Smart Data Fusion: Probabilistic Record Linkage adapted to Merge two Trajectories from Different Sources

Dario Martinez INNAXIS research institute dm@innaxis.org Samuel Cristobal INNAXIS research institute sc@innaxis.org Seddik Belkoura INNAXIS research institute sb@innaxis.org

Abstract—Traditional merging methodologies (e.g sort-join) are not enough to match de-identified data from two (or more) different datasets. Deterministic matching only works if the records are perfect and present in all the databases linked, e.g. there is a single unique identifier (key). This condition rarely exists due to errors, typos or lack of convention, among other reasons. This paper is a first step to adapt probabilistic merging to aviation. The technique uses a wider range of potential identifiers, computing weights for each identifier based on its estimated ability to correctly identify a match or a non-match and using these weights to calculate the probability that two records on different databases actually correspond to the same entity. We adapt several techniques to two sources of trajectories, namely: radar and GPS. Results show not only the ability to link more records than rule-based sort-join strategies, but also to link the trajectories even when the key identifiers have been removed. This work paves the way for a range of applications and allows secure data merging of anonymised data in aviation.

Keywords—Probabilistic Record Linkage, CPR, ADS-B, DTW, Hausdorff distance

I. INTRODUCTION

Aviation is a novel field for application of general Artificial Intelligence (AI) methods and Machine Learning techniques (ML) in particular. Most machine learning algorithms rely heavily on available data - not just in volume, number of samples on the data set, but also in variety, number of parameters or features in the dataset. Intuitively, having a wide variety of datasets makes it easier to find the most relevant features simply because the search space is larger and therefore more likely to be relevant, provided the volume is adequate enough to ensure confidence on the models. However, due to the independence of the different systems in aviation and lack of consolidation, data that's relevant to the same event or object are often distributed among datasets. For instance, air traffic controllers rarely have access to airline operators' data and vice versa. Any machine learning model derived from each others' data in isolation will be unavoidably biased and rarely complete. This potentially limits the applicability of many AI and ML techniques in aviation.

Full disclosure of datasets could allow the identification of flights using traditional methods, e.g. sort-join exact methods. However, in practice, this approach is not possible. For instance, airline operators and ATC are reluctant to share identifiable data to respect their personnel's privacy, e.g. addressing the EU General Data Protection Regulation (GDPR). In most cases, the de-identification process consists in eliminating the references required for exact matching between datasets; for example, removing the flight date and callsign.

In this paper, we present a probabilistic methodology for merging datasets without using identifiers. The method finds the best linkage between elements of different radar sources. Contrary to exact methods, any probabilistic linkage would have false positives and negatives, which ultimately affects the machine learning models trained with them. However, the improvement on accuracy of a more varied dataset overcomes the induced errors of probabilistic linkages.

This method would enable in the future a number of potential data-driven AI applications in aviation. Datasets could be shared with the research community, de-identified, and still enable researchers to match records with enough confidence to generate datasets with richer features distributions that aid in training more robust Machine Learning algorithms. This is the principle behind DataBeacon, the multi-sided, Big Data platform for aviation. DataBeacon was born from the Horizon 2020 safety research program SafeClouds.eu in which airlines, ANSPS and airports shared data with researchers of universities and private institutions to develop data analytics and bridge the gap between Data Science and Aviation.

The aim of this paper is to match two data sources: Correlation Position Reports (CPRs) and ADS-B captured positions. We implemented, adapted and compared several merging techniques. We used, as baseline, the result of a naive inner join using date and callsign identifier. Then we shifted to probabilistic merging strategies in which we showed that matching (identified) callsigns is not as trivial as it seems. Finally, we show that by using the Hausdorff distance between trajectories, we can merge the two datasets without using any identifier other than the premise that the flights occurred on the same de-identified day. This premise is only necessary because flights may repeat routes at the same time in different dates. We show that Probabilistic Record Linkage using trajectory distances may be the ultimate technique that solves the problem of matching de-identified flight trajectory data.

This paper is organised as follows: in section II, we gather past research on probabilistic record linkage, and discuss why





probabilistic record linkage methodologies provide a good solution for the flight matching problem. Then, we will review some well-used distance metrics for measuring trajectories similarities. This kind of metrics are often used in clustering problems, but we will innovate by incorporating them as a cost function for a linkage algorithm. In sections III and IV, we present the datasets and detail the methodology followed, with the explanation of some considerations taken during the implementation. Section V presents the results and compares different linkage methodologies. Finally, we summarize and identify some ideas for future work and potential applications.

II. STATE OF THE ART

A. Deterministic (rule based) vs probabilistic record linkage

Record linkage is a solution to the problem of recognizing records in two files which represent identical persons, objects, or events [2]. Deterministic (rule based) linkage is the simplest method of matching as it consists of a sort-merge operation that finds an exact match [4]. It works best with a single unique identifier (key) and when the identifiers have equal importance. Rule based linkage works well if the keys do not contain errors and are always present in all datasets to link.

