



Aerodrome Movement Monitoring Using ADS-B Data: A Case Study at Lommis Airfield

Alex Fustagueras * and Manuel Waltert 

Centre for Aviation, Zurich University of Applied Sciences, Winterthur, Switzerland

*Corresponding author: alex.fustagueras@zhaw.ch

(Received: 30 Oct 2025; Revised: 26 Feb 2026; Accepted: 23 Mar 2026; Published: 24 Mar 2026)

(Editor: Tatiana Polishchuk; Reviewers: Alex Vallés Villanova, Junzi Sun, and Anastasia Lemetti)

Abstract

This study presents and validates a machine learning pipeline that transforms raw Automatic Dependent Surveillance-Broadcast (ADS-B) data into structured aerodrome movement reports, addressing regulatory needs for continuous monitoring of aircraft operations at small, non-towered airfields. The approach automatically identifies aerodrome-specific flight events, particularly repetitive traffic circuits, which constitute a significant portion of General Aviation traffic at such airfields. Using ADS-B data observed at Lommis Airfield, a representative regional airfield in Switzerland, we filtered and preprocessed raw flight trajectories and segmented those meeting the filtering criteria into aerodrome circuit candidates. We then formulated circuit detection as a supervised binary classification problem and compared five machine learning approaches: Logistic Regression, Random Forest, unidirectional and bidirectional Long Short-Term Memory (LSTM) networks, and a 1D Convolutional Neural Network (CNN). Each traffic circuit candidate was characterised by eight engineered features capturing kinematic and flight-phase information. The 1D CNN model achieved 99.15% accuracy, outperforming rule-based heuristics by 25.5 percentage points in recall, while simpler models (Logistic Regression, Random Forest) reached comparable performance with higher interpretability and efficiency. End-to-end validation of the proposed pipeline over a three-month period yielded 67.6% overall detection coverage (438 of 648 flights), limited primarily by ADS-B data availability rather than model performance. The validated pipeline demonstrates the potential for a scalable path toward automated, data-driven movement reporting, with full end-to-end validation conducted at a single airfield and additional cross-airfield evidence shown at the model level.

Keywords: Aerodrome movement monitoring; ADS-B data; machine learning; traffic circuits; non-towered airfield

Abbreviations: JOAS: Journal of Open Aviation Science, ATM: Air Traffic Management, FOCA: Federal Office of Civil Aviation, ADS-B: Automatic Dependent Surveillance-Broadcast, ML: Machine Learning,

1. Introduction

Aerodrome movement monitoring refers to the continuous observation, registration, and analysis of aircraft operations at an airport or airfield, including take-offs, landings, taxiing, and touch-and-go manoeuvres. Such detailed records are essential for infrastructure operations and planning, safety evaluations, environmental assessments, resource allocation, or regulatory compliance [1]. To ensure aerodrome movement monitoring, national aviation authorities often mandate standardised logs that capture timestamps, operation types, aircraft identifiers, runway usage, and route origin/destination [2]. For example, in Switzerland, all public-use aerodromes, even those without control towers, are required to submit monthly movement statistics to the Federal Office of Civil

Aviation (FOCA) in prescribed formats, such as the one depicted in Figure 1.

Currently, many small, uncontrolled (i.e., non-towered) airfields rely on manual or semi-manual methods to register aircraft movements. Common practices include handwritten logbooks, manual tallies based on radio communications, or spreadsheets updated by ground staff [3]. Despite their prevalence, these traditional approaches are labour-intensive, error-prone, and susceptible to data inconsistencies, omissions, and transcription errors [2, 4]. Recent studies confirm that such methods often produce incomplete datasets and unreliable statistics [5].

AIRPORT	DAY	MONTH	YEAR	TIME	ARR_DEP	REASON	AIRCRAFT	REGISTRATION	RUNWAY	MOVEMENTS	PASSENGERS	FREIGHT	ORI_DEST	ROUTE
LSR1	26	12	2025	1015	A	1	A	HBAAA	30	1	3	0	LSZA	SO
LSR1	26	12	2025	1115	D	1	A	HBAAA	30	1	2	0	LSZA	SO
LSR1	26	12	2025	1405	D	6	A	HB BBB	30	1	0	0	LSR1	SW
LSR1	26	12	2025	1405	V	6	A	HB BBB	30	10	0	0	LSR1	SW
LSR1	26	12	2025	1430	A	6	A	HB BBB	30	1	0	0	LSR1	SW
LSR1	27	12	2025	1435	A	4	C	HB2000	12	1	0	0	LSPL	NW
LSR1	27	12	2025	1500	D	4	C	HB2000	12	1	0	0	LSPL	NW
LSR1	28	12	2025	925	A	3	B	HBXXX	6	7	0	7000	LSZZ	SW
LSR1	28	12	2025	935	D	3	B	HBXXX	24	7	0	1400	LSZZ	SW

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Aerodrome	Day	Month	Year	Local Time	Flight Type	Reason for Flight	Aircraft Type	Registration Number	Runway	Number of movements	Passengers	Freight	Origin / Destination	Flight Direction

Figure 1. Section of the official form mandated by the Swiss Federal Office of Civil Aviation (FOCA) for the monthly submission of flight movement statistics by public-use aerodromes. Adapted from Federal Office of Civil Aviation (FOCA) [6].

To address these shortcomings, information from surveillance technologies has emerged as a valuable resource for aerodrome movement monitoring. Among major airports, established systems such as RADAR and Multilateration are widely used to provide real-time tracking of surface movements [7]. While these systems offer high-fidelity data, ADS-B has gained increased attention as a more accessible and cost-effective alternative for airfields. Recent studies have demonstrated the potential of ADS-B for airport applications, especially at aerodromes with excellent coverage, enabling continuous tracking of aircraft during surface operations and at lower altitudes. This includes efforts to derive operational milestones of flights, such as touchdown, runway exit, etc. from ADS-B messages to support data-driven airport management [8] and to implement Collaborative Decision Making (A-CDM) functionalities using only ADS-B data [9]. This capability for close-range, low-cost aerodrome monitoring is notably met at Lommis Airfield (LSZT), a representative regional, non-towered airfield in Switzerland. There, improved ADS-B reception, achieved after the installation of a dedicated, low-cost receiver by the authors, has allowed for detailed trajectory-based analyses.

The sheer volume and heterogeneous quality of raw ADS-B trajectories often necessitate automated methods to identify, classify, and analyse flight manoeuvres. In the literature, two main methodological paradigms can be distinguished: rule-based and machine-learning-based approaches [10]. Rule-based methods rely on deterministic kinematic criteria, such as thresholds on airspeed, altitude, or vertical rate, to infer specific operational states or manoeuvres. Such approaches have been widely applied for the detection and/or timestamping of characteristic flight phases and surface movements, including go-around prediction [11, 12], landing and take-off time determination [13], push-back detection [14], taxi-time estimation [14, 15], or the identification of Arrival Sequencing and Metering Area (ASMA) times [12]. Moreover, rule-based techniques have been used to infer runway configurations at airports [16].

In contrast, data-driven and machine-learning-based methods aim to capture more complex, non-linear relationships within flight trajectories, leveraging pattern recognition and statistical learning to generalise across diverse operational contexts. For instance, Olive and Morio [17] clustered traffic flows in the vicinity of aerodromes by using the DBSCAN algorithm. Kumar et al. [18] utilised an unsupervised clustering algorithm (HDBSCAN) to analyse the characteristics of go-around events, effectively distinguishing between 'nominal' and 'anomalous' trajectories based on their energetic and kinematic features. Dhief et al. [19] proposed a tree-based model (XGBoost) to identify go-arounds from flight records, demonstrating the applicability of supervised methods to this problem.

Other sophisticated manoeuvres, such as airborne holding patterns, have also been successfully identified using supervised learning. Moreover, Olive et al. [20] trained a convolutional neural network to classify trajectory segments, accurately detecting the distinct racetrack shape of holding patterns. These demonstrated capabilities reinforce that Machine Learning (ML) is a promising candidate for detecting complex, non-linear trajectory patterns.

