became widely available at the end of the 1980s, following the Boeing Vertol (BV) 234 North Sea accident and the consequently stricter regulations from major regulatory bodies such as the US Federal Aviation Administration (FAA) and United Kingdom Civil Aviation Authority (UK CAA) [1], [2]. These systems collect a variety of signals by means of on-board sensors, which are later processed through signal processing methods to extract relevant information for the helicopter's safety, availability and maintenance. In particular, HUMS are concerned with two main issues (as the name suggests): *Health Monitoring* and *Usage Monitoring*.

[a] The authors are with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, Via G. Ponzio 34/5, 20133, Milano MI, Italy. Email to: {francesco.zinnari,mara.tanelli}@polimi.it

[b] The authors are with the Department of Mechanical Engineering, Politecnico di Milano, Via Privata Giuseppe La Masa, 1, 20156, Milano MI, Italy. Email to: {giovanni.coral,gabriele.cazzulani}@polimi.it

[c] The author is with Leonardo Helicopter Division, Via Giovanni Agusta, 520, 21017 Samarate VA, Italy. Email to: andrea.baldi01@leonardocompany.com

[d] The author is with Istituto di Elettronica e Ingegneria dell'Informazione e delle Telecomunicazioni - IEIIT CNR, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

The former has historically been the main focus of most of the research and development, due to the immediate benefits of its implementation. Health monitoring is, in fact, concerned with the indirect measure of the state of damage/wear of a component by observation of its structural behaviour and response characteristics [3]. In particular, vibration monitoring has been proved efficient in early identification of incipient damage of rotating critical components, which might have otherwise degenerated and compromised the system functionality [4]–[7]. The operating assumption is that the vibration behaviour of compromised components deviates from the characteristic one under normal operating conditions. Comprehensive reviews of the methods for structural health monitoring through vibration patterns analysis are available in [8], [9]. Another notable health monitoring application is the detection of metal wear debris in the oil of engines and gearboxes through magnetic, electrical, optical or acoustic sensors [10]–[12].

On the other side, *Usage Monitoring* is concerned with understanding the operating conditions of the helicopter from the sensor recordings of the HUM system. The advantages of this surveillance task are, perhaps, not as immediate as their *Health Monitoring* counterpart and are more concerned with the long term effects in terms of both safety and cost management. Being able to understand which maneuvers were performed by an aircraft is, in fact, fundamental to understand the fatigue damage accumulated by different helicopter structures, as a result of mechanical stresses. Indeed, the fatigue content is strongly related with the type of maneuver being carried out [13]. An example of this is provided in [14], where it was demonstrated that for a Finnish Air Force F-18 aircraft, roll maneuvers are responsible for more than 50% of elevator spindle damage. Helicopter manufacturers first calculate the fatigue loads mathematically or from previous designs, but later use instrumented prototypes in Load Survey Flights to adjust the hypothesized loads according to the measured values [13]. A detailed example of damage fraction calculation based on S-N curves for different dynamic components is available in [15].

For all the stated reasons, aircraft manufacturers need to design the components according to an ideal usage spectrum, which attempts to capture the worst anticipated roles that the rotorcraft might undertake [16]. Component Retirement Times (CRTs) are evaluated on the basis of fatigue damages defined from components strength, flight loads and occurrences defined in the usage spectrum. The spectrum is defined in terms of percentage of time and/or the number of occurrences for

each flight condition. Such hypothesis are formulated based on the helicopter's primary role and the relative anticipated mission profiles. Furthermore, they can be also defined based on qualitative customer feedback from similar helicopter usage [16], [17]. The publication [18] offers an insight regarding how the design spectrum of the Fire Scout helicopter has been formulated, using standard Ground-Air-Ground cycles and pre-established sample mission profiles.

Helicopters, however, are very flexible rotorcrafts and their actual usage spectrum rarely corresponds to the one predicted by the manufacturer at design time, resulting in different levels of structural fatigue than the expected ones. Indeed, when the actual usage spectrum is substantially more damaging than expected, this may result in potential risk, as the components have not been certified for such high loads [19], [20]. In particular, according to [21], military attack aircrafts are known to experience substantial variability in their mission, and may have usage spectrum more severe than the assumed design spectrum. The author, therefore, argues that Individual Aircraft Tracking (IAT) and Condition Based Maintenance (CBM) programs are necessary, given the large variability between squadron operations, missions and pilots. Up to 24.8% of the aircraft system malfunctions listed in [22] could have potentially been avoided with a well-planned maintenance.

Given the safe-life philosophy behind the manufacturing and maintenance of helicopters' components, another common situation is that of having lower usage severity than the design usage [19], resulting in excessive underestimation of the CRT. Improved and customized maintenance plans can thus lower life-cycle costs: this is especially important for rotor-wing aircrafts, since maintenance costs are very high and represent about 24% of the direct operating costs of the helicopter [7]. Measurement of service usage of the Westland Lynx Mk7 under four types of service operation has demonstrated that service life might be extended up to an additional 50% [13]. Similarly, [7] reports a study where, thanks to an appropriate use of HUM systems, there was a 5-10% decrease in scheduled maintenance and a 30% reduction in the interruption of missions.

Due to the strategic value and interest of Usage Monitoring, the FAA has assigned a high priority to the development of efficient and precise regime recognition algorithms [23]. However, as noted, not many works have explored the issue in depth, especially if compared to the extensive research carried out on other aspects of HUMS like vibration monitoring. In the next section we will provide an overview of the relevant research studies in the field of Flight Condition Recognition (FCR).

II. State of the art

After the widespread adoption of HUMS, the first methods for FCR were developed. A common approach in avionics is to use logic-based systems based on rules, which are established by industry experts according to mathematical models. The method developed in [24] for the Boeing MH-47 Chinook helicopter achieves a 90% accuracy. Similarly, the algorithm developed and perfected over the course of 8 years for the Boeing AH-64 Apache attack helicopter, was able to correctly identify 98% of the regimes from a test set of 1127 maneuver executions (multiclass classification framework with a total of 31 possible conditions). Although these might be considered satisfactory performance metrics, manual threshold-based systems require intensive and continuous refinement from domain experts over the course of many years, with rules tailored to the specific helicopter model. Such algorithms are not flexible nor scalable: a change in parameters, maneuver categories or helicopter model might necessitate the full reworking of the logical tests. Another critical aspect of this approach is mentioned in [25], which highlights the difficulty of defining the logical tests such that no overlap exists between the different maneuver categories. This would hold especially true when scaling to hundreds of maneuvers.

