World Academy of Science, Engineering and Technology
International Journal of Aerospace and Mechanical Engineering
Vol:18, No:10, 2024

The UAV Feasibility Trajectory Prediction Using Convolution Neural Networks

Marque Adrien, Delahaye Daniel, Marechal Pierre, Berry Isabelle

Abstract—Wind direction and uncertainty are crucial in aircraft or unmanned aerial vehicle trajectories. By computing wind covariance matrices on each spatial grid point, these spatial grids can be defined as images with symmetric positive definite matrix elements. A data pre-processing step, a specific convolution, a specific max-pooling, and specific flatten layers are implemented to process such images. Then, the neural network is applied to spatial grids, whose elements are wind covariance matrices, to solve classification problems related to the feasibility of unmanned aerial vehicles based on wind direction and wind uncertainty.

Keywords—Wind direction, uncertainty level, Unmanned Aerial Vehicle, convolution neural network, SPD matrices.

I. Introduction

PATH wind forecasting is paramount, especially in aircraft flight or Unmanned Aerial Vehicle (UAV) planning contexts. When a UAV flies with a headwind, energy consumption increases significantly, potentially leading to unsuccessful crossings. Under adverse conditions, the UAV risks sustaining damage or, in the worst-case scenario, being destroyed or lost. The Mermoz project, spearheaded by a team of researchers from ISAE-SUPAERO's aerodynamics laboratory, endeavors to demonstrate the feasibility of a South Atlantic crossing from Senegal to Brazil using a hydrogen fuel cell-powered UAV system. However, the South Atlantic's unpredictable nature introduces a significant risk of the UAV running out of energy. This potential challenge is a crucial consideration in the Mermoz Project, as readers can further explore it in [10].

The challenge of predicting the feasibility of a UAV trajectory is the focus of this paper. The prediction task focuses on three key aspects:

- determining the wind direction relative to the UAV's path,
- assessing the level of uncertainty in the wind forecast,
- selecting the safest UAV trajectory.

These three problems can be defined as classification problems. Furthermore, the last two problems are related to the wind uncertainty. These problems are detailed in Section II.

The spatial grids in the ERA5 database are structured around regular latitude and longitude grids. By extracting spatial grids with the ensemble members dataset, each spatial grid point is a real 10×2 matrix. By computing the wind covariance matrix for each point in the spatial grid, we generate additional spatial grid information referred to as *wind covariance spatial grids*. Subsequently, we extract spatial grids from the ensemble spread dataset. These spatial grids are also regular latitude-longitude grids whose elements are wind uncertainties. For each spatial grid extracted from the ERA5 database, we consider a UAV trajectory connecting an origin to a destination. We then focus on the wind direction and uncertainty levels relative to this trajectory. We assign labels to each covariance spatial grid. One label is dedicated to a classification problem. The creation of a UAV trajectory between Senegal and Brazil is illustrated in Fig. 1 by extracting spatial grids from an ensemble member dataset.

We will have to solve three classification problems. Covariance spatial grids $\mathcal{C}(i,j)$ are similar to images except that each $\mathcal{C}(i,j)$ is a covariance matrix. In machine learning and statistics, the most efficient algorithms for classifying images are convolutional neural networks (CNN). Since covariance spatial grids are not traditional images, we implement a CNN capable of classifying these specific images. In image processing, CNN is designed for pixel image classification. Here, our image pixels are covariance matrices. We must fully redesign our CNN to process such images. More specifically, we implement new convolution, max-pooling, and new flatten layers capable of processing covariance spatial grids. This CNN is detailed in Section III. Then, the results obtained by our CNN on the wind direction to a UAV trajectory and uncertainty level problems are presented and detailed in Section IV.

This document has three main sections: Section II presents the ERA5 database and describes its resolution models and the assignment label for the three classification problems. Section III presents a detailed description of this CNN. In particular, we implement a pre-processing step and we redefine the convolution, maximum pooling, and flatten layers to process covariance spatial grids. Some results on the South Atlantic area are presented in Section IV.

II. The ERA5 Database

A. Spatial Grids Definition

ERA5 is the fifth generation of European Centre for Medium-Range Weather Forecasts (ECMWF) reanalysis for global climate and weather spanning the past eight decades. Data are available from 1940 onwards, and ERA5 replaces the ERA-Interim reanalysis. ERA5 provides hourly estimates for various atmospheric parameters, ocean-wave, and land-surface variables. Based on the ensemble members from ERA5 (ten wind speed vectors), one can estimate the average wind and the covariance matrix for each spatial point. This uncertainty

Marque Adrien is with CerCo, CNRS UMR5549, France (e-mail: adrien.romain.marque@orange.fr).

Delahaye Daniel is with OPTIM Team, French Civil Aviation University, France.