A new approach to record linkage was introduced by Newcombe and Kennedy (1962) [1] and formally presented by Fellegi and Sunter (1969) [2]: probabilistic record linkage. Also known as fuzzy matching, this approach allows a wider range of potential identifiers by computing weights for each identifier based on its estimated ability to correctly identify a match or a non-match. Then these weights are used to calculate the probability that two given records refer to the same entity.

While deterministic record linkage requires a series of potentially complex rules (e.g. SQL queries) to be programmed ahead of time, probabilistic record linkage methods can be "trained" to perform well with little human intervention [3].

B. Review on probabilistic record linkage

The principal idea of probabilistic linkage is that two probabilities are estimated: the M-probability as the prospect that a field agrees given that the pair of records is a true match; and the U-probability as the prospect that a field agrees given that the pair of records is not a true match. Following the definition of Felligni and Sunter [2], given two sets of records denoted as A and B, with each record belonging to each set as $a \in A$ and $b \in B$. The available information regarding one record is denoted with $\alpha(a)$ and $\beta(b)$. The comparison set $A \times B$ is partitioned into two subsets $M = \{(a, b) \in A \times B | a = b\}$ of matching pairs and $U = \{(a, b) \in A \times B | a \neq b\}$ of unmatching pairs [2].

Then, two datasets are compared by means of a comparison vector γ which is a vector function of the record pairs. From a probabilistic point of view, is an event, and as a result a conditional probability can be attached to the vector as $m(\gamma) = P(\gamma|(a,b) \in M) = P(\gamma|M)$ and $u(\gamma) = P(\gamma|(a,b) \in U) = P(\gamma|U)$, that are respectively, the probability of observing the event given a match and giving a non-match. The algorithm labels record pairs as A_1 , if they match, A_2 , if they do not match, and A_3 if they possibly match. Hence, a linkage rule is a decision function $d(\gamma) = \{P(A_1|\gamma), P(A_3|\gamma), P(A_2|\gamma)\}$ and it is such that $P(A_1|\gamma) + P(A_3|\gamma) + P(A_2|\gamma) = 1$. Finally, the decision rule is defined as $R = \frac{m(\gamma)}{u(\gamma)}$, then when $R \ge t_l$ a match is found and $R \le t_u$ when a non-match is found, being t_l and t_u thresholds to be set.

Modern improvements upon the classical probability linkage methodology include the application of the expectationmaximisation (EM) algorithm for parameter estimation [18], the use of approximate string comparisons to calculate partial agreement weights when string based values are expected to contain typographical errors [7], and the application of Bayesian Networks [14]. In recent years, researchers have started to explore Machine Learning (ML) based approaches, such as supervised learning based on a training dataset with known linkage [13]. Also some interesting studies that borrow ideas from the Natural Language Processing (NLP) field suggest representing records as document vectors and computing the cosine distance between them [12]. Although ML-based approaches seem very promising, this paper will not focus on them, mainly due to the lack of validated linkage techniques for radar data that are required to conform a reliable training dataset.

Probabilistic linkage is heavily limited by dimensionality of the data, with d databases of n records each, since brute force approaches, using all-to-all comparisons, require comparisons $O(n^d)$ [18]. Blocking methodologies attempt to restrict comparisons by reducing the number of records using one or more discriminating identifiers [9]. Since blocking strategies can influence linkage success, Christen and Goiser recommended that researchers report the specific steps of their blocking strategy [17]. While simple blocking strategies [9] compares all the pairs that "hash" to the same value, there are more advanced methodologies that involve unsupervised clustering, e.g. form clusters around certain key values (canopies method) [18]. An alternative to standard blocking and clustering is the sorted neighbourhood approach [11], where records are sorted by a blocking variable before a sliding window is moved over the sorted set, enabling comparisons between the records within the window.

Probabilistic linkage has been successful for matching records using string-based identifiers such as names, addresses or postal codes. As showed by Jaro (1989) [10], the problem can be approached through string matching with typographical errors as a cost-based linkage problem. Winkler (1993) [7] formally defined and extended this similarity metric as an edit-distance that uses an EM algorithm to estimate the parameters. This is known as the Jaro-Winkler distance.

Taking into account the maturity of the record linkage research, we believe that a cost-based methodology using trajectories similarities can be implemented in order to solve the problem of flight matching with radar data.





C. Review on trajectory distance measures

Euclidean distance, Manhattan distance and other L^p -norms are the most used metrics for measuring distance between points in two time series. They compare discrete series and can only be used if the two time series are of equal length, or if some length normalisation technique is applied. In the context of comparing radar data, these distances are normally used to compare sub-trajectories of fixed length and can't be used to compare entire trajectories.