Despite these advances, reliable detection of a particularly important and frequent General Aviation (GA) procedure, the aerodrome traffic circuit, remains an open challenge. This manoeuvre, commonly known as "*Platzrunde*" in German or "*tours de piste*" in French, involves an aircraft taking off, flying a predefined rectangular circuit around the runway, and landing again [21]. Since aerodrome traffic circuits are often repeated multiple times during training flights, they constitute a significant proportion of GA activity and are highly relevant for operational efficiency and airspace safety due to their frequency and concentration around small airfields [22]. The primary difficulty in their automated detection lies in the discrepancy between the circuit's standard definition and its highly variable real-world execution. While existing research has addressed more granular aspects of flights, such as classifying discrete approach outcomes (e.g., a touch-and-go landing) [23] or identifying elementary flight phases (e.g., climb, descent) using fuzzy logic [24], these methods fall short of identifying the complete, multi-stage operational pattern of aerodrome traffic circuits. Simply chaining elementary flight phases together with rules proves insufficient, as such methods lack the robustness to handle the free-form and variable nature of GA operations and require extensive fine-tuning that does not generalise well [4].

Therefore, this paper addresses the primary research question of how ML can be leveraged to transform raw ADS-B time-series data into accurate, standardised aerodrome movement reports at non-towered airfields. To answer this, our study pursues three main objectives: (i) developing a robust method for detecting take-off and landing events by adapting existing toolsets to the specific constraints of small airfields like LSZT; (ii) designing, training, and evaluating a range of ML architectures, from classical classifiers (Random Forest, Logistic Regression) to deep learning models (CNN, LSTM), in order to identify the optimal trade-off between detection accuracy and computational cost; and (iii) extracting relevant metadata (aircraft identifier, inferred aircraft type when available, timestamps, and runway usage) and integrating detected events into a rule-based module that produces standardised movement records compatible with national reporting templates. A case study at LSZT serves to demonstrate the effectiveness of the pipeline and its potential for broader applicability.

The remainder of this paper is structured as follows. Section 2 presents the overall methodology, including the acquisition and preprocessing of raw ADS-B trajectories, the proposed ML approach for detecting and classifying aerodrome events, and the procedure for compiling the standardised aerodrome movement log. Then, Section 3 describes the results, which are subsequently discussed in Section 4. Finally, Section 5 summarises the main findings and outlines directions for future work.

2. Methods

The methodological framework developed for this study was designed to systematically transform raw ADS-B surveillance data into a structured record of aerodrome movements. This process is structured as a multi-stage pipeline where each module prepares the data for the next. To this end, Section 2.1 explains the acquisition and preprocessing of the ADS-B trajectory data used in this study, ensuring that the raw signals are suitable for geometric analysis. Afterwards, these trajectories are subdivided into individual aerodrome circuit candidates by identifying potential arrival and departure segments, as detailed in Section 2.2.

Section 2.3 presents the core methodologies used to identify and classify specific aerodrome events within these segments. Specifically, Section 2.3.1 describes the rule-based procedures for detecting

discrete take-offs and landings based on vertical and horizontal movement. Building upon these results, Sections 2.3.2 and 2.3.3 present a comparative evaluation of rule-based and supervised ML approaches, respectively, for the higher-level task of detecting and classifying traffic circuits from the previously identified candidates. Finally, Section 2.4 outlines the integration procedure for compiling these disparate detections into a single, cohesive, and structured record of aerodrome movements.

2.1 Data Acquisition and Preprocessing

We used historical ADS-B trajectory data obtained from the OpenSky Network [25] covering an observation period from December 2024 to July 2025. To ensure the resulting models were robust and generalisable beyond a single aerodrome location, a diverse dataset was curated. This dataset was primarily composed of trajectories of aircraft operating from Lommis Airfield (LSZT) but was supplemented with operations from other Swiss GA airfields, such as Lausanne Airport (LSGL) and Speck-Fehraltorf Airfield (LSZK), providing examples of varied aircraft types and circuit patterns.

To isolate traffic relevant to aerodrome operations, the dataset was filtered: Trajectories were kept only if the aircraft descended to a minimum of 300 feet or less above the aerodrome field elevation, an informed middle ground between 500 feet and 150 feet thresholds identified by Karboviak et al. [23] that effectively isolates committed landing sequences from overflight noise, and demonstrated alignment with a runway, specifically targeting final approach or initial departure paths. A final check ensured the retained flights came within one nautical mile or less of the runway centre point. Data retrieval and manipulation were performed using the `traffic` Python library [26], which provided essential functions for trajectory analysis. Specifically, temporal slicing was accomplished via `.between()` to isolate trajectory segments, distance computations using `.distance()` for feature extraction, and runway-specific filtering with `.inside_bbox()` to identify aerodrome circuit candidates. For data quality assurance, missing numerical values (position, speed, and track data) were handled using forward and backward imputation, and any entirely NaN values or all zeros in the numeric columns were explicitly discarded.

2.2 Trajectory Segmentation

The filtered trajectories were subsequently segmented into individual *aerodrome circuit candidates* using a polygon-based overflight detection method, where a distinct rectangular polygon is defined for each runway of the airport at which the aircraft operate. This polygon is aligned with the runway's longitudinal axis and positioned according to the geographic coordinates of its thresholds, which are obtained from the `traffic` library's airport database. The runway threshold coordinates are used to construct the rectangular polygon that defines the capture area for each runway, as illustrated in Figure 2.

This approach was selected because it offers a simple and reliable means of identifying runway crossings across varying traffic and operational conditions. Alternative methods based on Instrument Landing System (ILS) alignment, which rely on angular and distance tolerances within temporal windows, proved unsuitable for GA traffic at non-towered airfields: Their sensitivity to shallow approach angles, irregular circuit geometries, and visual flight deviations often caused valid manoeuvres to be missed. In contrast, the polygon-based method depends only on the geometric relation between the aircraft trajectory and the runway area, making it robust and broadly applicable.

Whenever an aircraft trajectory intersected the *runway polygon*, we recorded an *overflight event*. Because aircraft operating under Visual Flight Rules (VFR) rarely align perfectly with the extended runway centreline during approach, for example due to crosswinds or manual flight control, the *runway polygon* was intentionally defined wider than the actual runway. This width adjustment, realised through a width scaling factor greater than one, ensured that small lateral deviations did not prevent valid overflight detections. The resulting overflight timestamps provided natural delimiters

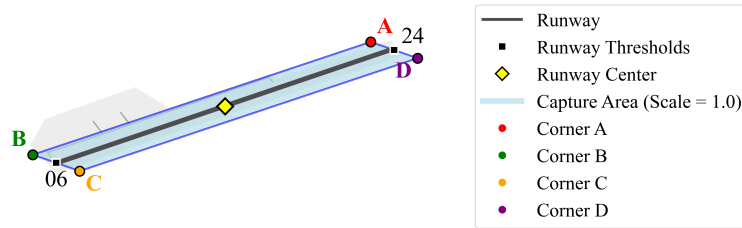


Figure 2. Definition of the polygon-based runway detection at LSZT. The rectangle defined by the thresholds of runway 06/24 serves as the basis for trajectory segmentation. The scale parameter (scale = 1.0 in this visualisation) controls the extent of the capture area around the runway, allowing for adjustment to accommodate flight deviations.

for segmenting continuous trajectories into a number of discrete time series, each representing an *aerodrome circuit candidate*.

After segmentation, each resulting *aerodrome circuit candidate* was stored as an individual time series containing timestamped kinematic states of the aircraft. Figure 3 illustrates segmented circuit candidates and corresponding flight patterns for an aircraft performing a pilot training flight at LSZT. Specifically, the figure shows raw data from three representative aerodrome circuit candidates, depicted in terms of the aircraft's distance from the runway centreline, altitude profile, unwrapped track, and geographic position.