Algorithms based on automatic statistical inference are generally able to overcome the limitations of manual rule-based systems, due to their flexibility and learning ability. In fact, machine learning approaches have the advantage of easily adapting to different configurations, making these solutions the most versatile and adaptable to different fleets. However, the applications of machine learning to the field of FCR are few and limited. In [26], the authors use a hierarchical maneuver recognition scheme based on 10 neural networks, each recognizing dealing with a regime group. The approach achieves a 76.21% accuracy, with mixed results for different regime groups (98% accuracy for levelled flights regimes and 33% for turns). In [27], the author uses supervised machine learning algorithms (such as Feedforward Neural Networks, Decision Trees and Nearest Neighbors Classifiers) to evaluate whether it would be possible to perform regime recognition from a single strain gauge measurement. The algorithms achieve 94% accuracy on a limited scenario of five regimes and 10 test flights (each with 14 examples). Indeed, the author suggests that the results are not conclusive enough to verify that the learning is robust and reliable. In [28], instead, the authors use a logistic regression model to classify maneuvers according to their intensity and severity only using on-board MEMS IMUs data. The output can be both discrete (high intensity, low intensity, or no maneuver) or continuous (reflecting the intensity of the maneuver). However, the work is not concerned explicitly with regime recognition, but rather with identifying the general severity of maneuvers. In [29], the authors train different statistical models on data acquired from nonlinear rotorcraft simulators, in order to recognize a set of 57 regimes. Both supervised (Neural Networks, Logistic Regression) and unsupervised (Gaussian Mixture Models - GMMs) approaches were explored. Surprisingly, the unsupervised method was able to obtain the highest accuracy score of 91.47%. However, the size of the test set is unspecified, as well as its composition: due to the likely class imbalance, accuracy might not accurately capture the complete algorithm's performance. Furthermore, the described method treats every sample of the multivariate time-series as independent, losing the temporal relation between adjacent samples. The authors later introduce the temporal relationship between successive samples in [30],

using Markovian transition probabilities. Results show an approximate accuracy of 92% on a 30 minutes piloted flight simulator data, with a substantial reduction of rapid transitions between predicted regimes. However, in this case the test set is very limited: the study does not mention which classes are included in this test set, but we can assume that only few instances of each maneuver could be executed in such a short time window. Again, accuracy might not be sufficient for evaluating the performance of the algorithm on the full regime spectrum.

A final group of solutions adopts a different approach, including an internal model of vehicle dynamics to produce meaningful predictions. In [31], a set of 5 interacting multiple model estimators (each containing a bank of extended Kalman filters) are used to estimate the dynamic mode of the system, which are assumed to correspond to the regimes. The method achieves a 84.1% normalized accuracy on a simulated 7 minutes flight. The authors also provide empirical examples of the threshold-based methods limits when working on noisy experimental data. In [32], the estimation of flight conditions from measured flight parameters is performed through Kalman filters, which output a performance index for each identified regime. The efficacy of the proposed method is only validated through visual inspection of sample flights.

III. Contributions

This work aims to fill the literature gap by proposing an effective and versatile supervised machine learning approach for multivariate time-series segmentation, applied to Flight Condition Recognition. The main contribution of the paper can be summarized as follows:

- We analysed, processed and cleansed a massive dataset of partially labeled load survey flights, containing 11078 maneuvers from 490 flights.
- We designed a process for multivariate time-series segmentation based on stacking ensemble learning on multiple sliding windows.
- We demonstrated the potential of machine learning for FCR by validating the algorithm on a stratified test set, obtaining a 96% accuracy and a 0.94 F1-score (macro averaged) on a multiclass classification problem with 49 classes.

To the best of the author's knowledge, this is the first time that such a large and comprehensive dataset of Load Survey Flights is used for training and validating a FCR model. The proposed approach is able to outperform the previous automatic classification methods reviewed in the literature. Furthermore, we believe the model to be very robust and versatile, with potential application in many other time-series segmentation tasks, since with suitable features and window sizes it can deal with both stationary and non-stationary signals.

IV. Paper Structure

The paper is organized as follows. Section V describes the structure of the dataset acquired from the on-board HUM system. Section VI explains how the dataset was preprocessed and cleaned from mislabeled examples. Section VII formulates an optimization problem for the stratified train-test split on a per-flight basis. Section VIII illustrates the process used for the time-series segmentation and its components. In particular, Section VIII-A describes the statistical features extracted to classify the maneuvers. Section VIII-B specifies how the first-level model was trained and validated. Subsequently, Section VIII-C introduces the sliding-windows on the full flights and explains the role of the second-level model (also called supervisor). Finally, Section IX presents the algorithm's performance on the test set, using various metrics.

V. Dataset Description and Exploration

A. Dataset Description

In order to develop an FCR model based on Machine Learning techniques a great amount of data has to be collected and processed. Load Survey Flights are possibly the only available sources of real labeled flight data, collected from the on-board HUMS. As mentioned in Section I, load survey campaigns are carried out to evaluate load distributions and fatigue experienced in different flight conditions and helicopter configurations.

Our work is based on two extensive load survey campaigns and their relative datasets, performed for two different Leonardo Helicopter models. Such helicopters are used for very different applications: from passenger transport to Search And Rescue (SAR) missions.

Given the nature of the load survey activity, the flights were carried out from professional pilots, who were tasked with the execution of a list of standard flight conditions. The pilots signal the beginning and end of each maneuver through an on-board interface. Only portions of the flights are labeled, while the rest of the flight is unlabeled (we don't know a-priori which maneuvers are being performed). Hence, every flight corresponds to a multivariate time-series, each having some maneuvers that have been labeled with the corresponding standard flight condition (as defined in the design spectrum) and have been assigned a unique code, which uniquely identifies them from their start to their end. A schematics of the flight structure is presented in Figure 1. The set of signals that make up the flight's multivariate time-series are here symbolically identified as V0,,VN. The flight is partially labeled with standard flight conditions belonging to the design spectrum (sections L0 and L1 with the respective labels MCi and MCj), while the remaining part of the flight is unlabeled (sections NL0,NL1 and NL2).

The dataset is composed of 490 flights, containing a total of 11078 maneuver instances. In particular, 263 flights were carried out by two equipped prototypes of the "Model 1" , while 227 others were flown using two "Model 2" helicopters. A more detailed quantitative description of the dataset is available in Table I.

The HUM system records a wide set of signals through on-board sensors scattered across the rotorcraft. A subset of 30 signals sampled at $12.5\,\mathrm{Hz}$ is available as the input our FCR problems. The recorded signals include the rotation

angles of the vehicle frame around the aircraft principal axes (for visual reference, see Figure 2), their rate of change, the speed and acceleration of the vehicle along these axes, a set of parameters to monitor the engines operating conditions, altitude information and more.