Marechal Pierre is with IMT, University Toulouse 3 Paul Sabatier, France

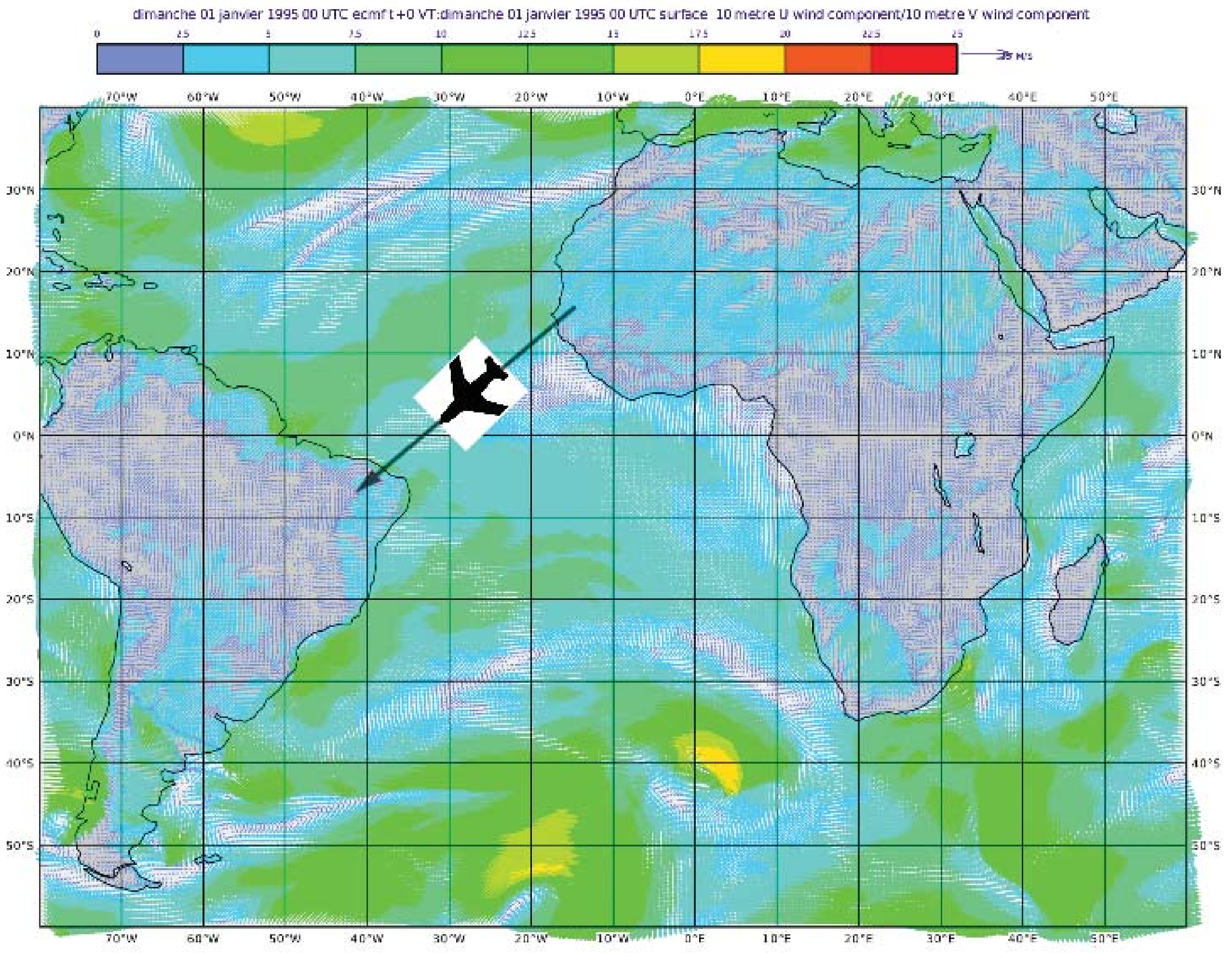
dimanche 01 janvier 1995 00 UTC ecmf t+0 VT:dimanche 01 janvier 1995 00 UTC surface 10 metre U wind component/10 metre V wind component
0
2.5
5
7.5
10
12.5
15
17.5
20
22.5
25
25 m/s
70°W
60°W
50°W
40°W
30°W
20°W
10°W
0°E
10°E
20°E
30°E
40°E
50°E
30°N
20°N
10°N
0°N
10°S
20°S
30°S
40°S
50°S

Fig. 1 Example of a created UAV trajectory between Senegal and Brazil on a spatial grid extracted with ensemble members; the first wind speed vector is represented

is mainly linked to the divergence with time of the evolution of each member. It represents flow-dependent sensitive areas, which are areas where the wind conditions are particularly variable and can significantly impact the trajectory of a UAV. Monthly mean averages are pre-computed to support various climate applications, although they are unavailable for the ensemble mean and spread. ERA5 is updated daily with a latency of approximately five days. Spatial grids extracted every three hours with reduced resolution are mapped with the following structure (1):

$$\mathcal{G}\colon \{1,...,n\} \times \{1,...,m\} \longrightarrow \mathbb{R}^{10\times 2}. \qquad (1)$$

where n is latitude, m is longitude.

Each $\mathcal{G}(i,j)$ comprises ten wind speed vectors, where each vector encapsulates u and v components, representing speed values along the x (longitude) and y (latitude) axes, respectively. Fig.2 illustrates a wind speed vector between Brazil and Senegal.

We selected the ensemble member dataset because it contains a representative estimate of the wind and its associated wind covariance matrix. Considering each $\mathcal{G}(i,j)$ as ten realisations of a random vector, we evaluate the covariance matrix along x and y directions. Usually, a covariance matrix is a semi-positive definite matrix. The specific case of the wind covariance matrix is an SPD matrix. Therefore, we obtain another set of spatial grids known as covariance spatial grids, which are mappings (2):

$$\mathcal{C}\colon \quad \{1,...,n\} \times \{1,...,m\} \quad \longrightarrow \quad \mathbb{S}_2^+ \qquad (2)$$

where $\mathbb{S}_2^+$ is the 2×2 real symmetric positive definite matrix set.

The ERA5 database dataset also contains ensemble spread and ensemble mean for surface-level analysis parameter data ensemble means. These values are computed from the ERA5 member ensemble run at a reduced resolution. The ensemble spread dataset is particularly significant, as it is based on the standard deviation computation of the ensemble members, providing a measure of the uncertainty in the data. Spatial grids are mapped with the following structure (3):