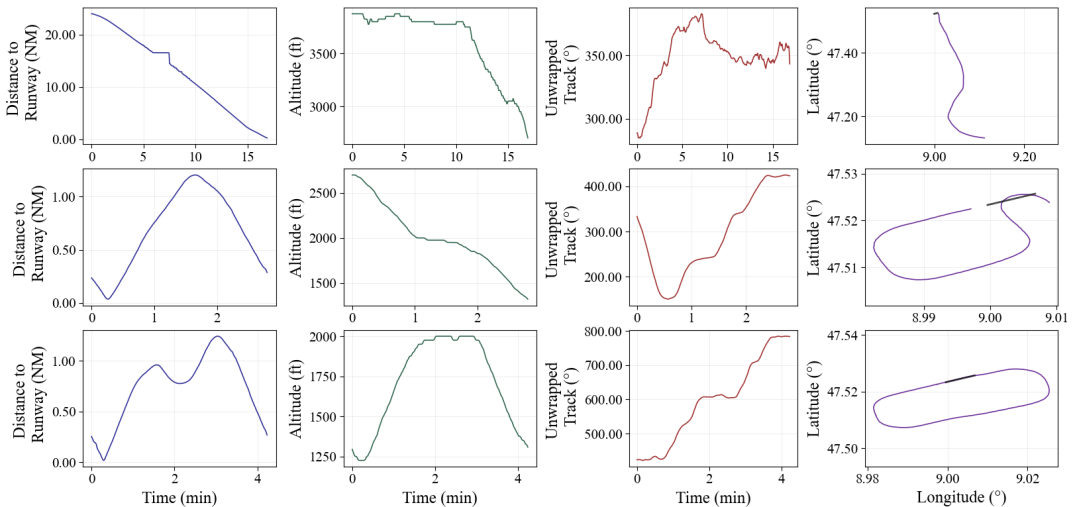


Figure 3. Example of a segmented trajectory from aircraft HB-KLA landing at LSZT. Each coloured segment corresponds to a flight manoeuvre or circuit candidate, providing excellent material for further analysis of the segments.

2.3 Aerodrome Event Detection and Classification

The event detection aims to populate the structured record of aerodrome movements with information on arrivals, departures, and traffic circuits. In the present work, we prioritised the reliable identification of traffic circuits as an operationally relevant, recurring class of manoeuvre. We deliberately did not distinguish the antecedent event that precipitated a detected circuit, whether it was a continuation following a micro-event (e.g., a touch-and-go) or an independent commencement of the pattern. The taxonomy and scope used to construct the record of aerodrome movements are

therefore focused on circuit recognition as a high-level class, independent of preceding micro-events such as touch-and-go or aborted landings (go-around). To this end, we first present in Section 2.3.1 a number of rule-based methods by means of which we detect take-off and landing events. Subsequently, Section 2.3.2 focuses on a rule-based method to detect aerodrome circuits, while Section 2.3.3 presents ML methods used to detect and classify aerodrome circuits on the basis of *aerodrome circuit candidates*.

2.3.1 Rule-Based Detection of Events

A deterministic, rule-based detector constituted the first analytical layer for detecting take-off and landing events on the full flight trajectory. As such, take-off and landing events were identified together with their corresponding timestamps. A take-off was detected when a flight transitioned from a period of ground movement along the runway to a continuous increase in altitude and airspeed. Specifically, take-off detection involved extracting the first minute of a climb segment, computing the distance to both runway thresholds, and selecting the threshold crossed more recently as the departure point. Conversely, landing detection used the `landing_at()` method from the `traffic` library to identify when an aircraft lands at the specified aerodrome. For detected landings, the algorithm extracted the final minute of the flight trajectory by identifying the last change in vertical rate, computed proximity to both runway thresholds during this final segment, and selected the closest threshold crossing as the landing point.

2.3.2 Rule-based Traffic Circuit Detection

We developed a rule-based traffic circuit detection algorithm which uses rule-based heuristics, combining geometric and kinematic features (e.g., distance to the runway, altitude profile, and cumulative unwrapped track change) with signal processing techniques. These heuristics employed peak-valley detection using SciPy library [27] to identify local maxima (peaks) and local minima (valleys) in distance data, and validated circuit sequences through flight phase patterns. We applied a prominence threshold of 0.1 to extract meaningful flight dynamics while filtering out signal noise. These flight phases (e.g., climb, level flight, descent) were identified using the `OpenAP.phases()` routine from [28]. The resulting heuristics were formulated to recognise repeated, spatially confined laps and to count their repetitions, leveraging the features illustrated in Figure 4.

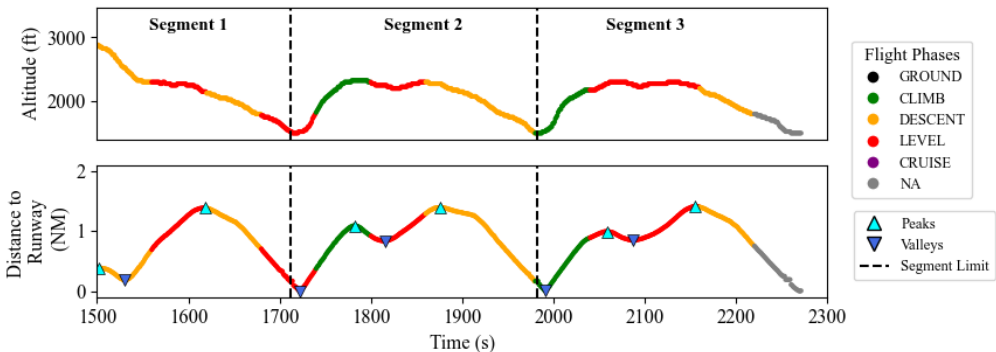


Figure 4. Illustration of the kinematic and signal features used in the exploratory rule-based heuristic approach for circuit detection. The flight trajectory is depicted alongside its altitude profile and distance to the runway, with signal-processed peaks and valleys identified as inputs for the heuristic rules.

2.3.3 Supervised Machine Learning Traffic Circuit Classification

Our set of *aerodrome circuit candidates* comprises a wide variety of flight patterns, of which only a subset represents *true* aerodrome circuits. We therefore developed a supervised ML method to assign each candidate to one of the following two categories: *traffic circuit* or *not a circuit*. The main

challenge of this classification lies in reliably identifying traffic circuits given their high variability as they are predominantly flown under Visual Flight Rules (VFR). For this purpose, we applied the following definition of a traffic circuit as the classification criterion: an *aerodrome circuit* is a recurrent flight pattern designed to return an aircraft to the runway, typically consisting of an *upwind leg*, *crosswind leg*, *downwind leg*, *base leg*, and *final leg*. Unlike standardised instrument procedures, these VFR patterns are strongly influenced by pilot technique and environmental factors, resulting in a wide range of valid geometries.

The diversity of *traffic circuits* observed in our dataset is illustrated in Figure 5, which shows canonical left-hand and right-hand patterns, repeated sequences, and other atypical yet legitimate circuits.

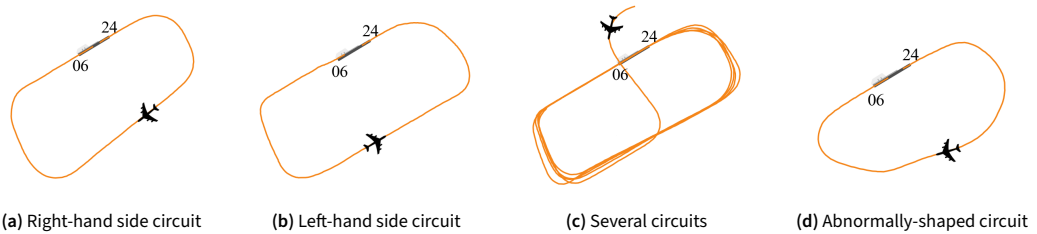


Figure 5. Examples of segmented *traffic circuits* at LSZT, showing (a) a right-hand circuit, (b) a left-hand circuit, (c) multiple consecutive circuits, and (d) an abnormally shaped variant

Our dataset of *aerodrome circuit candidates* also contains segments which are *not a circuit*. These flight trajectories intersect the *runway polygon* but do not form a complete traffic circuit as defined above. The *not a circuit* trajectory segments, some examples of which are shown in Figure 6, often include partial manoeuvres, perpendicular runway crossings, or wide deviations that superficially resemble circuit legs. Accurately filtering such false positives is therefore just as important as identifying true circuits to ensure the integrity of the final structured record of aerodrome movements.