Warping distances try to overcome the L^p -norm limitations. They are especially designed to compare locations from different trajectories and different indices. The main idea is to find the optimal alignment between two trajectories by using a given cost between a matched location. The most used algorithm is Dynamic Time Warping (DTW) [16]. DTW is a time-normalisation algorithm initially designed to eliminate timing differences between two time series. It does not require the two series to be of the same length, and allows for time shifting between the two time series by repeating elements.The DTW normalisation is done by warping the time axis of one time series to match the other, an example of this technique can be appreciated in figure 1.



Figure 1. DTW distance calculation

DTW relies on dynamic programming and requires each element of one time series be compared with each element of the other; this evaluation is slow and has a minimum computational complexity of $O(n^2)$. However, recent studies propose FastDTW [20] as an alternative to the cost-intensive standard DTW algorithm. The authors claim that FastDTW yields a computational complexity of O(n).

There are important limitations on the applicability of warping methodologies. They are based on one-to-one comparisons between sequences meaning that sequences need to be balanced (e.g. similar number of points) in order to correctly capture the similarity between trajectories. Another limitation is that these methodologies do not perform correctly when a large amount of noise and outliers are present. Normally, the solution to these problems is to correct the time index or have the effect of time removed from the trajectory study. These limitations are very influential in applying the metrics to aviation-related trajectories.

As an alternative to wrapping distances, we meet shapebased distances. The Hausdorff distance [21] is a well-known metric for expressing the spatial similarity between two curves. Informally, two sets are close in the Hausdorff distance if every

TABLE I CPR dataset description

Field	Description	Sample
2	TACT: unique flight identifier. (De-Identified).	ec16fcae3ac614385d73412f89a03114
9	Callsign.	XGV410
10	Departure airport.	EBBR
11	Destination airport.	LFSB
time	Time of the samples formated as seconds per day. Extracted from field 4 (timestamp) before de-identification.	[17082, 17099,]
lat	Latitude point per time sample. Extracted from field 13 (position).	[50.903055555555554,]
lon	Longitude point per time sample. Extracted from field 13 (position).	[4.4713888888888889,]
14	Flight level (FL).	[4, 9, 16, 16,]

TABLE II ADS-B DATASET DESCRIPTION

Field	Description	Sample
id	Unique flight identifier. (De-Identified).	ea5b899291d9b19bff76a9cb2d1b2744
aircraft_id	Callsign extracted from the id before the de-identification.	XGV041
time	Time of the samples formated as seconds per day.	[17096, 17115,]
lat	Latitude point per time sample.	[50.89705, 50.89243,]
lon	Longitude point per time sample.	[4.45221, 4.43714,]
alt	Altitude in ft.	[900, 1600, 2200,]

point of either set is close to some point of the other set. The Hausdorff distance between two sets of metric spaces is mathematically defined as

$$d_H(A,B) = \max\left\{\sup_{a\in A} \inf_{b\in B} ||ab||_2, \sup_{b\in B} \inf_{a\in A} ||ab||_2\right\}$$

This formula is roughly translated to the largest distance from any point in one of the sets, to the closest point in the other set, and can be computed in a computational time. However, the main problem with Hausdorff distance is that the largest distance in the set might be an outlier and not representative of the "real" trajectory. Furthermore, data imprecision is a phenomenon that has existed as long as data is being collected. In practice, data is often sensed from the real world and as a result has a certain error region, making the Hausdorff algorithm unreliable for real-world trajectories. Because of this, in our case, it makes more sense to use the Hausdorff distance as a measure of dissimilarity between two point sets.

Recent research suggests using early break and random sampling techniques to reduce the complexity of the algorithm that computes the Hausdorff distance between two trajectories [22]. The best case performance is O(n), which is satisfied by selecting an inner loop distance that leads to an early break as often as possible. The authors have formally shown that the average runtime is also closer to linear complexity.

III. DATASETS

The datasets to be matched are Correlated Position Report (CPR) and Automatic Dependent Surveillance-Broadcast





(ADS-B) data. The scope of the study has been limited to one day of flights over the ECAC (European Civil Aviation Conference) area. Both datasets have been de-identified in order preserve privacy. Dates and the unique identifiers (e.g. flight id, TACT, etc.) have been de-identified by hashing the string values using the same format. For the sake of experimentation and validation, the callsigns have been left as plaintext. Production environments would also expect to work with them as hashed fields. The size of the data is considerable. For one day, CPR presents 32.484 flights and ADS-B 32.673 flights.