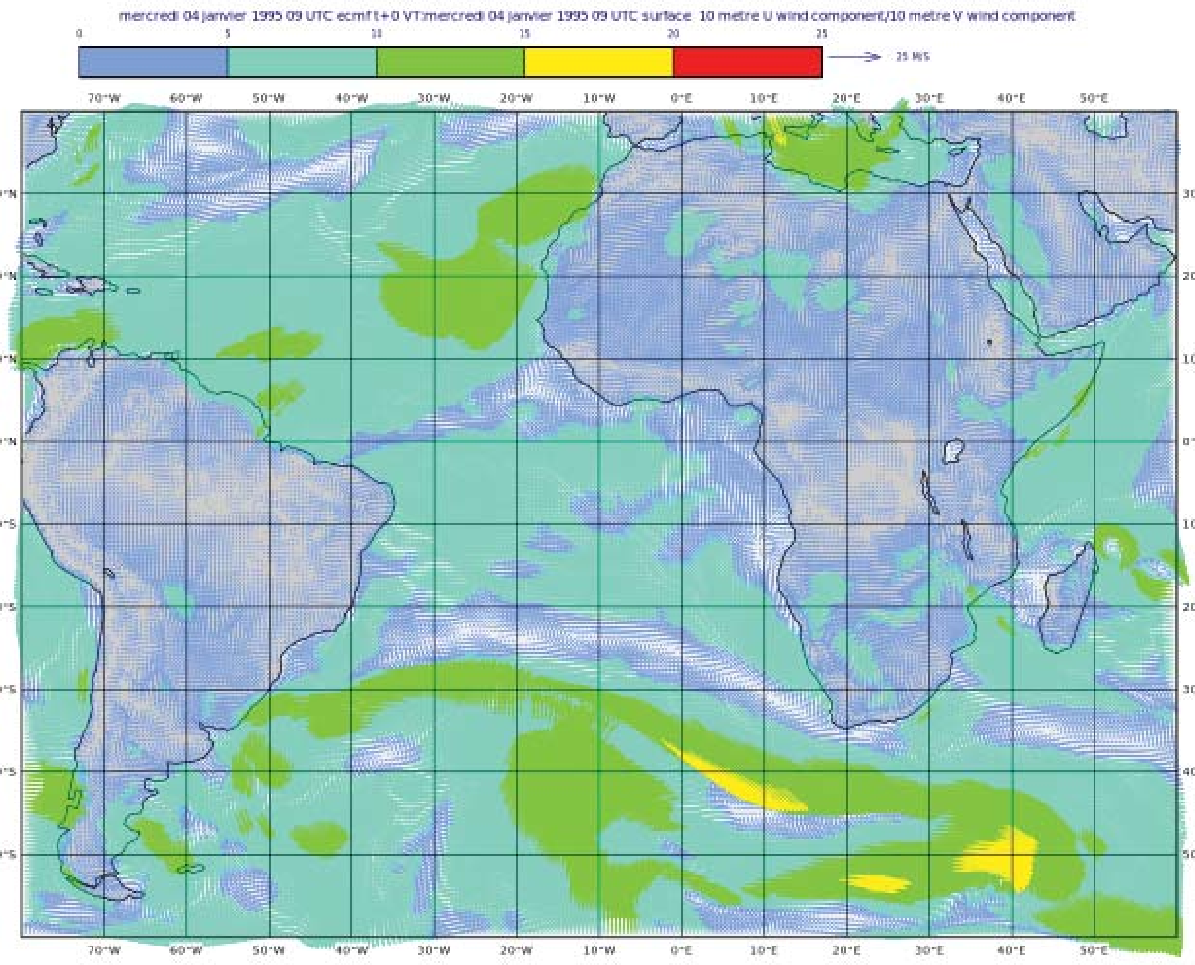
mercredi 04 janvier 1995 09 UTC ecmf t+0 VT:mercredi 04 janvier 1995 09 UTC surface 10 metre U wind component/10 metre V wind component
0
5
10
15
20
25
25 M/S
70°W
60°W
50°W
40°W
30°W
20°W
10°W
0°E
10°E
20°E
30°E
40°E
50°E
30°N
20°N
10°N
0°N
10°S
20°S
30°S
40°S
50°S

Fig. 2 Illustration of wind speed vector between Brazil and Senegal; the data are extracted from the ERA5 database

$$\mathcal{S}\colon \{1,...,n\} \times \{1,...,m\} \longrightarrow \mathbb{R} \tag{3}$$

where $\mathcal{S}(i,j)$ is the wind uncertainty located at position (i,j) on the spatial grid. Fig. 3 illustrates a spatial grid extracted from the spread ensemble dataset.

For more information on ERA5 datasets, readers can refer to [5]. The input data of a neural network are the covariance spatial grids defined by (2).

B. The Creation of a UAV Trajectory

We create a UAV trajectory on each spatial grid extracted from the spread ensemble and ensemble members datasets. A UAV trajectory is defined by an origin point (x_o, y_o) and a destination point (x_d, y_d). We create straight or ellipsoid UAV trajectories on all spatial grids. Fig. 1 shows a UAV trajectory created between Senegal and Brazil.

The straight and the ellipsoid equations define the point trajectory set $\mathcal{P} = \{(x_o, y_o), ..., (x_d, y_d)\}$. For each point $\mathcal{P}_i$, we select the nine nearest neighbours that define the set $\mathcal{A}$. The set $\mathcal{A}$ defines, therefore, an area around the UAV trajectory where each $\mathcal{A}_i$ corresponds to the nine nearest neighbours of the point $\mathcal{P}_i$.

In Fig. 4, a UAV trajectory passes through the origin point O, the destination point D, and the point U, which are included in $\mathcal{P}$. We then select the nine nearest spatial grid points closest to the point U, which are included in the set $\mathcal{A}$.

The ten wind speed vectors extracted from the ensemble members create the set $\mathcal{W}$:

$$\forall p_m = (x_m, y_m) \in \mathcal{P}, \mathcal{W} = \{\mathcal{G}(x_m, y_m), ..., \mathcal{G}(x_m, y_m)\}$$

We assume the set $\mathcal{W}$ includes P points.

C. The Overall Wind Direction

Once covariance spatial grids $\mathcal{C}$ and UAV trajectory have been created, we assign labels related to a specific classification problem. We extract spatial grids with the ensemble members dataset. We then create a UAV trajectory