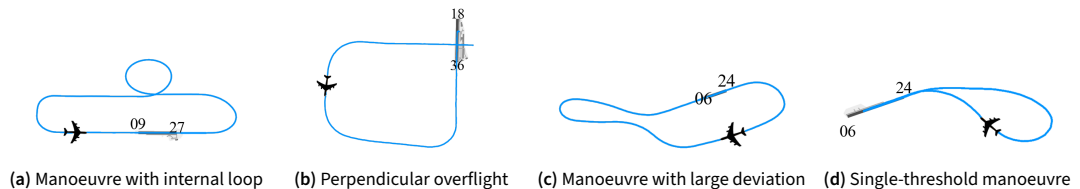


Figure 6. Examples of segmented trajectories that must be correctly identified as non-circuits. These movements, while crossing the runway area, do not constitute a complete traffic circuit pattern.

We formulated the traffic circuit detection problem as a supervised binary classification task, labelling each preprocessed *aerodrome circuit candidate* as either *traffic circuit* or *not a circuit*. A manually constructed ground-truth dataset was developed using the collected ADS-B data from Section 2.1 by visually inspecting and annotating *aerodrome circuit candidates*, with edge cases resolved through interactive review, and low-quality segments excluded from the training pool. The final dataset consisted of 3297 labelled segments, with 695 labelled as *traffic circuit* and 2422 as *not a circuit*, representing an intrinsic class imbalance. The labelled dataset was split using stratified random splitting (70% train, 15% validation, 15% test) to create training, validation, and held-out test sets. Due to the class imbalance, class weighting was employed during model training to mitigate bias.

For supervised learning, each *aerodrome circuit candidate* was transformed into a fixed-length multivariate time series of shape (500,8) to ensure a uniform input tensor with sufficient resolution to capture circuit dynamics. This transformation was achieved by resampling the variable-length segments: trajectories longer than 500 points were downsampled by selecting 500 evenly spaced indices,

while shorter trajectories (with more than one point) were upsampled to 500 points using linear interpolation. In the rare case of a single-point segment, that point’s values were repeated to fill the tensor. Eight features were engineered to capture the essential characteristics of aircraft performing a circuit, comprising four kinematic and four flight phase related features. The kinematic features included (i) the horizontal distance of the aircraft from the runway centre point, (ii) the barometric altitude, (iii) the Runway Longitudinal Alignment Angle (RLAA), and (iv) the unwrapped track. As illustrated in Figure 7, the RLAA measures the unsigned angle between the vector from the runway centre point to the aircraft and the runway longitudinal axis, thereby capturing the aircraft’s lateral position relative to the extended centreline. The unwrapped track, in contrast, provides a monotonically increasing signal of cumulative heading change, exposing the cyclical nature of the manoeuvre through repeated accumulations of 360° . The remaining four features are one hot encoded flight phases (CLIMB, DESCENT, LEVEL, NA) derived using the `OpenAP.phases()` routine from [28], which provide temporal context at each time step. The integration of these temporal features—linking rotation (unwrapped track) with alignment (RLAA) and enriched by flight phase information—forms the core methodological contribution of this work.

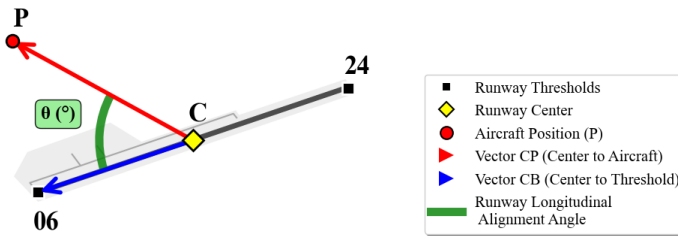


Figure 7. Visualisation of the Runway Longitudinal Alignment Angle (RLAA), defined as the unsigned angle between the vector from the runway centre point to the aircraft and the runway centreline

Prior to model ingestion, all features were standardised using *scikit-learn*’s `StandardScaler` [29], fitted exclusively on the training set and applied consistently to the validation and test partitions. For sequential models, standardisation was applied per feature across the time dimension to preserve the internal temporal structure.

While aerodrome traffic circuits are characterised by an ordered progression of spatial and kinematic states across characteristic phases such as take-off, climb, *upwind leg*, *crosswind leg*, *downwind leg*, *base leg*, *final leg*, and landing, we explored both temporal and non-temporal modelling approaches to determine their relative effectiveness. We therefore implemented five distinct architectures: two non-sequential baselines (Logistic Regression and Random Forest) implemented using *scikit-learn*, and three sequential architectures (LSTM, Bidirectional LSTM, and 1D CNN) implemented using the *TensorFlow* library [30]. This ensemble spans a principled spectrum from simple, interpretable statistical summaries to temporal-based deep learning architectures.

We began with the two non-sequential baseline models, which provide interpretability and establish performance baselines. The Logistic Regression classifier was applied to engineered statistical features, computing five summary statistics (mean, standard deviation, minimum, maximum, and median) for each of the eight input channels, resulting in a 40-dimensional descriptor. This approach captures aggregate circuit characteristics without temporal ordering. The Random Forest classifier utilised the same engineered features with an ensemble of 100 estimators and `min_samples_leaf=10` to effectively manage complexity. Both baseline models provide highly interpretable feature importance scores and serve as interpretability benchmarks.

For temporal modelling, we implemented Long Short-Term Memory (LSTM) networks and their bidirectional variants. LSTMs are architecturally designed to capture long-term temporal dependencies

and learn order-sensitive patterns through their gating mechanisms. We trained a unidirectional LSTM with two stacked layers (64 and 32 units) using *TensorFlow*. Subsequently, we developed a bidirectional LSTM (BLSTM) architecture with two stacked bidirectional layers (128 units in the first layer and 64 units in the second), enabling pattern recognition in both forward and backward temporal directions. This bidirectional design captures the full temporal context, as circuit patterns may exhibit characteristic signatures when analysed in either direction. Both LSTM architectures were regularised via dropout (rate=0.3) and trained with the Adam optimiser while utilising class weights. Early stopping based on validation loss was employed to prevent overfitting.

To explore convolutional approaches for temporal feature extraction, we implemented a 1D CNN architecture. Convolutional layers can effectively identify local temporal patterns and invariants across different phases of the circuit, making them suitable for recognising characteristic sequences such as the progression from take-off through crosswind, downwind, base, and final legs. The 1D CNN employed multiple convolutional layers followed by pooling operations to progressively extract hierarchical temporal features, culminating in classification layers.

All models were trained using the manually annotated *aerodrome circuit candidates* dataset with stratified random split. The stratification was performed across candidates, meaning multiple candidates from the same flight could appear in different sets. Hyperparameters for all models were systematically optimised on the validation set using grid search and manual tuning procedures. This training methodology reflects the operational scenario where the model processes each candidate independently, enabling robust evaluation even when candidates from the same flight are distributed across train, validation, and test sets.

For the evaluation of the ML models, we assessed performance on the held-out test set using standard classification metrics from *scikit-learn* including *accuracy*, *F1 score*, *ROC AUC* (Receiver Operating Characteristic - Area Under the Curve), and *average precision*. To compare the outputs of the best performing ML model with the results of the rule-based aerodrome circuit detector from Section 2.3.2, an additional evaluation was conducted at the flight level by aggregating predictions across all candidates belonging to each test flight. Since rule-based detectors operate on complete flight trajectories rather than isolated segments, a direct comparison required aggregating predictions at the flight level. While this approach means that some candidates from test flights may have been seen during training (due to the segment-level stratification), model predictions are made independently on each candidate, and ground truth labels are derived from all labelled candidates of test flights regardless of their original split assignment. This ensures a fair comparison between ML and rule-based approaches, as both methods are evaluated on complete flight trajectories with the same ground truth structure.