Regarding the quality of the data, and due to the nature of the source, CPR is expected to be cleaner than ADS-B. As ADS-B is a broadcast signal captured over different receptors, it is expected to present missing segments and considerable inconsistencies on the sampling. One of the advantages of ADS-B, as opposed to CPR, is that its position determination can be more accurate. Also, it is important to remark that position samples will be irregularly distributed along the trajectory.

We present a brief description of the datasets in tables I and II.

IV. METHODOLOGY



Figure 2. Linkage methodology

A. Cleaning and Standarization

The main purpose of data cleaning and standardisation is the conversion of raw input radar data into well-defined signals, eliminating all the possible inconsistencies in the way the data is represented or encoded.

We define a trajectory T_r as a finite sequence of geo-locations with timestamps, i.e., $T_r = (p_1, t_1), (p_2, t_2), ..., (p_n, t_n)$ with $t_i < t_{i+1}$ for i = 1, 2, ..., n1. Being p_i a sampling point observed at time t_i . In this paper, each sampling point is represented by a pair (x, y), denoting longitude and latitude respectively.

When working with coordinates, normalization is recommended so that the distance between two time series is invariant to amplitude scaling and (global) shifting of the time series. In this paper, all data is normalized as follows: for T_r of length n, let the mean of the data in dimension d be μ_d and let the standard deviation be σ_d . Then, to obtain the normalized data $N(T_r)$, we can evaluate $\forall i \in t_{i,d} = \frac{t_{i,d} - \mu_d}{\sigma_d}$ on all elements of T_r . This process is repeated for all dimensions. When working with aviation trajectories it is very common to apply complex interpolations such as the Piecewise Cubic Hermite Interpolating Polynomial (PCHI) [25]. In this paper, we do not perform any interpolation because the distance metrics that are applied do not require the same number of points and they are well. Another common pre-processing step for aviation data is to use the Ramer-Douglas-Peucker (RDP) [23] algorithm to remove redundant trajectory information [25]. We noted no benefit of following this methodology, and it was not relevant in the merging results.

B. Blocking

Without using a blocking technique, when performing a record linkage between ADS-B and CPR data, the number of comparisons equals the product of the number of flights in the two datasets $|A| \times |B|$. For only one day of flights, CPR presents 32.484 flights and ADS-B 32.673 flights, therefore making more than 1 billion possible combinations, a number computationally infeasible.

The aim of the blocking methodology is to cheaply remove as many record pairs from the subset of non-matches $U = \{(a, b) \in A \times B | a \neq b\}$ that are obvious non-matches without removing any records pairs from the subset of matches $M = \{(a, b) \in A \times B | a = b\}$. Due to the nature of our datasets, applying a sort-based blocking technique makes sense, as radar data is naturally indexed by the time dimension.

Sorted neighbourhood is one of the most common indexing methodologies when working with sortable records. It follows the idea of sorting tuples in a way in which similar entries are close to each other, enabling comparisons of tuples within a small window (neighbourhood).

The method consists of three steps. First, a key for each record is created. For our problem, we select the time of the first point registered. Note that the choice of this key is not as trivial as it seems, because the samples from two different radar sources that correspond to the same flight may not be registered at the same time or the case might be that the beginning of the flight is missing. Further techniques can be applied for inferring the real starting time of the flight (e.g. filtering by flight level and interpolating), but they are out of the scope of this research. Next, the data is sorted based on the selected key - in our case, the flights will be sorted in time. Finally, a fixed size window is moved through the sequential list of records in order to limit the comparisons.

In order to validate the blocking technique, there are two complexity measures useful for quantifying the efficiency and quality of the blocking strategy [24]:

- The reduction ratio is defined as $rr = 1 \frac{N_b}{|A| \times |B|}$ with $N_b \leq |A| \times |B|$ being the number of record pairs not removed by blocking. This is a metric of the relative reduction of the comparison space, but without taking into account the quality of the reduction.
- The pairs completeness is measured as $pc = \frac{N_m}{|M|}$ with N_m the total number of correctly classified true matched record pairs within the window and |M| the total number of matches.





Normally there exist a trade-off between these two metrics, similar to the precision-recall trade-off in classification problems. Usually, it is desirable to achieve the highest reduction ratio at the highest pairs completeness [24]. To test influence of this trade-off in our blocking methodology, we ran the linkage algorithm for different sizes of the window.

In windows from 5 minutes to 2 hours, the rr was constant at 0.99. the pc raised from 0.92 to 0.95 for window sizes of 5 minutes to 1 hour (testing larger windows is difficult, manual review is mandatory to count true positives). Therefore, the trade-off is irrelevant in this case study and the only limitation to the window size is the computational time desired for the algorithm.

The length of the window has been fixed to 2 hours in order to capture all the relevant flights with a reasonable computational time.