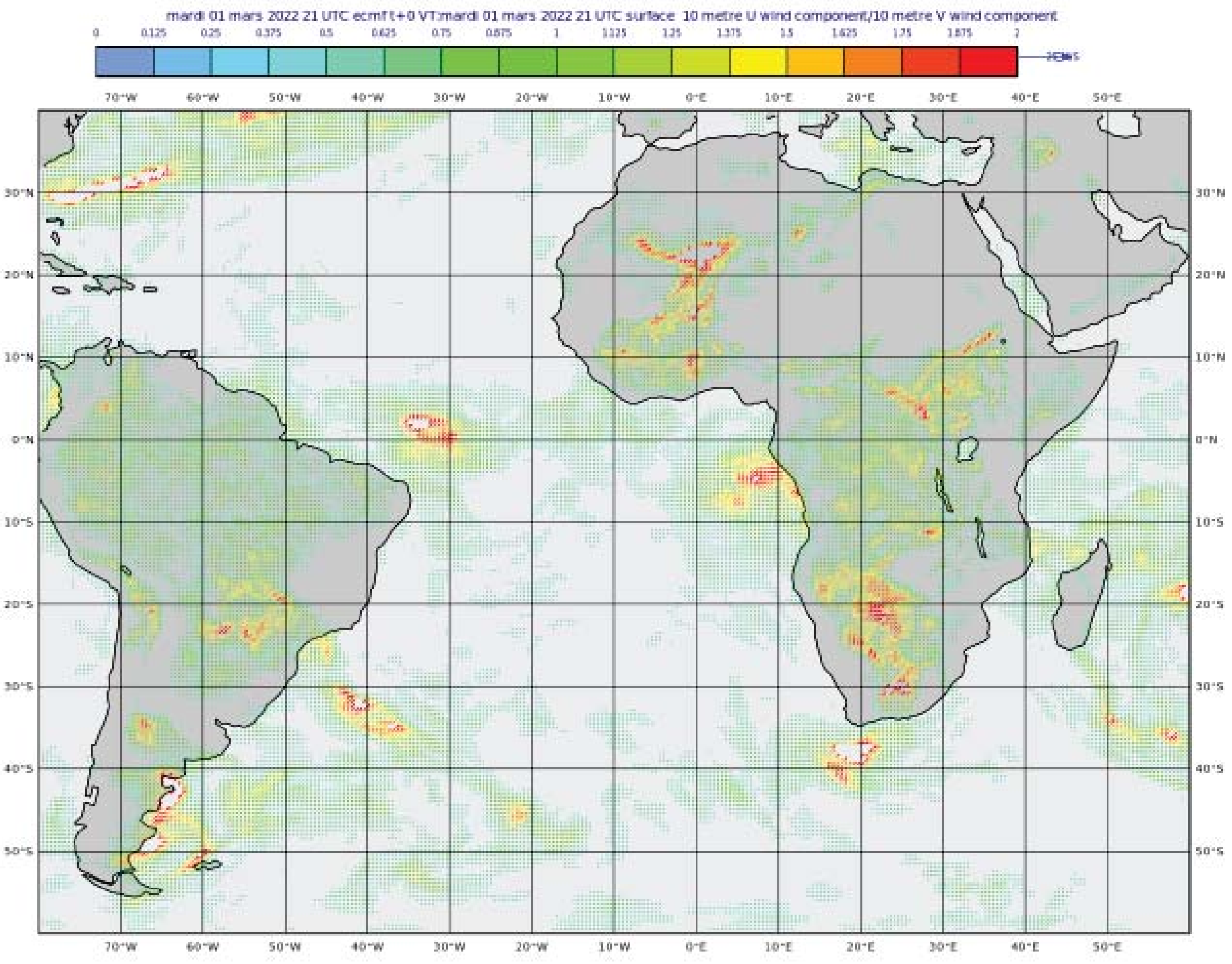

Fig. 3 Spatial grid extracted from ERA5 dataset with spread ensemble between Senegal and Brazil

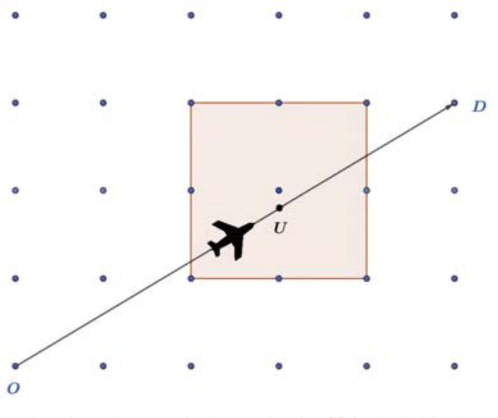

Fig. 4 The nine closest neighbors of point U included in the set $\mathcal{A}$ represented by the red area

on each extracted spatial grid. A mapping defines this UAV trajectory:

$$\begin{array}{rccc} \mathcal{T}: & \{1,...,P\} & \longrightarrow & \mathbb{R}^{10\times 2}\times\mathbb{R}^2 \\ & \mathcal{T}_i & \longmapsto & (w_i,t_i) \end{array}$$

where $w_i \in \mathcal{W}$ and t_i is the trajectory vector on the grid point $\mathcal{T}_i$ respectively.

Each point $\mathcal{T}_i$ contains ten wind speed vectors:

$$w_j = (u_j, v_j)^\top.$$

Then, we compute σ, γ, δ angles.

$$\sigma_{i,j} = \arctan(\frac{v_{i,j}}{u_{i,j}}) \times \frac{180}{\pi}$$
$$\gamma_{i,j} = \arctan(\frac{b_{i,j}}{a_{i,j}}) \times \frac{180}{\pi}$$
$$\delta_{i,j} = |\gamma_{i,j} - \delta_{i,j}|$$

These angles are illustrated in Fig. 5. For each grid point $w_i \in \mathcal{W}$, we compute the angle δ_i by applying the function h:

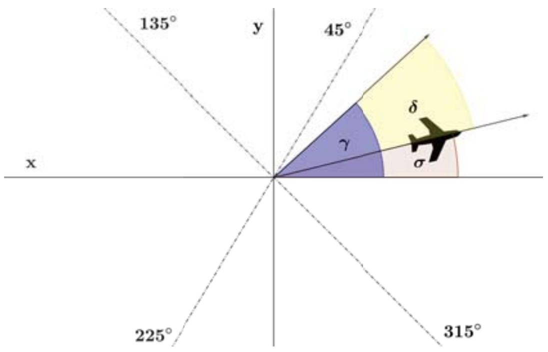

Fig. 5 Illustration of f function

$$\delta_i = h(w_i) = \frac{1}{10}\sum_{j=1}^{10}\delta_{i,j}$$

By computing the angle δ_i on the set $\mathcal{W}$, we obtain the set:

$$\Delta = \{\delta_i = h(w_i) \in \mathbb{R}, w_i \in \mathcal{W}\}.$$

We then introduce the function f, as illustrated in Fig.5. This function categorises the elements in the set Δ into four distinct groups based on their values. By applying this function to the set Δ, we can effectively categorise the wind direction into four specific groups.

$$f(\delta) = \begin{cases} 1 & \text{if } \delta \in [0, 45[\cup[315, 360[\\ 2 & \text{if } \delta \in [45, 135[\\ 3 & \text{if } \delta \in [135, 225[\\ 4 & \text{otherwise.} \end{cases}$$

By applying the f function on the δ set, we obtain another set $\mathcal{O} = \{1, 2, 3, 4\}^P$. Each element in $\mathcal{O}$ represents a specific wind direction category. The overall wind direction is obtained by applying the search maximum algorithm on the set $\mathcal{O}$. We associate with covariance spatial grids defined by (2), a label corresponding to a wind direction.