2.4 Automated Generation of Structured Record of Aerodrome Movements

The final stage of the pipeline integrates the outputs from the rule-based event detector and the ML circuit classifier to generate a structured, regulatory-compliant record of aerodrome movements. This integration is accomplished through a function that processes each flight trajectory to detect take-offs and landings (via the rule-based detector in Section 2.3.1), segment the trajectory into circuit candidates (via the polygon-based method in Section 2.2), and classify these candidates using the ML classifier (from Section 2.3.3). For each detected event, whether it is a take-off, a landing, or a traffic circuit, the system extracts critical metadata including the aircraft identifier, timestamps, inferred runway usage, and approximate route direction. The route direction is determined geometrically from the aircraft's approach or departure path using the entry point relative to the airport centre coordinates, providing a simplified cardinal direction indicator (e.g., NE for north-east, SW for south-west) that approximates the approach or departure orientation.

To validate the performance of this pipeline, we conducted a case study using data exclusively from Lommis Airfield (LSZT). For this validation, ADS-B data for the vicinity of LSZT was programmatically fetched over a representative three-month period (January to March 2025), encompassing both weekday training activity and weekend leisure operations. Each flight was processed through the complete pipeline, generating a list of detected aerodrome movements. This detected data was then sorted chronologically by date and time and formatted into a tabular structure conforming to the Swiss Federal Office of Civil Aviation (FOCA) reporting template. The resulting automatically generated structured record of aerodrome movements was compared against manually compiled ground truth movement records for the same period to assess the system’s accuracy and reliability.

3. Results

This section presents the results of the aerodrome movement detection pipeline, evaluated through three complementary analyses: (i) *aerodrome circuit candidate* level classification performance of multiple ML models on the labelled dataset; (ii) comparison between ML-based and rule-based circuit detection on a per-flight basis; and (iii) end-to-end validation of the complete pipeline against ground truth records from Lommis Airfield over a three-month period.

3.1 Machine Learning Model Performance

The *aerodrome circuit candidate* level classification performance of all five ML models presented in Section 2.3.3 on the test set is summarised in Table 1. The evaluated models include two non-sequential approaches and three sequential architectures. Performance is assessed using four metrics: *ROC AUC*, which measures discriminative capacity across classification thresholds; *average precision (AP)*, which quantifies precision-recall performance for the imbalanced dataset; *F1-score*, representing the harmonic mean of precision and recall; and *Accuracy*, representing the proportion of correctly classified *aerodrome circuit candidates*. The columns of Table 1 correspond to these four metrics, and for each metric, the highest value among all models is highlighted in bold.

Table 1. *Aerodrome circuit candidate* level classification performance of the ML models on the test set. All metrics except ROC AUC are evaluated at the optimal decision threshold derived from the precision-recall curve.

Model	ROC AUC	AP	F1-Score	Accuracy
Logistic Regression	0.9982	0.9941	0.9577	0.9808
Random Forest	0.9973	0.9915	0.9533	0.9786
LSTM	0.9725	0.9296	0.9124	0.9594
BLSTM	0.9904	0.9687	0.9458	0.9765
1D CNN	0.9995	0.9984	0.9810	0.9915

A comparative visual analysis of model performance is presented in Figure 8, which shows both ROC and precision-recall curves for all five models. The ROC curve evaluates the true positive rate against the false positive rate across all classification thresholds, while the precision-recall curve focuses on the trade-off between precision and recall. The precision-recall representation is particularly informative for imbalanced datasets like ours, where it provides better insight into model performance on the minority (positive) class.

The relative importance of the input features for the non-sequential models is illustrated in Figure 9. For the Random Forest model, importance is measured via Gini importance, which quantifies each feature’s contribution to reducing node impurity. For the Logistic Regression model, importance is represented by signed coefficients to indicate both the magnitude and the direction of the feature’s influence on the classification probability.

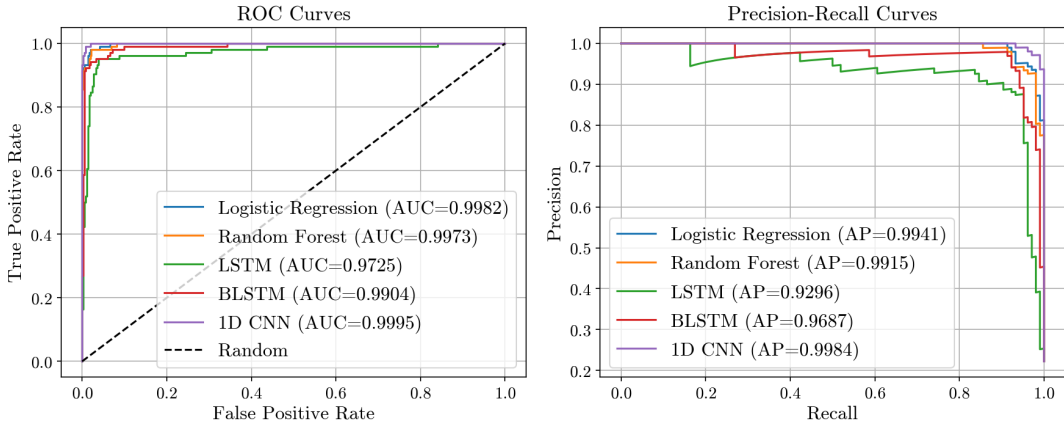


Figure 8. ROC and precision-recall curves for the evaluated ML models on the test set

Across both models, the standard deviation (*_std*) of various parameters emerges as a consistently significant predictor for classification. Specifically, the runway centreline alignment (*Alignment_std*) is the most influential feature in both architectures. In the Random Forest model, this is followed by *Track_std* and *Phase_CLIMB_std*. For Logistic Regression, the magnitude of the *Alignment_std* coefficient similarly dominates, followed by *Altitude_std* and *Phase_DESCENT_std*. Regarding flight phases, variables related to *climb* and *descent* exhibit higher importance compared to other phase-related features. Among top 12 features shown, *Altitude_max* contributes the least in both models.

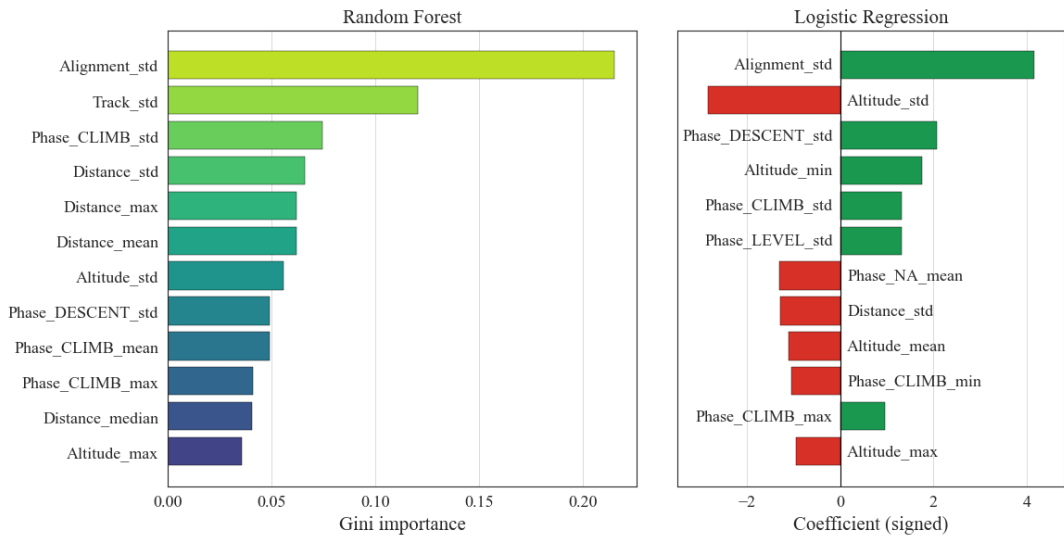


Figure 9. Top 12 feature importances for the Random Forest model (left) and the Logistic Regression model (right)

3.2 ML-Based vs Rule-Based Circuit Detection Performance

To assess the operational applicability of the best-performing ML classifier, in the case of this study the 1D CNN model, in comparison with the rule-based heuristics for circuit detection, we evaluated both approaches on 315 flights from the held-out test set. Table 2 presents the performance of both detectors on a per-flight basis.