C. Cost-based probabilistic linkage

To perform the linkage, we will follow the standard layout for generic probabilistic record linkage algorithms [26]:

1) Cost function estimation: The estimator should be able to measure trajectories with similar shape and physically close to each other. Also, the distance function should be able to differentiate curves that are similar as a whole with more than just similar sub-parts. Note that, because of the previous blocking step, time indexing should not be an issue. We will approach our linkage probability as a "cost based model". Instead of directly using directly probability of matching to each record, we assign a misclassification cost to each pair of trajectories. This means that we can use a similar metric between 2-dimensional series to measure the difference between the trajectories as a cost that should minimised.

As we have presented in the state-of-the-art, there are two main methodologies to measuring similarity: warping or shape-based. Warping methods, such as DTW, are based on one-to-one comparisons between sequences. Hence, it often requires the choice of a particular series as a reference, onto which all other sequences will be matched. The indexes of two sequences that are compared should be well-balanced in order to best capture the variability. For instance, to detect if there are accelerations and decelerations during the measurement of the time series, making the choice of the reference sequence is very relevant. Also, the computation of all these comparisons, even after the sorted neighbourhood, can be too computationally demanding.

For this particular case, we will show in further sections that the Hausdorff distance is the best metric. However, for the sake of experimentation, we will implement the linkage algorithm using both the DTW distance and the Hausdorff distance.

2) Weight computation: As stated, a distance will be the cost-based estimator used to calculate statistics associated to three pairs of 2-dimensional series: Longitude-Latitude, Time-Latitude and Time-Longitude. Another approach would have been to directly use the 3-D trajectory or even a 4-D trajectory including altitude/FL. The problem with this approach is that

the shape-based distance is better measured in a plane. Also for the 4-dimensional case we find different units (altitude in feet and FL) between the two sources, making an approximated unit conversion necessary.

We define the test statistic as $w(\gamma) = w_1 + w_2 + w_3$, which represent the weights used to define the cost of misclassification. Given the CPR flight $a \in A$, the of ADS-B flight $b \in B$ and a similarity metric d_m , each weight is defined as

$$w_1(a,b) = d_m(a(lon, lat), b(lon, lat))$$
$$w_2(a,b) = d_m(a(t, lat), b(t, lat))$$
$$w_3(a,b) = d_m(a(t, lon), b(t, lon))$$

with the two possible similarity metrics being the Hausdorff distance d_H and the DTW distance d_{DTW} .

3) Weight aggregation: A composite for the misclassification cost of the flights pair (a, b) is computed using an aggregation function which takes all the weights as input:

$$c_{ab} = i_1 \cdot w_1(a, b) + i_2 \cdot w_2(a, b) + i_3 \cdot w_3(a, b)$$
$$i_1 + i_2 + i_3 = 1$$

The distribution of weights is purely empirical, and for the results presented in this paper, we have defined the importances as $i_1 = 0.7$, $i_2 = 0.15$ and $i_3 = 0.15$. The reasoning behind this is that because we are already forcing a time constraint with the sorted neighbourhood methodology, most of the flights will be close "time-wise". Therefore, the contribution of the misclassification cost of the purely spatial series is considerably more important than the contribution of the spatial-temporal series. We cannot leave out the time dimension as some flights may be present the same trajectory but have a gap in time. However, the distributions of the importances vector is not fixed and in future work the impact of the weight can be studied further.

4) Threshold selection: Each record pair is classified into either set M or U according to the value of the score and the threshold level chosen. Note that we have not considered the Following the definition of Fellini et al., the threshold is $R = \frac{P((a,b) \in A \times B|M)}{P((a,b) \in A \times B|U)}.$

In our case, the threshold could be estimated by minimising the probability of making an incorrect decision when deciding a marching status of a flight pair. But in practice, a simple minimisation of the probability of error is not the best criterion to use in designing a decision rule due to different wrong decisions that may have different origins and consequences. For example, because not every flight of a dataset is included in the other dataset.

As suggested in [8], misclassification costs can be stored into a cost matrix C. Then, given $n \times m$ matching candidates after the blocking strategy, each element of $C c_{ij}$ represents the cost of matching or un-matching $j \in \{U, M\}$ the flight pair (a_i, b_j) . In this model, each cost value is two-fold, the first part takes into account the cost of the decision itself and the other part takes into account the cost of the consequences