D. The Wind Uncertainty Problem

The spatial grids are extracted from the spread dataset. Trajectories created on the spatial grids extracted from the member dataset are also made on those extracted from the spread dataset. For each spatial grid i, we compute the average wind uncertainty based on the $\mathcal{A}$ set:

$$\sigma_i = \frac{1}{P}\sum_{p\in\mathcal{A}}\mathcal{S}(p)$$

For each spatial grid i, we compare $\overline{\sigma_i}$ with a predefined threshold value called $\overline{\sigma}$. The average wind uncertainty defines the average risk and feasibility of the UAV trajectory. For example, the UAV trajectory presents a critical risk if the average wind uncertainty exceeds the predefined threshold $\overline{\sigma}$. For each spatial grid i, we also compute the maximum value of the uncertainty:

$$\sigma_M = \max\{\mathcal{S}(p), p \in \mathcal{A}\}$$

We compare this maximum value with a threshold value called σ_M. The maximum uncertainty threshold arises from a simple observation: the average uncertainty of the wind along the trajectory may be low, but the UAV may cross an uncertainty area representing a non-negligible risk that could lead to its loss. The spatial grids are, therefore, divided into three groups as follows:

- $\sigma_{i,j} > \sigma_M$,
- $\sigma_{i,j} < \sigma_M, \overline{\sigma_i} > \overline{\sigma}$,
- $\sigma_{i,j} < \sigma_M, \overline{\sigma_i} < \overline{\sigma}$.

Each spatial covariance grid is assigned a label that corresponds to one of these groups.

E. The Best Trajectory Problem

This problem is based on the wind uncertainty problem. However, instead of creating a single UAV trajectory, we create two different UAV trajectories: a straight and an ellipsoid. We denote σ_L, σ_E, the average wind uncertainties related to the UAV trajectories defined as a straight and an ellipsoid. We also note $\sigma_{M,L}, \sigma_{M,E}$ the maximum wind uncertainties associated with the UAV straight and ellipsoid trajectories. The spatial covariance grids are divided into three groups. The three groups define specific situations by comparing wind uncertainties with each other and their respective threshold values.

The first group describes the failure of both routes:

- $\sigma_{M,E} \geq \sigma_M, \sigma_{M,L} \geq \sigma_M$,
- $\sigma_{M,E} < \sigma_M, \sigma_{M,L} < \sigma_M, \overline{\sigma_E} \geq \overline{\sigma}, \overline{\sigma_L} \geq \overline{\sigma}$.

The second group defines the ellipsoid UAV trajectory as the best UAV trajectory based on the following criteria:

- $\sigma_{M,L} < \sigma_M$, $\sigma_{M,E} < \sigma_M, \overline{\sigma_E} < \overline{\sigma}$,
- $\sigma_{M,E} < \sigma_{M,L} < \sigma_M, \overline{\sigma_L} \geq \overline{\sigma}, \overline{\sigma_E} < \overline{\sigma}$,
- $\sigma_{M,E} < \sigma_{M,L} < \sigma_M, \overline{\sigma_E} < \overline{\sigma_L} < \overline{\sigma}$,

The third group defines the straight-line UAV trajectory as the best based on the following criteria:

- $\sigma_{M,E} \geq \sigma_M$, $\sigma_{M,L} < \sigma_M, \overline{\sigma_L} < \overline{\sigma}$,
- $\sigma_{M,L} < \sigma_{M,E} < \sigma_M, \overline{\sigma_E} \geq \overline{\sigma}, \overline{\sigma_L} < \overline{\sigma}$,
- $\sigma_{M,L} < \sigma_{M,E} < \sigma_M, \overline{\sigma_L} < \overline{\sigma_E} < \overline{\sigma}$,

III. Convolutional Neural Network

Our neural network will have to solve classification problems. Covariance spatial grids are similar to 2D images whose pixels are covariance matrices. In the specific case of the ERA5 ensemble members dataset, covariance matrices are symmetric positive definite matrices (SPD). We will first present a state-of-the-art neural network. Secondly, we will present our neural network model, inspired by traditional CNN, by implementing a data pre-processing step that transforms SPD matrices into symmetric matrices by computing their natural logarithm. We then implement a new convolution, max-pooling, and flatten layers adapted to air covariance spatial grids.

A. State-of-the-Art

A neural network consists of node layers, typically including an input layer, one or more hidden layers, and an output layer. Each node connects to others and has an associated weight. A non-linear activation function is applied to each node, and the resulting data are passed to the next layer. Various types of neural networks are used for different use cases and data types. Convolutional neural networks (CNN) are often used for classification and computer vision tasks. A CNN is divided into a feature extraction network and a classifier network. A feature extraction network includes convolution and subsampling layers. The succession of convolution and sub-sampling layers highlights the main features defining the images (structure, contour). The output of each convolution and subsampling layer is called a feature map. The classifier network is an artificial neural network (ANN). The ANN consists mainly of a succession of fully-connected layers to which non-linear activation functions are applied. The sequence of layers is used to classify feature maps into different labels. The feature maps are passed to the classification neural network via the flatten layer. The CNNs have demonstrated efficiency in tasks such as image classification [8], object detection [2], facial recognition [10], [6], video analysis and [7]. However, the input data of the neural network are covariance spatial grids whose $\mathcal{C}(i,j)$ is a covariance matrix. A significant milestone was recently achieved by successfully implementing a neural network to classify images with SPD matrix elements. This specific neural network includes two operation modules. On the one hand, a Riemannian batch regularisation layer is first proposed. On the other hand, a second module realises the Riemannian pooling operation with geometric computations on the Riemannian manifolds based on the Riemannian barycenter, metric learning, and Riemannian optimisation. The implemented neural network is applied to three visual classification tasks (video-based emotion recognition, dynamic scene classification, and hand action recognition). For more information on this neural network, the readers can refer to [13].