B. Definition of the Macroclasses

The initial dataset was very unbalanced, and contained a very large number of labels: the maneuvers were, in fact, divided in more than 340 standard regimes (or Standards, for short). While some of these Standards were massively populated (as in the case of the various level flights), many others only have only been recorded few times (like for taxiing maneuvers). However, many of these Standards are alteration of the same *Macro-regimes* (or *Macro-classes*), and their definition differs solely on the basis of a single parameter's variation (as it is the case for the Roll Angle of Bankturns). Thus, we decided to group the Standards into logical clusters (*Macro-regimes* or *Macro-classes*), and later used these new labels as the new target of our segmentation task. The definition of these *Macro-classes* was undertaken with the help and supervision of domain experts. This approach had several advantages:

- **Reduced number of targets and increased number of elements in each class**: because the grouped maneuvers share distinctive commonalities, while having - at the same time - some variation, the learning algorithm will have an easier time focusing on the important features, avoiding overfitting and resulting in improved generalization.
- **Compensation for inaccurate maneuvers executions**: maneuvers labeled under the same *standard* contain a high variability, due to the human factor. As such, it may happen that a maneuver execution does not exactly correspond to the assigned label, but rather to a similar variation of the same flight regime (e.g. forward flights with slightly different speeds). These imprecise executions would pollute our dataset with erroneous labels. We can, thus, compensate for this inaccuracy by assigning the variations to the same *Macro-class*

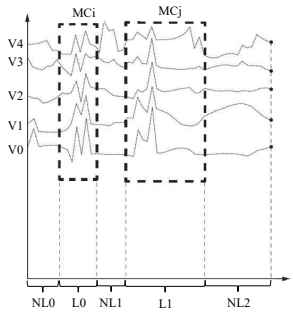


Fig. 1: Structure of a load survey flight

Fig. 2: Helicopter rotational axis

The list of 49 *Macro-classes* can be inspected in Figure 3, together with their numerosity. Despite the aggregation in larger groups, the dataset is still strongly unbalanced, a problem that must be taken into consideration when training the model (see Section VIII).

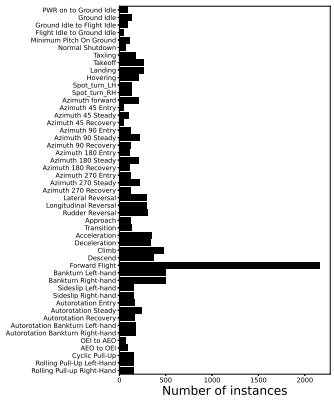


Fig. 3: Number of instances per Macro-class

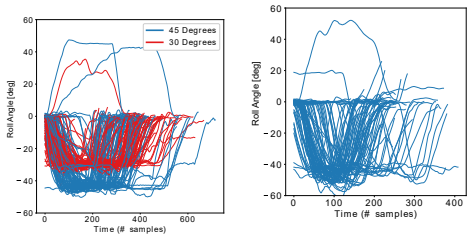
C. Exploratory Data Analysis

Before proceeding with any kind of alteration to the dataset, we performed an Exploratory Data Analysis (EDA) to get a better understanding of the signals' properties and inspected their value distributions. We started exploring the maneuvers belonging to the same standard category by visualizing their typical patterns. As an example, we report the *Roll* parameter variation for the *Bankturn Left-hand* macro-class in Figure 4. It is clear that the value of this parameter is one of the defining characteristic of this type of maneuver, and something we aim to capture when extracting our features. At the same time, 4b illustrates how the roll attitude has also a similar trend for the *Rolling Pull-Up Left-hand* maneuver, illustrating the need to simultaneously consider multiple parameters at the same time to correctly classify the executed maneuver. Figure 4a clearly shows the advantages of grouping maneuvers into macro-classes: macro-classes are more numerous and varied, making it easier for the model to learn the actual distinguishing feature of a certain label and improving the generalization of the model in unforeseen conditions. These plots also expose one of the main problems that we had to tackle when dealing with our data: mislabeled maneuver instances. In these examples, some of the maneuvers erroneously labeled as Left-handed were actually performed by turning on the Right-hand side. The method for dealing with this problem is described in Section VI

Additionally, we inspected the duration of the maneuvers with respect to their *Macro-classes* (Figure 5). It's clearly visible that different regimes correspond to different durations:

TABLE I: Dataset overview (before cleaning)

Helicopter Model	Number of flights	Number of maneuvers	labeled hours of flight
Model 1	263	5768	45 h
Model 2	227	5310	37 h 42 m
TOTAL	490	11078	82 h 42 m



(a) Bankturn Left-hand (b) Rolling Pull-Up Left-hand

Fig. 4: Roll Angle for two different macro classes

this is a challenging characteristic of our dataset, and is the basis of the approach developed in Section VIII.

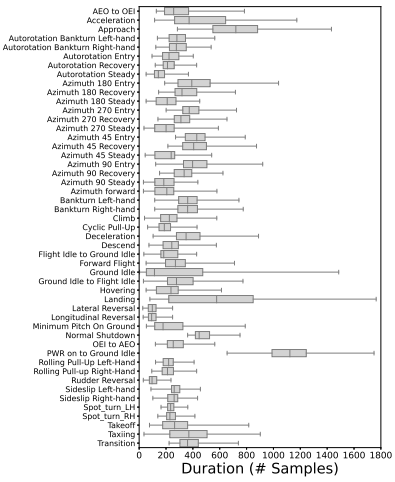


Fig. 5: Duration boxplot

VI. Data Preprocessing and Cleansing

The data has been thoroughly cleansed from any error, through a multi-step processing. Since the information has been sampled directly from equipped prototypes, we can expect the dataset to contain a number of acquisition errors. For this reason, we initially inspected the parameters' distributions of each flight to detect anomalous time-series. This way, we were able to remove all those flights containing flat, noisy or missing signals. As a second step, we dealt with a very important issue in machine learning problems: mislabelling. We have already mentioned how the pilots are instructed to perform certain sequences of maneuvers during the Load Survey Flights, and how the Standard labels are manually recorded based on such instructions. This is an error-prone procedure, as pilots sometimes, due to unexpected events (gust or other external events), perform corrections that lead to carry out maneuvers slightly different respect to the ones required, or the flight test engineers do not correctly signal the beginning and ending of each maneuver. However, visually or manually examining each instance, to make sure that it has been correctly tagged, is inconvenient and unfeasible.