Table 2. Per-flight performance comparison between best-performing ML classifier and the rule-based circuit detector

Detector	Accuracy	Precision	Recall	F1-Score
ML-Based	97.8%	100.0%	93.9%	96.8%
Rule-Based	87.3%	95.1%	68.4%	79.6%

Performance differences between ML-based and rule-based detectors were observed across all metrics. The ML-based detector consistently outperformed the rule-based approach: accuracy improved from 87.3% to 97.8%, precision from 95.1% to 100.0%, recall from 68.4% to 93.9%, and F1-score from 79.6% to 96.8%. The magnitude of improvement varied across metrics, with the largest gains observed in recall (25.5 percentage points) and F1-score (17.2 percentage points).

3.3 End-to-End Pipeline Validation

To evaluate the effectiveness of the complete pipeline in an operational context, a three-month case study was conducted at LSZT from January to March 2025. During this period, the pipeline processed ADS-B trajectory data and automatically generated a structured record of aerodrome movements. The resulting record was compared with manually compiled ground truth data provided by the airfield operator. Table 3 summarises detection statistics and overall performance on a monthly basis. The analysis differentiates between flights that could not be detected due to missing ADS-B data and those missed by the detector despite available data. Accordingly, Table 3 lists the monthly flight movements reported by the airfield operator and those detected from ADS-B data in the columns *Ground Truth* and *Detected Flights*, respectively. The column *Detection Rate* indicates the percentage of successfully detected flights relative to the ground truth, while *Pipeline Performance* specifies the true detector performance by showing the percentage of flights for which ADS-B data were available and successfully detected by the pipeline.

Table 3. Validation results comparing pipeline detection against ground truth records at LSZT (January to March 2025)

Month	Ground Truth (ATM)	Detected Flights (ATM)	Detection Rate	Pipeline Performance
January	74	46	62.2%	100.0%
February	127	96	75.6%	100.0%
March	447	296	66.2%	98.2%
Overall	648	438	67.6%	99.1%

Across the three-month observation period, the automated pipeline detected 438 of the 648 flights listed in the ground truth records, yielding an average detection rate of 67.6%. Detection performance varied by month, with the highest rate observed in February (75.6%) and the lowest in January (62.2%). An analysis of the 210 undetected flights revealed that 202 cases (96.2%) resulted from missing ADS-B data in the OpenSky Network dataset, while only 8 flights (3.8%) were missed by the detector despite available ADS-B data. When considering only flights with available ADS-B coverage, the pipeline achieved an average detection rate of 99.1%.

4. Discussions

Our study successfully demonstrated that standardised movement reports for small, non-towered airfields can be generated from ADS-B data. These reports include detailed information about each flight operating at the airfield, including timestamps of key milestones such as take-off and landing as well as information on aerodrome traffic circuits. The methods for detecting take-off and landing events and for determining flight phases (climb, cruise, descent) have already been described in the

literature and were successfully implemented and applied by us in a non-towered airfield environment. However, for the detection and classification of aerodrome traffic circuits, which also need to be included in standardised movement reports, no methods were previously available. Therefore, we presented and evaluated suitable approaches in this study.

Our results suggest that ML-based detection of aerodrome traffic circuits from ADS-B trajectories substantially outperforms traditional rule-based heuristics for circuit identification at non-towered airfields. Our best-performing ML classifier, which is a 1D CNN, achieved an F1-score of 96.8% and an accuracy of 97.8% on the test set, significantly exceeding the performance of the rule-based approach (87.3% accuracy, 79.6% F1-score). The largest improvement was observed in recall (93.9% vs. 68.4%), representing a 25.5 percentage-point increase that is operationally critical, as missed circuit detections directly affect the completeness and reliability of movement logs required for regulatory compliance. These results validate that supervised ML approaches can overcome the inherent limitations of deterministic rule-based methods when applied to highly variable VFR operations.

The rule-based heuristics used to detect aerodrome traffic circuits proved brittle due to ADS-B irregularities and measurement jitter, where signal noise and intermittent data gaps produced spurious extrema that disrupted peak-valley detection and phase sequence rules. Moreover, legitimate circuit executions frequently departed from *textbook geometry* due to crosswinds, pilot technique variations, and environmental factors, creating edge cases outside fixed threshold windows. This finding aligns with observations in related aviation literature. Fala, Georgalis, and Arzamani [4] noted similar limitations when applying rule-based methods to GA operations, highlighting their dependence on extensive fine-tuning and resulting in large variance in misidentification rates. Zhang et al. [5] demonstrated that rule-based approaches for GA flight phase identification often produce unreliable statistics when dealing with the free-form nature of VFR operations. Our supervised ML approach overcomes these limitations by learning robust decision boundaries from labelled examples rather than relying on brittle threshold-based rules.

The comparative model analysis revealed that our 1D CNN model achieved the highest performance among all tested ML models (99.15% accuracy, 0.9995 ROC AUC), demonstrating that convolutional architectures are exceptionally well-suited for detecting local temporal patterns inherent in circuit manoeuvres. More notably, engineered statistical features enabled simple baseline models to achieve exceptional performance using only 40-dimensional feature vectors derived from basic statistical aggregations (mean, standard deviation, minimum, maximum, median) across the eight input channels. Logistic Regression reached 98.08% accuracy and Random Forest 97.86% accuracy, rivalling the CNN while offering greater computational efficiency and interpretability. Feature importance analysis revealed that standard deviation metrics, particularly RLAA variability (*Alignment_std*), dominated predictive power across both interpretable models, suggesting that pattern regularity and geometric consistency are key discriminators of circuit behaviour. That such simple statistical abstractions achieve performance comparable to complex deep learning architectures suggests that core circuit characteristics can be effectively captured through aggregate feature representations rather than requiring sophisticated sequential modelling. This finding has practical implications for deployment scenarios with limited computational resources or where interpretability is valued.

The most significant architectural insight emerged from the LSTM's relative underperformance. Despite LSTM's reputation for capturing long-term temporal dependencies, our unidirectional LSTM achieved the lowest performance among all tested ML models (95.94% accuracy, ROC AUC 0.9725), significantly lagging behind both the best-performing 1D CNN model (by 3.21 percentage points) as well as the simple baseline models. While the BLSTM improved performance slightly (ROC AUC 0.9904 vs. 0.9725), it still fell short of approaches optimised for local pattern detection. One plausible explanation is that circuit patterns are better characterised by local temporal motifs, such as the transition from the *downwind leg* to the *base leg*, or the cyclical return to runway proximity, rather than

by long-range sequential dependencies. Convolutional layers capture these local patterns through sliding windows, whereas LSTM gating mechanisms, designed for long-term memory, may be over-engineered for this particular temporal structure. This finding resonates with literature showing context-dependent LSTM effectiveness in aviation applications [4], where methods focusing on local kinematic patterns often outperform those emphasising long-range temporal coherence. Notably, Olive *et al.* [20] successfully employed CNNs to detect the characteristic loop pattern of holding manoeuvres, reinforcing that convolutional approaches are well suited for spatially confined, repetitive flight patterns.

Examination of misclassified cases provides further insight into these architectural differences. False negatives predominantly occurred when ADS-B trajectories were severely incomplete due to signal dropouts, missing critical circuit legs such as the base or final approach, or when circuits exhibited highly irregular geometries deviating substantially from standard patterns due to extreme wind conditions or non-standard entries. False positives arose primarily in characteristic scenarios at ground-level segments where aircraft were taxiing or completing landing roll-out. In this case, the LSTM exhibited systematic failures by misinterpreting flat altitude profiles at field elevation (e.g., sustained altitudes of 1550 ft at LSZT with <5 ft variation) and proximity to the runway (<0.2 NM) as the final approach phase of a circuit, particularly when preceded by descent segments. The sequential context led the LSTM to hallucinate circuit completion despite the aircraft remaining on the ground throughout the segment. The 1D CNN correctly classified these cases by relying on local geometric features, specifically, the absence of altitude variation and turning manoeuvres characteristic of actual circuit legs, rather than on sequential transitions. This qualitative analysis reinforces that geometric consistency and local pattern regularity, rather than global temporal coherence, serve as the primary discriminators for circuit classification.