in making such decision. Finally, we can calculate the average cost as

 $\bar{c} = c_{(a_1,b_1),M} P((a_1,b_1)|M) + c_{(a_1,b_1),U} P((a_1,b_1)|U) + c_{(a_1,b_1),U} P((a_1,b_1)|U) + c_{(a_1,b_1),M} P((a_1,b$ +...+ $c_{(a_n,b_m),M}P((a_n,b_m)|M) + c_{(a_n,b_m),U}P((a_n,b_m)|U)$ Mathematically, this is correct approach for estimating the threshold in our case, but the main issue with this cost is that we need to know prior probabilities $\pi_M = P(M)$ and $\pi_U = P(U)$, this are the probabilities that trajectories match/not-match given the distances between them. This estimation could be done using a baseline dataset of trusted known matches and performing a EM approach. Because obtaining a reasonable set of trusted match is unfeasible, we decided to estimate an approximate threshold empirically. For estimating the threshold, we used a sorted neighbourhood with a window of 2 hours of flights in which we calculated the cost for all of the possible pairs of trajectories and plotted the distribution of costs. Using the distance distributions for matched and unmatched data, we could graphically estimate a possible decision threshold. In figure 3 we present the threshold definition for the Hausdorff case.



Figure 3. Threshold definition for the Hausdorff distance

The difference between the distribution matched trajectories and non-matches trajectories based on distance is clear. The threshold can be set as $R : d_H(a,b) < 0.085$ for the Hausdorff distance. Using the same methodology for the DTW distance, $R : d_{DTW}(a,b) < 50000$. These values are not final, and they just represent sufficiently reliable thresholds so experimentation can be done.

In further research, this approach could be improved by proposing methods for estimating the missing prior probabilities. For now, and because of the lacking of a more reliable methodology for estimating the thresholds in a formal way, we will use these values as sufficiently reliable thresholds so experimentation can be done.

V. RESULTS AND VALIDATION

A. Comparison between different strategies

We will use, as baseline, a simple sort-join using the PySpark's "spark.sql" module. The identifiers selected for the rules are date and callsign. Note that several flights during a day can present the same callsign. Theoretically, this problem should be solved by the sorted neighbourhood since two flights

TABLE III Comparison between linkage methodologies

Methodology / Classifier	Match	
	CPR	ADS-B
Naive Inner Join using callsign as identifier.	39.5%	41.5%
Probabilistic record linkage using the Jaro- Winkler distance for callsigns.	44.3%	46.3%
Cost-Based probabilistic record linkage: DTW for temporal series.	57.3%	59.3%
Cost-Based probabilistic record linkage: Haussdorff distance for temporal series.	63.7%	66.7%

with the same callsign would not fly at the same time. But this strategy presents obvious flaws for these datasets due to callsigns containing errors - e.g. the CPR flight "DLH74" expressed in ADS-B as "DLH074". Also, note that if callsign de-identification is mandatory, there is no virtual form of validation, as you don't have any reference for the "right" matches.

To improve the baseline, and for the sake of trying out a non distance-based linkage method, we used the Jaro-Winkler algorithm to identify those flights with "similar" callsigns. The matched percentage improved over a 5% though some obvious problems were noticed. For example, callsigns may present similar codes that make the algorithm produce similar scores, e.g. "DLH423"-"DLH243" present the same distance as "DLH74"-"DLH074". Obviously, callsigns for the same flight may not be the same, e.g. IATA-ICAO codes mismatching or de-identified data.

Finally, we compare the two of the most popular strategies of measuring trajectories similarity: DTW and Hausdorff distance. Showing that, due to the nature of the aviation flight rules, shape-based distances are the superior metric in this case. The main issue with DTW is that different trajectories may present the same "signal wrapping" over time but be parallel in latitude-longitude, even after applying the right weight scoring. In both cases, setting a threshold is not trivial. We used the Python Scipy implementation [27] for calculating directed Hausdorff distances and the FastDTW [20] library for computing DTW distances.

In table III, we present the matched percentage per strategy. By looking at the results, we can state that, the Hausdorff distance is the best performing algorithm. Additionally, when computing the percentage of matched data that were also matched by the sort-join linkage using the callsign as key, it scored an outstanding 79%. This means that only 21% of the total flights are missed by the algorithm. Also, we can use this metric as validation for the performance of the record linkage, taking into account that flights with the same callsign within a day are often a record match.

B. False negative cases

False negative cases occur when two trajectories classify as non-matches but actually correspond to the same flight. This cases are normally linked with missing data and errors in ADS-B. These quality changes are normally linked with range and frequency congestion of the antenna.







Figure 4. False negative pair of flights



Figure 5. False positive pair of flights

In figure 4, a very common case of false negative is presented, the Hausdorff misclassification cost for this case is 1.53, which is actually 18 times above the threshold. But, we know that the flights match due to callsign and time inspection. The distance between the trajectories is actually significant because half of the trajectory of ADS-B data is missing, making the distance of the last points of CPR contribute to a high cost of misclassification.

The problem with this case is that there is no methodology for determining if the trajectory has actually ended because the flight may have arrived at the destination. We can't verify this because the ETA and ATA is missing from the dataset. The only solution for these cases would be to match them using a deterministic linkage with the callsign as identifier. This solution is sometimes not feasible due to callsigns requiring de-identification.