For this reason, similarly to the strategies explained in [33], we carried out a filtering by Cross-Validation. Using the model that will be described in Section VIII-B (with the feature summarized in Section VIII-A), we performed a 10-fold Cross-Validation to detect possible mislabeling. After acquiring the model's output on the different folds, we concentrated on those maneuver instances for which the prediction differed from the target label. By manually/visually inspecting these maneuvers we verified whether the label was actually mistakenly assigned or not, and acted accordingly. The process was facilitated by the use of Model Predictions Interpreters such as SHAP [34] and LIME [35], which provided effective insights on which feature impacted the model prediction the most, thus suggesting us which variables to analyze during the inspection.
Once we confirmed the labeling mistake, we attempted to replace the erroneous label with the correct one when this was possible, and removed the label (thus invalidating the maneuver instance) when this was not possible.

Additional data cleaning was performed manually to detect anomalous patterns using domain knowledge. In particular, given the huge number of maneuvers in the *Forward Flight* class, we dedicated particular attention to making sure that this class did not contain any impurities, which might have otherwise affected the correct detection of other classes. We performed manual checks for detecting accelerations, decelerations, climbs or descents within such maneuvers and invalidated these instances.

The final result of this multi-step processing is visible in Table II. The dataset has been reduced of 7.4% (in terms of maneuver instances) from the original . However, with higher data quality, the model will be able to learn meaningful patterns and increase its prediction confidence.

VII. Stratified Flights Train-Test Split

When training supervised learning algorithms, splitting the dataset into Train and Test is fundamental to evaluate the model performance on unforeseen instances.
We have already mentioned that the dataset is strongly unbalanced (see Figure 3). In this case, it is especially

TABLE II: Dataset overview (after cleaning)

Helicopter Model	Number of flights	Number of maneuvers	labeled hours of flight
Model 1	237	5115	34 h 43 m
Model 2	223	5147	32 h 43 m
TOTAL	460	10262	67 h 26 m

important to perform a stratified partitioning of Train Set and Test Set, as we want to preserve the same proportions of maneuver classes as those observed in the entire dataset.

For our purposes (see Section VIII), we want to perform a Train-Test split on a per-flight basis, meaning that a full load survey flight - with all of its maneuvers - will either completely belong to the Train Set or completely belong to the Test Set. At the same time we want to maintain the class proportion. To solve this issue we formulated the problem as an Integer Linear Program (ILP), with the objective of obtaining a 70%-30% split while also respecting the stratification of the dataset. The ILP formulation is as follows:

minimize:

$$\sum_{j}^{C} \frac{|\sum_{i}^{F}(x[i] - 2.5 \times (1 - x[i])) \times MPF[i][j]|}{\sum_{i}^{F} MPF[i][j]} \tag{1}$$

subject to:

$$x[i] \in \{0, 1\} \qquad i \in F \tag{2}$$

$$MPF[i][j] \in \mathbb{Z}^{+} \qquad i \in F \qquad j \in C \tag{3}$$

where F is the set of all flights, C is the set of all Macro-classes. The variable x is binary, and it has the following meaning:

$$x[i] = \begin{cases} 1 & \text{Flight } i \text{ belongs to Train Set} \\ 0 & \text{Flight } i \text{ belongs to Test Set} \end{cases} \tag{4}$$

Finally, the variable MPF (short for Maneuvers Per Flight) contains the number of maneuver belonging to a certain class j within a flight i. The non-linear objective function was then linearized through the use of additional constraints and slack variables.

The results of the optimized split can be inspected in Table III. The average difference with respect to the class proportions of the entire dataset is included in the last column, to better appreciate the outcome of the stratification.

VIII. Time series segmentation for flight condition recognition

The FCR task consists in segmenting a group of signals (that is, a multivariate time-series) into smaller portions, based on the type of maneuver being performed. To achieve this goal, we will use pattern recognition algorithms, which are able to learn the optimal decision boundaries from the observed data: these are generally referred to as discriminative models.

We can, for example, teach our model to distinguish between the maneuvers contained in our dataset, by extracting a set of fundamental features that summarize the statistical properties of each macro-class.
However, for real-life scenarios in which the boundaries between adjacent regimes are unknown a-priori, the problem of determining how to partition the full time-series arises. A well-known solution to similar problems (used especially in signal processing) is that of creating a suitably sized sliding window to slide across the time-series, extracting the features from each window and then producing a classification output accordingly. The issue emerges when we deal with time-series whose constituent segments have very different lengths: a short window might be suitable for recognizing quick bursts or sudden changes, but is unable to capture the defining features of longer segments, especially when these have entry and exit conditions that differentiate them from other classes. Vice versa, longer windows cannot accurately capture quick variations and thus are not suitable for classes with shorter length.
Ideally, the window size should be as similar as possible to the actual duration of the maneuver. However, different maneuvers have different duration. For this reason we propose a method that uses different window sizes to produce a set of First-level predictions, which are later aggregated through a Second-level classifier (which we also call *Supervisor*) into a final probability distribution. The method shows improvement over the predictions produced using only a single sized window, confirming the efficacy of the proposed approach (see the Section IX for comprehensive results). An overall view of the method is provided in Figure 6, and detailed description of the process and of the individual blocks is presented in Section VIII-A,VIII-B and VIII-C.

A. Feature Extraction

As we have mentioned, the input of discriminative models is, in general, a set of features. The model will learn automatically to map these inputs to the corresponding target class. In our case, statistical features are used to capture the characteristic of a set of contiguous samples from the flight. The idea behind this method is to create a representation of a portion of time series as an array of statistical features (like mean value, standard deviation, etc.). The assumption is that, by selecting a set of comprehensive features through specific domain knowledge, we can condense the fundamental characteristics which define and distinguish such objects from others into a vector of features.