B. The Data Pre-processing Step

The input data of our neural network are covariance spatial grids $\mathcal{C}$. Since $\mathcal{C}(i,j)$ is a covariance matrix, usually, a covariance matrix is a semi-symmetric positive definite matrix. In the specific case of our database, each $\mathcal{C}(i,j)$ covariance matrix is an SPD matrix. Covariance spatial grids are thus mappings (4):

$$\mathcal{G}\colon \{1,...,n\}\times\{1,...,m\} \longrightarrow \mathbb{S}_2^+. \tag{4}$$

However, $\mathbb{S}_2^+$ is a Riemannian manifold; we can not directly apply the convolution layer. Since $\mathbb{S}_2^+$ is also a Lie group [1], by computing the natural logarithm of each $\mathcal{C}(i,j)$, we obtain another spatial grids which are mappings:

$$\begin{array}{rccc} X\colon & \{1,...,n\}\times\{1,...,m\} & \longrightarrow & \mathbb{S}_2 \\ & X(i,j) & \longmapsto & \log(\mathcal{C}(i,j)). \end{array}$$

The spatial grids X are called *symmetric spatial grids*. To compute the natural logarithm of each $\mathcal{C}(i,j)$, we use the singular value decomposition (SVD) method. More specifically, each $\mathcal{C}(i,j)$ can be written:

$$\mathcal{C}(i,j) = U_{i,j}.D_{i,j}.V_{i,j}$$

where $D_{i,j}$ is the diagonal matrix whose elements are eigenvalues associated with $\mathcal{C}(i,j)$.

As each $\mathcal{C}(i,j)$ is an SPD matrix, its eigenvalues are strictly positive. Each $X(i,j)$ is therefore computed by applying the natural logarithm to diagonal matrix $D_{i,j}$:

$$X(i,j) = \log(\mathcal{C}(i,j)) = U_{i,j}.\log(D_{i,j}).V_{i,j}.$$

The symmetric spatial grid X defined by Equation (4) is the input data of our neural network. In the sequel, we denote as X, Z the input and output of convolution, max-pooling and flatten layers respectively. We also denote $G_{\text{data}}, G_{\text{loss}}, G_{\text{loss}}$ as the gradient data, the gradient loss, and the gradient kernels of convolution, max-pooling, and flatten layers.

C. The Convolution Layers

The convolution layer is a fundamental component of a CNN. An input data X is filtered by a real kernel K. The real kernel K moves through the receptive fields of the image to detect features. The mathematical operation associated with the convolution layer is the convolution layer denoted as $*$. Unlike conventional neural networks, each $X(i,j)$ is a symmetric matrix. Since $\mathbb{S}_2$ is an Euclidean vector space, we adapt discrete convolution to spatial grids. Therefore, the input spatial grids X filtered by a real $r\times s$ kernel K are defined by (5):

$$Z = X*K = \sum_{k_1=0}^{r}\sum_{k_2=0}^{s} X(n_1-k_1, n_2-k_2)\times K(k_1,k_2) \in \mathbb{S}_2. \tag{5}$$

The backward pass for convolution layers can be divided into two steps: data gradient and kernel gradient computation. Firstly, the data gradient results from the discrete convolution operation between the padded loss gradient and the $180°$ rotated kernels, denoted by K_{180}, associated with the convolution layer. The data gradient for spatial grids is based on (6):

$$G_{\text{data}} = p(G_{\text{loss}}) * K_{180}, \tag{6}$$

where p corresponds to the zero-padding operation.

Secondly, the kernel gradient results from the input convolution layer X filtered by the loss gradient G_{loss} as defined by (7):

$$G_{\text{kernel}} = X * G_{\text{loss}}. \tag{7}$$

For more information on convolution layers, readers are referred to [12].

D. The Max-Pooling Layer

A max-pooling layer is a subsampling layer. A subsampling layer reduces the reliance on precise positioning within feature maps from a convolution layer. In the conventional subsampling layer, a kernel of dimensions $z \times t$ is associated with the subsampling layer, dividing images into $q = [\frac{n}{z}][\frac{m}{t}]$ cells where $[.]$ is the integer part. A mapping defines these cells:

$$c: \quad \{1, ..., z\} \times \{1, ..., t\} \quad \longrightarrow \quad \mathbb{S}_2.$$

where z, t are the kernel dimensions associated with the max-pooling layer.

Each cell in the max-pooling layer retains only one piece of information via a specific mathematical operation. In this case, the operation is the maximum operation. Each point grid $X(i,j)$ is a symmetric matrix, so the maximum function must be redefined. Since each $X(i,j)$ is a symmetric matrix, each cell includes a symmetric matrix. We apply the Frobenius norm, denoted by $||.||_F$, on each q cell:

$$\begin{array}{llll} d: & \{1, ..., n\} \times \{1, ..., m\} & \longrightarrow & \mathbb{R} \\ & I(i,j) & \longmapsto & ||q(i,j)||_F. \end{array}$$

We apply a maximum search algorithm to each d associated with all q cells. We retain the symmetric matrix related to the maximum Frobenius norm. The positions of maximum values are stored in a mask denoted by M. This mask M is then used to compute the gradient data. The operation associated with the backward max-pooling layer is defined by Equation (8).

$$G_{\text{data}}(i,j) = \begin{cases} G_{\text{loss}}(i,j) & \text{if } (i,j) \in M \\ 0 & \text{otherwise.} \end{cases} \qquad (8)$$

Since each element of G_{loss} is a symmetric matrix, zero values are replaced by a 2×2 zero matrix. Deeper within the network, feature maps become more exclusive and informative, reducing redundancy primarily due to repeated convolutions and information compression by subsampling layers. For more information on subsampling layers, readers are referred to [12].

E. Flatten Layer

Similar to conventional flatten layers, our flatten layer vectorizes the output of the feature extraction network. More specifically, our flatten layer is defined by (9):

$$Z(t) = X(i,j,a,b), \qquad (9)$$

where $t = im + 4j + 2a + b$.

The backward pass of our flatten layer reshapes the gradient from the classifier network. More specifically, the operation associated with our flatten layer is defined by (10):

$$G_{\text{data}}(i,j,a,b) = G_{\text{loss}}(t) \qquad (10)$$

where $t = im + 4j + 2a + b$.

The outputs of the flatten layer are real vectors. Then, our classifier network is a conventional ANN. For more information on the ANN, the readers can refer to [4]. Therefore, the covariance spatial grids are forwarded into a traditional ANN. The results are compared to the target using a loss function. Subsequently, the gradient is backwarded through our neural network. The hyper-parameters associated with our neural network, like batch size epochs and learning rate, play the same role as in conventional neural networks.