C. False positive cases

This case is harder than the detection of false negatives because the same flight callsigns may not match and the recordings might experience some latency. This means that flights require manual verification using flight tracking websites to make the verification based on additional information such as departure/arrival airport and departure/arrival time for both flights.

In figure 5, we can appreciate an interesting example of wrong match. The trajectories correspond to two flights that perform the same exact route between the same airports, but differ almost four minutes in time. They are operated by two different airlines that do not have partnership agreements. This means some flights can present the same spatial trajectory with very similar time-spatial trajectories. As we exposed in section IV, we distributed the weights in a way that purely spatial series have more influence when deciding the cost of misclassification as we didn't want radar delays to be a problem when determining the matches. Although this case is rare, it is presents a system flaw and the cost function requires attention and further testing.

VI. CONCLUSIONS AND FURTHER WORK

This paper explains and applies established Probabilistic Record Linkage (PRL) algorithms to match trajectories from two de-identified sources. We introduce a brand new methodology that uses trajectory similarities as a cost function. We have introduced a well-known blocking methodology based on a rolling window that enables a tractable computational complexity, even in Big Data environments. This means that the methodology is ready to be applied in production environments and using real-time batches.

As shown in the results section, the Probabilistic Record Linkage technique is promising when dealing with anonymised datasets, but it applicability goes beyond simply matching. It can help to merge not standardised heterogeneous datasets, eg. "DLH74" vs "DLH074". We achieved almost 80% in relation with a rule-based linkage. By using the Hausdorff distance as cost function, we achieve more than 65% of matching for one day of radar data, which is 20% more than matching achieved by deterministic linkage almost 50% relative increase. DTW distances didn't perform well enough due to problems with parallel trajectories. We also learned that trying to solve the problem by matching similar callsigns using an edit-distance is not a good solution.





Because the algorithm does not impose a format for the samples or sampling restrictions any radar source can be used, enabling unlimited possibilities for generalization: different radar sourced, different regions or even more . Even when working with low quality data, as long as the trajectories are well defined, the algorithm could be generalized.

As future work, supervised machine learning (ML) techniques could be applied to estimate the prior probabilities defined by Fellegi and Sunter. For this particular methodology, a ML-based approach would be perfect to estimate the threshold of decision, as a binary classification problem. The main issue is in acquiring or creating a matched dataset with verified true positives.

Still the techniques need further work in two directions: optimization and generalization. The algorithms are still running slow, due to the lack of optimized frameworks. Second generalization to arbitrary aviation datasets although possible will require specific work to adapt the techniques.

In any case their applicability is unquestioned. Many Multi-Party data Processing (MSP) platforms are been developed for aviation, all of them face the same challenge: in most cases data is too sensitive to be shared without performing prior de-identification, such as crew and passenger information, airlines procedures and, obviously, radar sets. Despite of the restrictive limitations, the data owners may be interested to share and even merge their datasets for their own profit. This is already a present problem in the industry when implementing applications such as cross-platform predictive analytics or blind-benchmarking. Therefore, to promote sharing data between stakeholders for its own benefits, we need ensure safety standards when managing and processing the data.

DataBeacon, the MSP platform developed under the Horizon 2020 safety research program SafeClouds.eu would be the first MSP to implement PRL techniques to fuse data from airlines, ANSPs and airports into a Smart Fused Dataframe (SDF). The SDF represents a standardised, secure data structure ready to be consumed by AI and ML models, or even full-stack applications.

ACKNOWLEDGMENT

This work has been funded by the EC project num. 724100 Safeclous.eu "Data-driven research addressing aviation safety intelligence" the authors wants to thank you all partners of the Safeclouds.eu consortium and in particular EUROCONTROL for making rada data available and OpenSky Network for sharing the ADS-B data.

REFERENCES

- Howard B. Newcombe and James M. Kennedy. *Record linkage: making maximum use of the discriminating power of identifying in-formation*. Commun. ACM 5, 11 (November 1962), 563-566. DOI: https://doi.org/10.1145/368996.369026
- [2] Fellegi, I.P. and Sunter, A.B., "A theory for record linkage", J Am Stat Assoc.1969;64(328):1183210.
- [3] Zhou, Yun, John Howroyd, Sebastian Danicic and J. Mark Bishop. Extending Naive Bayes Classifier with Hierarchy Feature Level Information for Record Linkage." AMBN@JSAI-isAI (2015).