The end-to-end pipeline validation over the three-month period at Lomms Airfield revealed both strengths and fundamental limitations. The overall detection rate of 67.6% (438 out of 648 flights) must be contextualised within ADS-B data availability constraints. Analysis revealed that the vast majority of the 210 undetected flights corresponded to aircraft whose trajectories were not retrieved by the OpenSky Network, indicating absent ADS-B transponders or signal coverage gaps due to missing ground receivers and/or line-of-sight issues. This finding is consistent with known limitations of crowd-sourced ADS-B data [31, 13, 14, 15], where coverage gaps and equipment variability produce inevitable data incompleteness. Some temporal discrepancies with ground truth records also arise from differences in airfield operator annotation methods, for instance, whether operators timestamp the moment an aircraft first touches down versus when it fully stops, though these were not the main focus of this study. These limitations are not specific to our approach but represent fundamental constraints of the input data source. Nevertheless, for the subset of flights adequately captured by ADS-B, the pipeline demonstrated high accuracy, correctly identifying the vast majority of circuit operations (i.e., 99.1% on average over the entire observation period).

A second important limitation concerns the dependency of classification accuracy on the quality of the trajectory segmentation. The polygon-based overflight segmentation method depends critically on the runway capture area being appropriately sized through the scale factor. If segmentation fails to properly capture a circuit's boundaries, for instance, when an aircraft disconnects its transponder or loses signal before reaching the runway, the subsequent classification may be compromised. We observed cases where the ML model could still identify that a circuit pattern existed within a wrongly segmented candidate, but the temporal boundaries were inaccurate, leading to mischaracterised traffic circuits or incorrectly assigned timings. This limitation arises from the manual labelling process, where low-quality or incomplete aerodrome movement candidates were excluded from the training set, representing a 5.46% of the original dataset of 3297 labelled segments, potentially biasing the model toward cleaner geometric patterns and creating a performance ceiling for candidates with

poor initial segmentation.

Despite these limitations, we observed promising evidence of generalisability. Our ML models successfully identified traffic circuits at different airports beyond Lommiss, and even correctly recognised IFR go-around manoeuvres in datasets from traffic library beyond the training domain, indicating that the learned representations capture fundamental circuit dynamics rather than location-specific patterns. The validated pipeline demonstrates strong potential for automated aerodrome movement monitoring at non-towered airfields, substantially reducing manual labour and human error inherent in current regulatory reporting practices. However, deployment must acknowledge fundamental data availability constraints; for scenarios requiring complete movement records, supplementary data sources or hardware upgrades may be necessary to close coverage gaps that currently limit detection to approximately two-thirds of all operations.

5. Conclusion and Outlook

This study addressed the question of whether and how a structured record of aerodrome movements for a small, non-towered airfield can be created solely from ADS-B trajectory data. Such a record includes timestamps of key flight events, such as take-off and landing, as well as information on traffic circuits, operation type, aircraft identifier, runway usage, and route origin or destination. We developed a methodological framework that systematically transforms raw ADS-B surveillance data into a structured record of aerodrome movements. This framework integrates preprocessing, trajectory segmentation, and event detection components to identify relevant flight phases and to detect and classify aerodrome traffic circuits. For traffic circuit detection, we compared classical ML classifiers, including Logistic Regression and Random Forest, with sequence-based architectures such as 1D CNNs and LSTMs to evaluate their relative performance in terms of accuracy and robustness.

Our results indicate that ADS-B data can be effectively used to generate structured records of aerodrome movements. The detection of take-offs, landings, and flight phases based on ADS-B trajectory data performs reliably for GA aircraft, confirming previous findings in the literature. The detection and classification of aerodrome traffic circuit patterns represent a novel contribution of this study. Our best-performing ML model, a 1D CNN, achieved an accuracy of 99.15%, demonstrating that aerodrome traffic circuits can be reliably detected automatically. Notably, simpler models such as Logistic Regression (98.08% accuracy) and Random Forest (97.86% accuracy) also achieved exceptional performance using engineered statistical features, suggesting that domain knowledge in feature design often outperforms architectural complexity. These findings depend on sufficient ADS-B data quality, which can be improved through strategically placing receivers near airports as well as through wider installation and consistent in-flight use of ADS-B transmitters in GA aircraft.

This study demonstrated that structured records of aerodrome movements no longer need to be compiled manually but can instead be generated using data-driven and automated methods. This is particularly relevant for small airfields with limited personnel resources. Using our approach, a preliminary version of the structured report can be automatically produced and subsequently reviewed by a human operator, who may complement it with additional information that cannot be inferred from ADS-B data, such as the number of passengers on board. The proposed solution is scalable and can be applied to airports of any size, ranging from small, non-towered airfields to large international hubs. In this way, the presented methods contribute to the ongoing transition toward more data-driven airport operations, complementing recent efforts on ADS-B trajectory based data-driven airport management [8] and A-CDM [9].

While our pipeline achieves 99.1% detection accuracy when ADS-B data is available, the primary operational limitation remains the 67.6% overall coverage rate due to missing ADS-B trajectories, which motivates future work on multi-source data fusion strategies or strategic receiver deploy-

ment to close coverage gaps. The finding that simple statistical feature models (Logistic Regression, Random Forest) achieve performance comparable to complex deep learning architectures, while offering superior computational efficiency and interpretability, highlights opportunities for real-time deployment and for hybrid architectures that combine statistical efficiency with advanced pattern-recognition capabilities. Evidence of cross-airport and cross-manoeuve generalisability, such as the successful identification of go-arounds flown by aircraft flying under instrument flight rules beyond the training domain, provides a foundation for extending the pipeline to other aerodrome-specific operations, leveraging the demonstrated effectiveness of convolutional architectures in capturing local temporal patterns. Finally, the segmentation dependency identified in Section 4 highlights the potential for approaches in which segmentation and classification are jointly optimised, or for adaptive segmentation methods that offer greater robustness to variations in trajectory quality.

Acknowledgement

The authors acknowledge the contributions of three reviewers that greatly enhanced the value of this study. No potential conflict of interest was reported by the authors. No funding was received for this research.

Author contributions

- Alex Fustagueras: Conceptualization, Methodology, Data Curation, Software, Validation, Visualisation, Writing (Original Draft and Editing)
- Manuel Waltert: Writing (Editing), Project Administration

Open Data Statement

The software code used to download the OSN-data employed in this study is published on the following repository: https://github.com/alexfustagueras/Lommis_Paper

Reproducibility Statement

The software code used to generate the results presented in this paper is published on the following repository: https://github.com/alexfustagueras/Lommis_Paper

References

- [1] Mohammad Farhadmanesh, Abbas Rashidi, Paul Schonfeld, Jasenka Rakas, and Nikola Marković. “Aircraft surface movement and operation monitoring systems in general aviation and commercial airports: A state-of-the-art review”. In: *Iranian Journal of Science and Technology, Transactions of Civil Engineering* 49.1 (2025), pp. 1009–1030. DOI: 10.1007/s40996-024-01643-y.
- [2] Nicoletta Fala, Christos Falas, and Anastasios Falas. “A method for automatic airport operation counts using crowd-sourced ADS-B data”. In: *Aviation* 26 (Nov. 2022), pp. 209–216. DOI: 10.3846/aviation.2022.18025.
- [3] Transportation Research Board and National Academies of Sciences, Engineering, and Medicine. *Evaluating methods for counting aircraft operations at non-towered airports*. Ed. by Maria J. Muia and Mary E. Johnson. Washington, DC: The National Academies Press, 2015. ISBN: 978-0-309-30843-4. DOI: 10.17226/22182.
- [4] Nicoletta Fala, Georgios Georgalis, and Nastaran Arzamani. “Study on machine learning methods for general aviation flight phase identification”. In: *Journal of Aerospace Information Systems* 20.10 (2023), pp. 636–647. DOI: 10.2514/1.I011246.