In this work, a set of general (yet meaningful) statistical properties is extracted from portions of the flights, to capture

TABLE III: Train Set and Test Set Split

Helicopter Model	Number of flights	Number of maneuvers	labeled hours of flight	Avg variation of class proportions
Train Set	310	7153	47 h 12 m	3%
Test Set	150	3109	20 h 13 m	6%

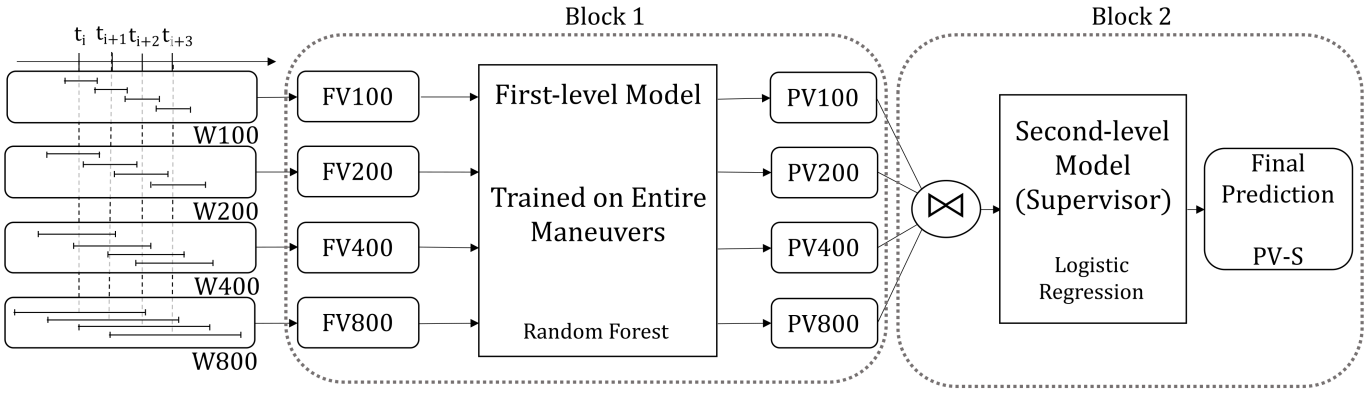


Fig. 6: Overall model

the behaviour of the helicopter within the analysed time-frame. Moreover, through a process of feature engineering (signal processing), we derived additional and important signals that were missing from the original dataset, which will also be included in the calculation of the statistical features. These signals consist in the approximation of the derivative for those signals that did not have a matching rate of change column (for example the control positions and the engine variables). The general feature engineering and feature extraction process is summarized in Figure 7

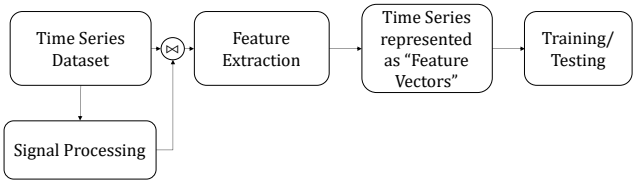


Fig. 7: Feature Engineering and Feature Extraction Process

TABLE IV: List of extracted features

All signals (original signals and engineered signals)	Maximum Minimum Mean Standard Deviation

The statistical features that were extracted over the set of both the original and the engineered signals, can be consulted in Table IV. Additionally, we discarded some of the newly obtained features, since we do not want them to influence the final classification outcome:

- *Heading*:[Maximum,Mean,Minimum] discarded because it represents the direction of the nose of the helicopter with respect to the geographical North. We don't want maneuvers to be classified based on such information, as every maneuver can be obviously carried out in every direction.
- *Barometric Altitude*: [Maximum,Mean,Minimum] discarded because dependent on the environment, and often has invalid values below 0. The more precise (at least at low altitude) *Radar Altitude* will provide the information required to distinguish between hovering regimes and higher altitude maneuvers.

The final feature vector, containing all the summarized statistical properties of the consider time-series segment, is composed of 170 features.

In the next sections, we will provide further details on how this feature extraction process has been applied within our approach.

B. Block 1: First-level model on entire labeled maneuvers

The first block of our method uses a First-level model (trained on the entire maneuvers) to produce a set of predictions, using sliding windows of different sizes over the entire flights.

We start by concentrating and detailing the First-level model and the process used for its training. As we mentioned, our dataset is the result of a Load Survey Flights: as such, it contains labeled maneuvers within the flights, and the relative information of start and end time. We can extract these portions and create a dataset which contains all (and only) the labeled maneuvers - let us call this dataset as the Individual

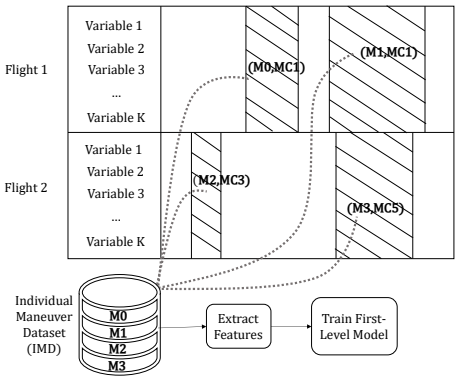


Fig. 8: Individual Maneuvers Dataset and First-level model training

Maneuvers Dataset (IMD), Figure 8. After this, we can apply the feature extraction procedure (detailed previously) to collapse each maneuver instance in the IMD into a feature vector (considering the maneuvers across their entire duration). The outcome of this process is a dataset containing a maneuver in each row (uniquely identified through an ID), the set of calculated features as columns and a vector of Macro-class labels as the targeted outcome of the classification. This tabular form, corresponds exactly to the type of input expected by a discriminative model. In this case, after a model selection with a 10-Fold Cross-Validation on the Training Set, a Random Forest model with 300 decision trees was chosen, since it achieved the best performance.
To cope with the high class imbalance, the model uses class weights to penalize more the classification errors on the minority classes.

This trained model is now ready to move to the more challenging scenario of full flight segmentation using sliding windows. For our application, we defined four window sizes based on the 5th, 25th, 75th, 95th percentiles of the maneuvers duration:

- Window *W100* of size 100 samples (corresponding to 8 s) and corresponding Feature Vector *FV100*.
- Window *W200* of size 200 samples (corresponding to 16 s) and corresponding Feature Vector *FV200*.
- Window *W400* of size 400 samples (corresponding to 32 s) and corresponding Feature Vector *FV400*.
- Window *W800* of size 800 samples (corresponding to 64 s) and corresponding Feature Vector *FV800*.

The windows slide across the entire flight, they are centered around the same timesteps (indicated as $t_0, t_1, t_2, ..., t_n$) and, thus, they are superimposed. In our case, the windows have a stride between timesteps of 25 samples (corresponding to 2 s), enforced to avoid excessive computational load; however, this is not a prerequisite of the proposed model, and the stride can also be of 1 sample.
The feature vectors are later processed from the First-level model, which produces a set of predictions, in the form of probability distributions over all the Macro-classes. In summary, the output of the first block are prediction vectors PV for each of the window sizes $W \in \{W100, W200, W400, W800\}$, centered around each timestep $t \in \{t_0, t_1, t_2, ..., t_n\}$. An example of the probability vectors PV100 is provided in Figure 9.
We argue (and demonstrate later in Section IX) that the individual predictions *PV100,...,PV800* can be improved through a Second-level model (or *Supervisor*) that aggregates them in a smart way, cleverly weighting the predictions from different window sizes and for different classes.