IV. Results

The wind database is extracted from the ERA5 database. This database contains 12000 spatial grids. By computing covariance matrices, as it is described in Section II, these spatial grids are mappings:

$$X: \quad \{1, ..., 100\} \times \{1, ..., 140\} \quad \longrightarrow \quad \mathbb{S}_2^+$$

These spatial grids, corresponding to the area used in the Mermoz project, are extracted between the city of Dakar in Senegal and the city of Natal in Brazil. The time frame for this extraction is between 2018 and 2022. Each covariance spatial grid is associated with several labels depending on the classification problem described in Section II. For the wind uncertainty and the best trajectory choice problems, we define the predefined average wind uncertainty and maximum wind uncertainty threshold as:

- $\overline{\sigma} = 1m.s^{-1}$,
- $\sigma_M = 1.5m.s^{-1}$.

The labels for the three classification problems are assigned using the hard labelling method. For a more detailed understanding of this labelling method, please refer to [3].

We implement a specific neural network illustrated in Fig. 6 to classify spatial grids. Our neural network begins with a data pre-processing step (orange). It is divided into two networks: a feature extraction network and a classifier network. A flatten layer (purple) separates these two networks. The feature extraction network contains three blocks. Each block contains a convolution layer (yellow) and a max-pooling layer (red). The classifier network includes three fully-connected layers. The first layer consists of 6279 neurons, the second layer has 567 neurons, and the last has two or three neurons. The sigmoïd activation function is used on the first and second layers, whereas the softmax activation function is applied to the last layer. The hyper-parameters of our neural network are similar to classical neural networks: the number of epochs, the training batch size, and the learning rate. We train our CNN model with 30 epochs, a batch size value equal to 50 and a learning rate equal to 0.05.

A. The Overall Wind Prediction Problem

In the context of the Mermoz Project, the wind direction prediction problem holds significant importance. The training step for this problem is illustrated in Fig. 7. Our neural network, designed to tackle this problem, begins by correctly classifying 6.14% of wind direction spatial grids and ends by classifying 96.75%. During the testing step, it correctly classifies 93.4% of the covariance spatial grids. The accuracy of this prediction is crucial for the success of UAV crossings over the South Atlantic.

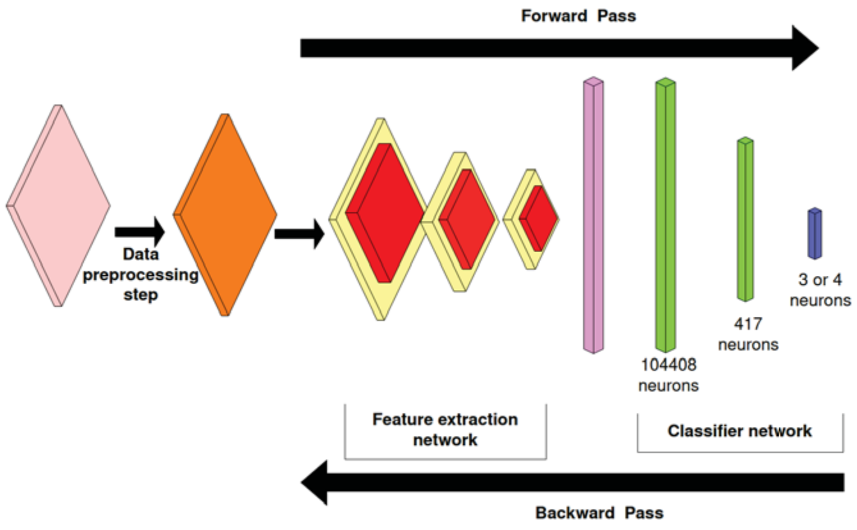

Fig. 6 Architecture of our neural network

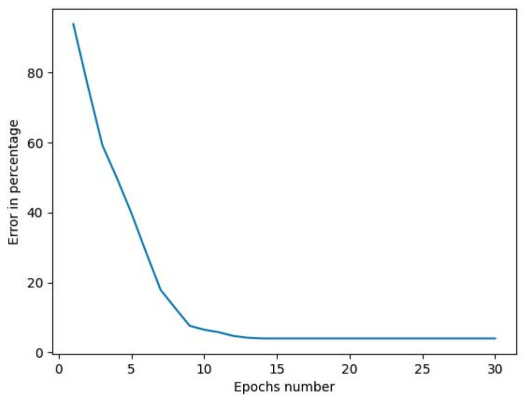

Fig. 7 Error percentage evolution concerning epochs number for the overall wind direction prediction

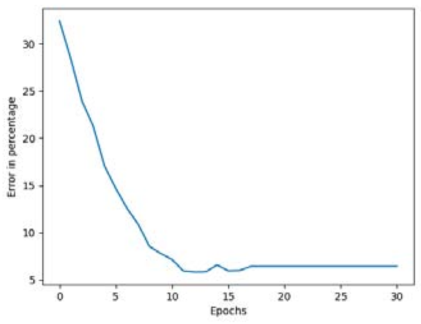

Fig. 8 Error percentage evolution concerning epochs number for the wind uncertainty prediction

B. The Wind Uncertainty

For the wind uncertainty classification problem, our neural network begins by correctly classifying 65.5% of covariance spatial grids and ends by classifying 97.75%. The training step is illustrated in Fig. 8. During the testing step, our neural network correctly classifies 92.5% of the covariance spatial grids.

The neural network predicts that the spatial grid Fig. 9 belongs to the third group. The UAV crossing the South Atlantic is thus expected to be safe because the average and maximum wind uncertainties do not exceed their respective thresholds.

C. The Best Trajectory Problem

For the wind uncertainty classification problem, our neural network begins by correctly classifying 6.13% of covariance spatial grids and ends by classifying 96%. The training step is illustrated in Fig. 10. Our neural network correctly classifies 91% of the covariance spatial grids during the testing step.

The neural network suggesting the best trajectory choice for the UAV is illustrated in Fig. 11. The straight UAV trajectory is the safest because the average wind and the maximum uncertainty are lower than their respective threshold values. On the other hand, the ellipsoidal UAV trajectory crosses an area where the maximum uncertainty exceeds its threshold value, which calls into question the trajectory.

If we propose several trajectory options for the Mermoz Project, the neural network will identify the safest trajectory based on the average wind uncertainty along the trajectory and the maximum values.