- [5] Qilei Zhang, John H. Mott, Mary E. Johnson, and John A. Springer. “Development of a reliable method for general aviation flight phase identification”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.8 (2022), pp. 11729–11738. DOI: 10.1109/TITS.2021.3106774.
- [6] Federal Office of Civil Aviation (FOCA). *Richtlinie zur Datenerhebungs- und -lieferungspflicht der Flugplätze*. Version 1.0. Effective as of 1 January 2025. Bern, Switzerland, Nov. 27, 2024. URL: https://www.bazl.admin.ch/bazl/de/home/themen/geoinformation_statistik/statistik/statistische_datenlieferungen.html.
- [7] Houcai Chen, Junxiang Ge, Deqing Kong, Zhenwei Zhao, and Qinglin Zhu. “A real-time monitoring method for civil aircraft take-off and landing based on synthetic aperture microwave radiation technology”. In: *Sensors* 22.10 (2022). ISSN: 1424-8220. DOI: 10.3390/s22103675. URL: <https://www.mdpi.com/1424-8220/22/10/3675>.
- [8] Michael Schultz, Judith Rosenow, and Xavier Olive. “Data-driven airport management enabled by operational milestones derived from ADS-B messages”. In: *Journal of Air Transport Management* 99 (2022). ISSN: 0969-6997. DOI: <https://doi.org/10.1016/j.jairtraman.2021.102164>.
- [9] Michael Schultz, Judith Rosenow, and Xavier Olive. “A-CDM Lite: Situation awareness and decision-making for small airports based on ADS-B data”. In: *Proceedings of the 9th SESAR Innovation Days, 2019*. Athènes, Greece, Dec. 2019.
- [10] Xavier Olive, Junzi Sun, Adrien Lafage, and Luis Basora. “Detecting events in aircraft trajectories: Rule-based and data-driven approaches”. In: *Proceedings of the 8th OpenSky Symposium 2020*. Vol. 59. 1. Presented at the 8th OpenSky Symposium (online, 12–13 Nov 2020). MDPI, 2020, p. 8. DOI: 10.3390/proceedings2020059008.
- [11] Benoit Figuet, Raphael Monstein, Manuel Waltert, and Steven Barry. “Predicting airplane go-arounds using machine learning and open-source data”. In: *Proceedings of the 8th OpenSky Symposium 2020*. Vol. 59. 1. Presented at the 8th OpenSky Symposium (online, 12–13 November 2020). MDPI, 2020, p. 6. DOI: 10.3390/proceedings2020059006.
- [12] Benoit Figuet, Rainer Koelle, Esther Calvo Fernández, and Manuel Waltert. “Analysing the impact of go-around occurrences at large European airports”. In: *Journal of Open Aviation Science* 1.2 (2023). DOI: 10.59490/joas.2023.7219.
- [13] Manuel Waltert and Benoit Figuet. “Using ADS-B trajectories to measure how rapid exit taxiways affect airport capacity”. In: *Journal of Open Aviation Science* 1.2 (2023). DOI: 10.59490/joas.2023.7207. URL: <https://journals.open.tudelft.nl/joas/article/view/7207>.
- [14] Manuel Waltert, Benoit Figuet, and Michael Felix. “Evaluating potential fuel savings of external alternative ground propulsion systems”. In: *Journal of Open Aviation Science* 2.2 (2024). DOI: 10.59490/joas.2024.7888.
- [15] Xavier Olive, Manuel Waltert, and Michael Schultz. “Assessing airport surface traffic performance from open sources of aviation data”. In: *Proceedings of the 1st USA–Europe Air Transportation Research and Development Symposium (ATRDS2025)*. Full paper details to be added. –: USA–Europe Air Transportation Research & Development Symposium, 2025.
- [16] Raúl Torres, Pedro C. Álvarez-Esteban, and Nicolas Peña. “An algorithm to determine airport runway usage/configuration based on aircraft trajectories”. In: *Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. San Diego, CA, USA: IEEE, 2019, pp. 1–7. DOI: 10.1109/DASC43569.2019.9081678.
- [17] Xavier Olive and Jérôme Morio. “Trajectory clustering of air traffic flows around airports”. In: *Aerospace Science and Technology* 84 (2019), pp. 776–781.
- [18] Satvik G. Kumar, Samantha J. Corrado, Tejas G. Puranik, and Dimitri N. Mavris. “Classification and analysis of go-arounds in commercial aviation using ADS-B data”. In: *Aerospace* 8.10 (2021). ISSN: 2226-4310. DOI: 10.3390/aerospace8100291.
- [19] Imen Dhief, Sameer Alam, Chan Chea Mean, and Nimrod Lilith. “A tree-based machine learning model for go-around detection and prediction”. In: *Proceedings of the 11th SESAR Innova-*

- tion Days (SIDs). Paper ID 66. Virtual Event (Online), Dec. 2021. URL: https://www.sesarju.eu/sites/default/files/documents/sid/2021/papers/SIDs_2021_paper_66.pdf.
- [20] Xavier Olive, Luis Basora, Junzi Sun, and Enrico Spinielli. “Training a machine learning model to detect holding patterns in aircraft trajectories”. In: *Journal of Open Aviation Science* 2.2 (Mar. 2025).
- [21] International Civil Aviation Organization. *Procedures for air navigation services: Air traffic management*. 16th ed. ICAO Doc 4444. Montreal, Canada: ICAO, 2016.
- [22] Jay Patrikar, Brady Moon, Jean Oh, and Sebastian Scherer. “Predicting like a pilot: Dataset and method to predict socially aware aircraft trajectories in non-towered terminal airspace”. In: *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, 2022, pp. 2525–2531. DOI: 10.1109/ICRA46639.2022.9812009.
- [23] Kelton Karboviak, Sophine Clachar, Travis Desell, Mark Dusenbury, Wyatt Hedrick, James Higgins, John Walberg, and Brandon Wild. “Classifying aircraft approach type in the national general aviation flight information database”. In: *Computational Science – ICCS 2018*. Ed. by Yong Shi, Haohuan Fu, Yingjie Tian, Valeria V. Krzhizhanovskaya, Michael Harold Lees, Jack Dongarra, and Peter M. A. Sloot. Cham: Springer International Publishing, 2018, pp. 456–469. ISBN: 978-3-319-93698-7.
- [24] Junzi Sun, Joost Ellerbroek, and Jacco Hoekstra. “Flight extraction and phase identification for large automatic dependent surveillance–broadcast datasets”. In: *Journal of Aerospace Information Systems* 14.10 (2017), pp. 566–572. DOI: 10.2514/1.I010520.
- [25] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. “Bringing up OpenSky: A large-scale ADS-B sensor network for research”. In: *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (IPSN '14)*. Berlin, Germany: IEEE, 2014, pp. 83–94. DOI: 10.1109/IPSN.2014.6846743.
- [26] Xavier Olive. “traffic, a toolbox for processing and analysing air traffic data”. In: *Journal of Open Source Software* 4 (2019), p. 1518. ISSN: 2475-9066. DOI: 10.21105/joss.01518.
- [27] Pauli Virtanen et al. “SciPy 1.0: Fundamental algorithms for scientific computing in python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [28] Junzi Sun, Jacco M Hoekstra, and Joost Ellerbroek. “OpenAP: An open-source aircraft performance model for air transportation studies and simulations”. In: *Aerospace* 7.8 (2020), p. 104.
- [29] F. Pedregosa et al. “Scikit-learn: Machine learning in python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [30] Martín Abadi et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.
- [31] Zhanji Yang, Xiaolei Kang, Yuanhao Gong, and Jiansheng Wang. “Aircraft trajectory prediction and aviation safety in ADS-B failure conditions based on neural network”. In: *Scientific Reports* 13.1 (Nov. 2023), p. 19677. ISSN: 2045-2322. DOI: 10.1038/s41598-023-46914-2.