C. Block 2: Second-level model on sliding windows

The second step can be considered a meta-learning step, since it takes as input the probability distributions computed from the upstream model and produces a final, refined classification output. This method of probability aggregation (or stacking), which leverages pattern recognition techniques, is expected to outperform simpler aggregation strategies, such as Soft-voting, Hard-voting and Maximum Probability. This is because the Supervisor model can exploit the power of data mining to automatically understand which prediction from the First-level model is particularly significant. As an example, if we consider the case of the *Lateral Reversal* Macro-class (a type of quick and sudden maneuver), the model will learn to attach greater importance to the probability output produced from the short window *W100*, compared to the one produced from longer windows such as *W800*.

TABLE V: Dataset after concatenating First-level predictions

Flight	Time	PV100	PV200	PV400	PV800	Target
F1	t_i					MC1
	t_{i+1}					MC1
	t_{i+2}					MC1
	t_{i+3}					MC1
	t_{i+4}					MC1
...	...	...	...	...	...	...

The learning dataset for the Block 2 is described in Table V. During the training phase, the algorithm learns the mapping between the prediction vectors *PV100,...,PV800* (obtained by the sliding windows across all the Load Survey Flights and later joined into a single table using window centers t as index) and the target label (also taken at t). Obviously, when building this new dataset, we will only consider those timesteps t for which a target label is available, and ignore all the other windows. Similarly to the first-level model, we use class weights to compensate for the class imbalance. The model that obtained the better compromise between training complexity and validation performances was a simple Logistic Regression model with L2 regularization. For additional details on the *Supervisor* model selection, see Section IX. The output of this Second-level (called *PV-S*) is a probability distribution over all the Macro-classes and represents the final prediction of the model.

D. Overall model

The proposed model, composed of Block 1 and Block 2, is now ready to be utilized for segmenting entire flights.

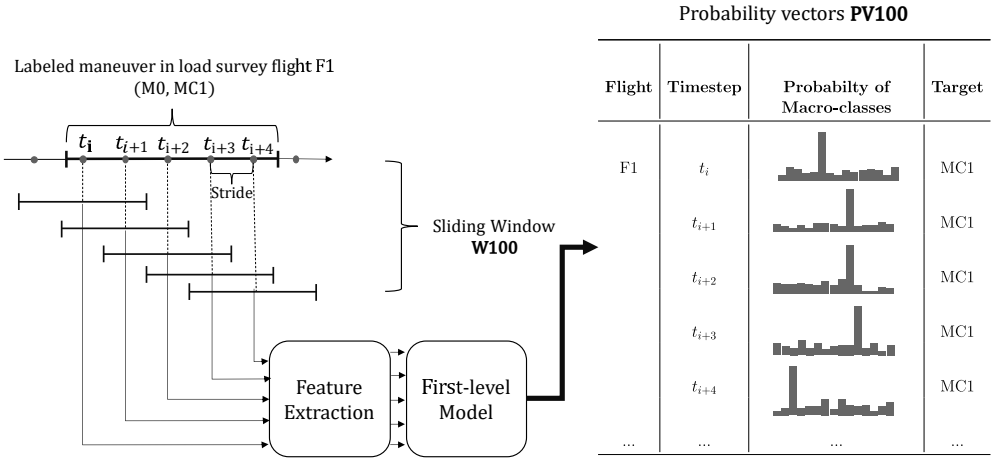


Fig. 9: From sliding windows to probability vectors

The sliding windows, each centered at the timesteps $t \in \{t_0, t_1, t_2, ..., t_n\}$, sweep across the entire flight, converting the raw signals through feature extraction. These are then ingested from the First-level model producing the first-level *PV100,...,PV800* probability vectors. Finally, these are aggregated from the *Supervisor* model, which produces the final prediction *PV-S* (Figure 6).

IX. Results

We now present the results of the overall framework on the test flights. As we explained, the First-level model produces predictions for each of the four different window sizes. We want to demonstrate that our stacking method is able to produce more accurate predictions than the individual ones yielded from the individual windows, as well as better predictions than those obtained using simpler aggregation methods. The additional aggregation methods for the First-level probabilities that we evaluated are:

- **Soft-Voting**: compute the average of the probability distributions from the first-level, the one with the highest probability is the predicted one.
- **Hard-Voting**: take the class with the highest probability for each window. Count the number of occurrences of each regime: the one with the greatest number is the predicted class. In case of tie, choose the class with the highest probability score.
- **Maximum Probability**: the prediction is the class with the highest probability score, out of all the predictions from all the windows.

Classic metrics such as the micro-averaged accuracy are not suitable for our problem, due to class-imbalance. We instead present the results as macro-averaged performance metrics such as accuracy, precision, recall and F1-Score. A Macro-average computes the metric independently for each class and then takes the average, thus treating all the classes equally (independently from their support).
The illustrated results follow the approach presented in Section VIII, applied to the Test Set using a Logistic Regression model. As such, the predictions refer to the center points of our sliding windows (with a 2 s stride interval). In total, we evaluated 35860 predictions from 150 test flights. The results can be consulted in Table VI. Our method, which combines the prediction of the First-level model using a Supervisor (implemented as a Logistic Regression classifier) outperforms all the other approaches on every considered performance metric. This results also represent an improvement over the other existing automatic regime recognition methods analysed in the literature.

TABLE VI: Performance metrics with different methods

	Strategy	**Acc.**	**Prec.**	**Recall**	**F-1**
Single-window	WS=100	72%	0.81	0.64	0.63
	WS=200	83%	0.85	0.78	0.78
	WS=400	87%	0.85	0.83	0.83
	WS=800	73%	0.72	0.65	0.63
Multi-window	Soft-voting	90%	0.9	0.86	0.87
	Hard-voting	89%	0.89	0.85	0.85
	Max prob.	89%	0.89	0.86	0.86
	Supervisor	**96%**	**0.94**	**0.95**	**0.94**

We also provide the full classification report, to inspect the efficiency of our algorithm in recognizing all the different Macro-classes (in Table VII).