- [4] Dusetzina SB, Tyree S, Meyer AM, et al. "Linking Data for Health Services Research: A Framework and Instructional Guide [Internet]." Rockville (MD): Agency for Healthcare Research and Quality (US); 2014 Sep. 4, An Overview of Record Linkage Methods. Available from: https://www.ncbi.nlm.nih.gov/books/NBK253312/
- [5] Andrea Tancredi, Brunero Liseo, A hierarchical Bayesian approach to record linkage and population size problems, Published in at http://dx.doi.org/10.1214/10-AOAS447 the Annals of Applied Statistics (http://www.imstat.org/aoas/) by the Institute of Mathematical Statistics (http://www.imstat.org); Annals of Applied Statistics 2011, Vol. 5, No. 2B, 1553-1585; doi:10.1214/10-AOAS447
- [6] S. N. Minton, C. Nanjo, C. A. Knoblock, M. Michalowski and M. Michelson, "A heterogeneous field matching method for record linkage," Fifth IEEE International Conference on Data Mining (ICDM'05), Houston, TX, 2005, pp. 8 pp.-.
- [7] Winkler, W. E. (1990). , "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage". Proceedings of the Section on Survey Research Methods. American Statistical Association: 354359 doi: 10.1109/ICDM.2005.7
- [8] V. S. Verykios, G. V. Moustakides, and M. G. Elfeky. A bayesian decision model for cost optimal record matching. The VLDB Journal, 12(1):2840, 2003. ISSN 1066-8888. doi: http://dx.doi.org/10.1007/s00778-002-0072v
- [9] Blakely, Tony; Salmond, Clare (December 2002). "Probabilistic record linkage and a method to calculate the positive predictive value". International Journal of Epidemiology. 31 (6): 12461252. doi:10.1093/ije/31.6.1246. PMID 12540730.
- [10] Jaro, M. A. "Advances in record linkage methodology as applied to the 1985 census of Tampa Florida". Journal of the American Statistical Association. 84 (406): 41420. doi:10.1080/01621459.1989.10478785 (1989)
- [11] T. Papenbrock, A. Heise, and F. Naumann, Progressive Duplicate Detection, IEEE Trans. Knowl. Data Eng., vol. 27, no. 5, pp. 13161329, May 2015.
- [12] Chengxiang Zhai and John Lafferty. 2001. A study of smoothing methods for language models applied to Ad Hoc information retrieval. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '01). ACM, New York, NY, USA, 334-342. DOI: https://doi.org/10.1145/383952.384019
- [13] Elfeky, Mohamed G., Vassilios S. Verykios and Ahmed K. Elmagarmid. Record Linkage : A Machine Learning Approach , A Toolbox , and A Digital Government Web Service '. (2003).
- [14] Hernandez, M.A. and Stolfo, S.J., *The merge/purge for large databases*, Proceedings of the SIGMOD 95 Conference, 1995:12738.
- [15] Hernandez, M.A. and Stolfo, S.J., *Real-world data is dirty: Data cleansing and the merge/purge problem*, Data Mining and Knowledge Discovery 2, Kluwer Academic Publishers, 1998.
- [16] Sakoe, Hiroaki; Chiba, Seibi. "Dynamic programming algorithm optimization for spoken word recognition". IEEE Transactions on Acoustics, Speech and Signal Processing. 26 (1): 4349. doi:10.1109/tassp.1978.1163055
- [17] Peter Christen and Karl Goiser, Quality and Complexity Measures for Data Linkage and Deduplication
- [18] McCallum, Nigam, Ungar, Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching, 2000, KDD
- [19] Mikhail Bilenko, Beena Kamath, Raymond J. Mooney, "Adaptive Blocking: Learning to Scale Up Record Linkage", Data Mining 2006. ICDM '06. Sixth International Conference on, pp. 87-96, 2006.
- [20] Stan Salvador, and Philip Chan. FastDTW: Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis 11.5 (2007): 561-580.
- [21] Cignoni, P.; Rocchini, C.; Scopigno, R. (1998). "Metro: Measuring Error on Simplified Surfaces". Computer Graphics Forum. 17 (2): 167174.
- [22] A. A. Taha and A. Hanbury, An efficient algorithm for calculating the exact Hausdorff distance. IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 37 pp. 2153-63, 2015.
- [23] D. H. Douglas and T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 10, no. 2, pp. 112122, 1973.
- [24] Baxter, Rohan Christen, Peter For Epidemiology, Centre. (2003). A Comparison of Fast Blocking Methods for Record Linkage. Proc. of ACM SIGKDD'03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation.





- [25] Luis Basora, Jrme Morio, Corentin Mailhot. A Trajectory Clustering Framework to Analyse Air Traffic Flows. SIDs 2017, 7th SESAR Innovation Days, Nov 2017, Belgrade, Serbia.
- [26] Maggi F. A Survey of Probabilistic Record Matching Models, Techniques and Tools, Scientic Report TR-2008. 2008.
 [27] Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/ [Online; accessed 2018-10.05]. 10-05].