As already mentioned, in the model selection phase we evaluated both a linear classifier (Logistic Regression with L2 regularization and C=10) and a nonlinear classifier (Feed-forward Neural Network with two layers, 1000 neurons

TABLE VII: Classification Report

	Prec.	Recall	F1
PWR on to Ground Idle	0.99	0.99	0.99
Ground Idle	0.91	0.97	0.94
Ground Idle to Flight Idle	0.82	0.90	0.86
Flight Idle to Ground Idle	0.74	0.59	0.66
Minimum Pitch On Ground	0.79	0.86	0.82
Normal Shutdown	0.98	0.90	0.94
Taxiing	0.98	0.97	0.97
Takeoff	0.98	0.96	0.97
Landing	0.95	1.00	0.97
Hovering	0.91	0.94	0.93
Spot_turn_LH	0.96	1.00	0.98
Spot_turn_RH	0.96	0.99	0.97
Azimuth forward	0.92	0.97	0.95
Azimuth 45 Entry	0.96	0.88	0.92
Azimuth 45 Steady	0.88	0.95	0.91
Azimuth 45 Recovery	0.85	0.95	0.90
Azimuth 90 Entry	0.96	0.95	0.96
Azimuth 90 Steady	0.92	0.97	0.94
Azimuth 90 Recovery	0.98	0.95	0.96
Azimuth 180 Entry	0.97	0.94	0.96
Azimuth 180 Steady	0.92	0.96	0.94
Azimuth 180 Recovery	0.96	0.94	0.95
Azimuth 270 Entry	0.99	0.98	0.98
Azimuth 270 Steady	0.97	0.98	0.97
Azimuth 270 Recovery	0.98	0.98	0.98
Lateral Reversal	0.97	0.94	0.95
Longitudinal Reversal	0.94	0.93	0.94
Rudder Reversal	0.97	0.98	0.98
Approach	0.95	0.95	0.95
Transition	1.00	0.95	0.97
Acceleration	0.95	0.90	0.92
Deceleration	0.94	0.92	0.93
Climb	1.00	0.99	0.99
Descend	0.97	0.97	0.97
Forward Flight	0.97	0.98	0.97
Bankturn Left-hand	0.98	0.98	0.98
Bankturn Right-hand	0.98	0.96	0.97
Sideslip Left-hand	0.97	0.96	0.97
Sideslip Right-hand	0.96	0.99	0.97
Autorotation Entry	0.94	0.93	0.94
Autorotation Steady	0.93	0.94	0.93
Autorotation Recovery	0.96	0.97	0.97
Autorotation Bankturn Left-hand	0.99	0.98	0.99
Autorotation Bankturn Right-hand	1.00	0.99	1.00
OEI to AEO	0.95	0.92	0.93
AEO to OEI	0.97	0.96	0.96
Cyclic Pull-Up	0.88	0.97	0.92
Rolling Pull-Up Left-Hand	0.94	0.93	0.94
Rolling Pull-up Right-Hand	0.88	0.95	0.91

and ReLu activation functions). The performance of the two models were evaluated using a 10-fold Cross-Validation with stratification on the Training Set. The results can be visualized in Table VIII. The neural networks has far more parameters than the logistic regression model, and thus its training requires longer computational time. At the same time, there are no clear advantages in using a complex classifier, so the Logistic Regression was preferred.

Finally, we show an example of a comparison between the labeled ground-truth and the predicted usage, over a randomly selected test flight, as reported in Figure 10 and Figure 11,

TABLE VIII: 10-Fold Cross-Validation for Model Selection of Block 2

Model	**Acc.**	**Prec.**	**Recall**	**F-1**
Logistic Regression	96%	0.94	0.95	0.94
Neural Network	96%	0.94	0.94	0.94

respectively. The x-axis provides a timescale, and the y-axis indicate the macro-classes. For simplicity, we only show the labeled sections of the flight, as the unlabeled ones would not be useful for this comparison. It's clearly visible how the algorithm is able to track a wide range of different maneuvers, including ground operations, hovering conditions, turns and transitions.

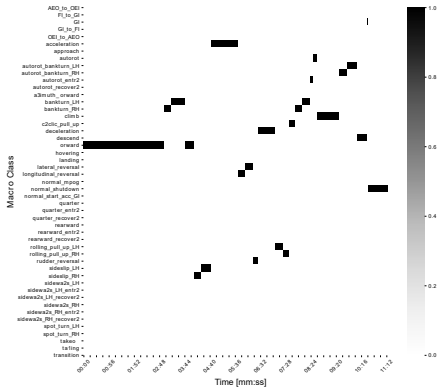


Fig. 10: labeled maneuvers in a test load survey flight

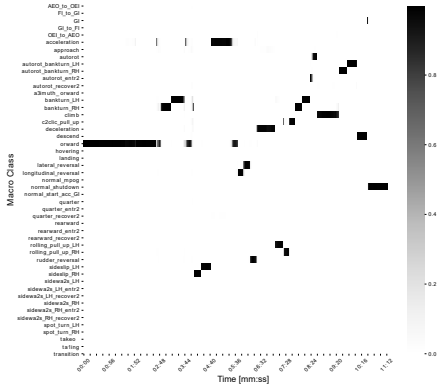


Fig. 11: PV-S Predictions (probability distributions) from the second-level model. Dark colours indicate high model confidence.

X. Conclusions

This paper demonstrated the effectiveness of machine learning in solving the Flight Condition Recognition problem. The proposed framework can be seen as two blocks: the first block produces prediction over a set of sliding windows of different sizes, using a First-level model trained on the entire maneuvers; the second block then joins the prediction using a Second-level stacking model (which we called *Supervisor*). The outcome of the segmentation shows

significant improvement over both single-window predictions and multi-window predictions when aggregated using simpler strategies, with a minimum F1-score increment of 0.7. The results were validated over a massive dataset of Load Survey Flights. To the best of the authors' knowledge, this is the first time that a FCR algorithm (and especially a learning-based one) has been extensively validated on such a large dataset, with real data collected directly by the on-board HUMS. This demonstrates the real-life applicability of our solution.

Furthermore, we believe that our algorithm is flexible and robust enough to deal with a range of time-series segmentation tasks, assuming that suitable window sizes are used and significant features are extracted. The use of multiple windows means that it is able to cope with both stationary and non-stationary time-series, even when these are composed by segments of different lengths.

This solution paves the way to large-scale adoption of FCR, since it is able to scale easily across helicopter models, as long as data is available. Once individual monitoring of the aircrafts is in place, fatigue estimation can be performed in a more precise manner, leading to improved and customized maintenance schedules.

REFERENCES

[1] J. E. Land, "Hums-the benefits-past, present and future," in *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*, vol. 6, pp. 3083–3094, IEEE, 2001.

[2] M. Greaves, "Towards the next generation of hums sensor," in *A Paper Presented at ISASI Seminar, Adelaide*, 2014.

[3] W. Staszewski, G. Tomlinson, C. Boller, and G. Tomlinson, *Health monitoring of aerospace structures*. Wiley Online Library, 2004.

[4] V. Giurgiutiu, A. Cuc, and P. Goodman, "Review of vibration-based helicopters health and usage monitoring methods," tech. rep., South Carolina Univ Columbia Dept of Mechanical Engineering, 2001.

[5] B. D. Larder, "An analysis of hums vibration diagnostic capabilities," *Journal of the American Helicopter Society*, vol. 45, no. 1, pp. 28–33, 2000.

[6] J. J. Zakrajsek, R. F. Handschuh, D. G. Lewicki, and H. J. Decker, "Detecting gear tooth fracture in a high contact ratio face gear mesh.," tech. rep., NATIONAL AERONAUTICS AND SPACE ADMINISTRATION CLEVELAND OH LEWIS RESEARCH , 1995.

[7] K. A. Jason and K. S. Aryan, "A survey of health and usage monitoring system in contemporary aircraft," *International Journal of Engineering and Technical Research (IJETR), ISSN*, pp. 2321–0869, 2013.

[8] D. Goyal and B. Pabla, "The vibration monitoring methods and signal processing techniques for structural health monitoring: a review," *Archives of Computational Methods in Engineering*, vol. 23, no. 4, pp. 585–594, 2016.

[9] P. D. Samuel and D. J. Pines, "A review of vibration-based techniques for helicopter transmission diagnostics," *Journal of sound and vibration*, vol. 282, no. 1-2, pp. 475–508, 2005.

[10] J. Sun, L. Wang, J. Li, F. Li, J. Li, and H. Lu, "Online oil debris monitoring of rotating machinery: A detailed review of more than three decades," *Mechanical Systems and Signal Processing*, vol. 149, p. 107341, 2021.

[11] H. Wei, C. Wenjian, W. Shaoping, and M. M. Tomovic, "Mechanical wear debris feature, detection, and diagnosis: A review," *Chinese Journal of Aeronautics*, vol. 31, no. 5, pp. 867–882, 2018.

[12] A. M. Toms and K. Cassidy, "Filter debris analysis for aircraft engine and gearbox health management," *Journal of Failure Analysis and Prevention*, vol. 8, no. 2, pp. 183–187, 2008.

[13] L. C. J. Milsom, "A study of usage variability and failure probability in a military helicopter," in *Third International Conference on Health and Usage Monitoring-HUMS2003*, p. 81, Citeseer, 2002.

[14] J. Jylhä, M. Ruotsalainen, T. Salonen, H. Janhunen, I. Venäläinen, A. Siljander, and A. Visa, "Link between flight maneuvers and fatigue," in *ICAF 2011 Structural Integrity: Influence of Efficiency and Green Imperatives*, pp. 453–463, Springer, 2011.

[15] P. Shanthakumaran, T. Larchuk, R. Christ, D. Mittleider, E. Hitchcock, and B. Rotorcraft, "Usage based fatigue damage calculation for ah-64 apache dynamic components," in *The American Helicopter Society 66th Annual Forum, Phoenix*, 2010.

[16] D. Lombardo and K. Fraser, "Importance of reliability assessment to helicopter structural component fatigue life prediction," tech. rep., DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION VICTORIA (AUSTRALIA , 2002.

[17] S. Moon and C. Simmerman, "The art of helicopter usage spectrum development," *Journal of the American Helicopter Society*, vol. 53, no. 1, pp. 68–86, 2008.

[18] S. Moon, C. Simmerman, and E. Flores, "Vtuav, fire scout fatigue usage spectrum development," 2009.

[19] R. Romero, H. Summers, and J. Cronkhite, "Feasibility study of a rotorcraft health and usage monitoring system (hums): Results of operator's evaluation.," tech. rep., PETROLEUM HELICOPTERS INC LAFAYETTE LA, 1996.

[20] F. G. Polanco, "Usage spectrum perturbation effects on helicopter component fatigue damage and life-cycle costs," tech. rep., AERONAUTICAL AND MARITIME RESEARCH LAB MELBOURNE (AUSTRALIA), 2000.

[21] L. Molent and B. Aktepe, "Review of fatigue monitoring of agile military aircraft," *Fatigue & fracture of engineering materials & structures*, vol. 23, no. 9, pp. 767–785, 2000.

[22] L. Iseler and J. De Maio, "An analysis of us civil rotorcraft accidents by cost and injury (1990-1996)," tech. rep., NATIONAL AERONAUTICS AND SPACE ADMINISTRATION MOFFETT FIELD CA AMES , 2002.

[23] D. Le and E. Cuevas, "United states federal aviation administration health and usage monitoring system r&d strategic plan and initiatives," in *Proceedings of the 5th DSTO International Conference on Health and Usage Monitoring*, 2007.

[24] R. Teal, J. Evernham, T. Larchuk, D. Miller, D. Marquith, F. White, and D. Deibler, "Regime recognition for mh-47e structural usage monitoring," in *Annual Forum Proceedings-American Helicopter Society*, vol. 53, pp. 1267–1284, AMERICAN HELICOPTER SOCIETY, 1997.

[25] G. Barndt, S. Sarkar, and C. Miller, "Maneuver regime recognition development and verification for h-60 structural monitoring," in *Annual Forum Proceedings-American Helicopter Society*, vol. 63, p. 317, AMERICAN HELICOPTER SOCIETY, INC, 2007.

[26] J. Berry, R. Vaughan, J. Keller, J. Jacobs, P. Grabill, and T. Johnson, "Automatic regime recognition using neural networks," in *Proceedings of American Helicopter Society 60th Annual Forum*, pp. 9–11, 2006.

[27] D. Lombardo, "Helicopter flight condition recognition: A minimalist approach," in *Australian Joint Conference on Artificial Intelligence*, pp. 203–214, Springer, 1998.

[28] M. Al Mansour, I. Chouaib, and A. Jafar, "Maneuver classification of a moving vehicle with six degrees of freedom using logistic regression technique," *Gyroscopy and Navigation*, vol. 9, no. 3, pp. 207–217, 2018.

[29] M. enipek and U. Kalkan, "Learning-based clustering for flight condition recognition," 09 2019.

[30] U. Kalkan, M. Şenipek, and A. Yücekayalı, "Markovian approach for learning based flight condition recognition methods,"

[31] D. Musso and J. Rogers, "Interacting multiple model estimation for helicopter regime recognition," *Journal of Aircraft*, vol. 57, no. 6, pp. 1134–1147, 2020.

[32] R. E. Rajnicek, "Application of kalman filtering to real-time flight regime recognition algorithms in a helicopter health and usage monitoring system," 2008.

[33] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *Journal of artificial intelligence research*, vol. 11, pp. 131–167, 1999.

[34] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[35] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.