D. Discussions

The data pre-processing step transforms the covariance spatial grids into symmetric spatial grids. This step is essential for a neural network to process spatial covariance grids. By computing the natural logarithm of each pixel, a neural network can process spatial covariance grids. Then, the symmetric spatial grids are propagated into a feature extraction network. The feature extraction network is a succession of convolution and max-pooling layers explicitly designed for symmetric spatial grids. The feature extraction network is crucial to highlight features of symmetric spatial grids. The implemented flatten layer facilitates the transfer

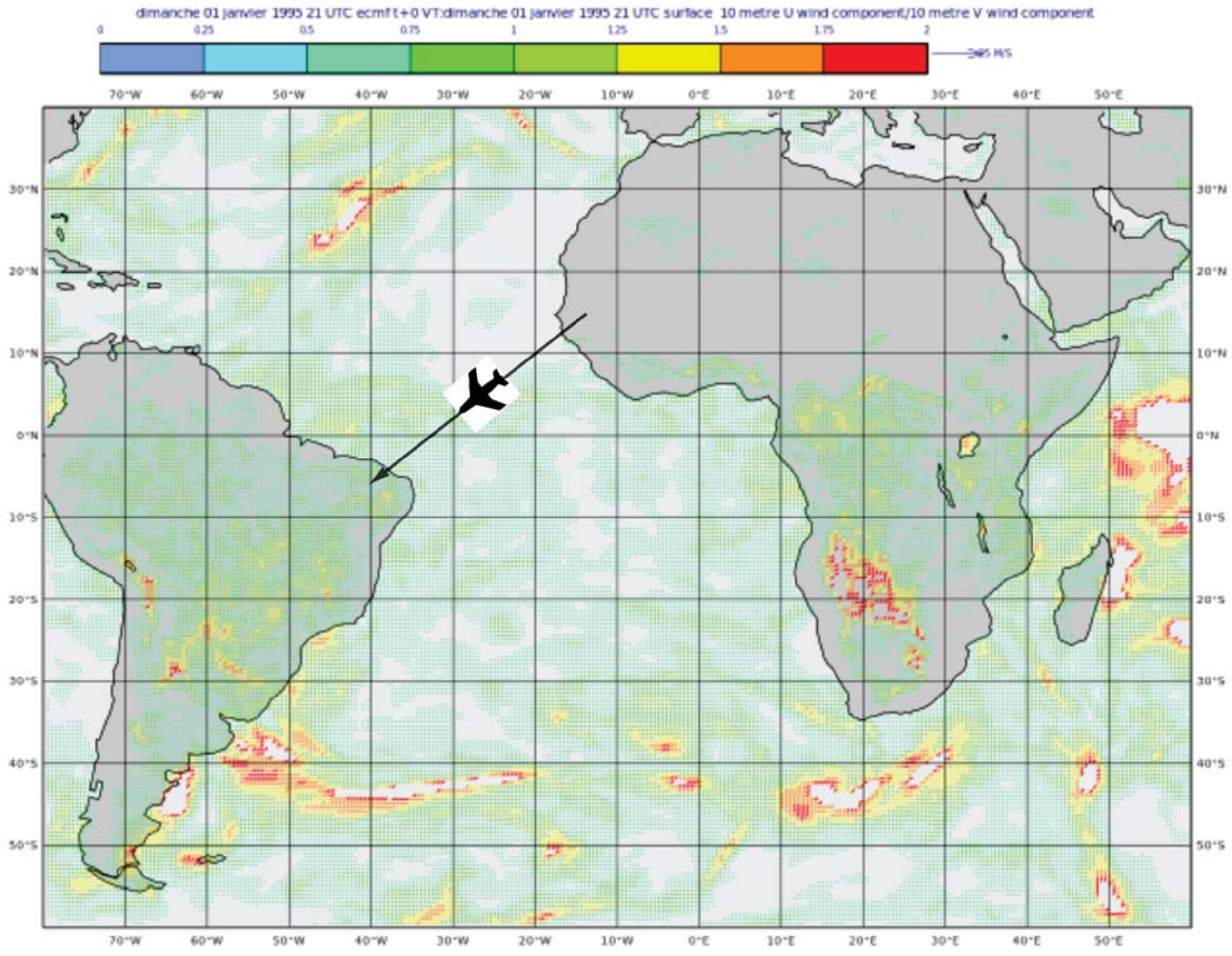
dimanche 01 janvier 1995 21 UTC ecmf t+0 VT:dimanche 01 janvier 1995 21 UTC surface 10 metre U wind component/10 metre V wind component
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
70°W
60°W
50°W
40°W
30°W
20°W
10°W
0°E
10°E
20°E
30°E
40°E
50°E
30°N
20°N
10°N
0°N
10°S
20°S
30°S
40°S
50°S

Fig. 9 The neural network predicts that the associated covariance spatial grid belongs to the third group

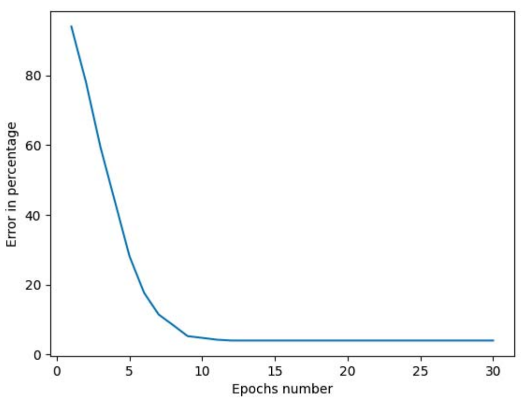

Fig. 10 Error percentage evolution with epochs number for the best trajectory prediction

of feature maps from the feature extraction network to the classifier network. The classifier network is a succession of fully-connected layers with activation layers. The classifier network uses the feature maps from the feature network to classify the spatial covariance grid maps.

Scientific studies focus on the planning of multi-UAV trajectories. Multi-UAV planning problems are solved by traditional and intelligent algorithms (heuristics and machine learning). For more information, readers can refer to [14]. Another study focuses on the high feasibility of UAV trajectories. It has access to UAV signals. It proposes a trajectory mapping network (TMN) based on deep learning to approximate the UAV system. Then, a new time series CNN neural network (TSCNN) is proposed for the TMN to improve its computation speed and prediction accuracy. The proposed neural network takes into account certain constraints, such as wind and possible obstacles. The results obtained by our neural network are comparable to those obtained in [9]. However, it is complex to compare the results obtained due to differences based mainly on the input data (real matrices vs. spatial covariance grids) and the architecture of the neural network. In fact, these results are complementary.

The neural network that is implemented predicts the overall wind direction of the UAV trajectory. In the context of the Mermoz project, the wind direction problem is of significant importance. The accuracy of this prediction is crucial to the

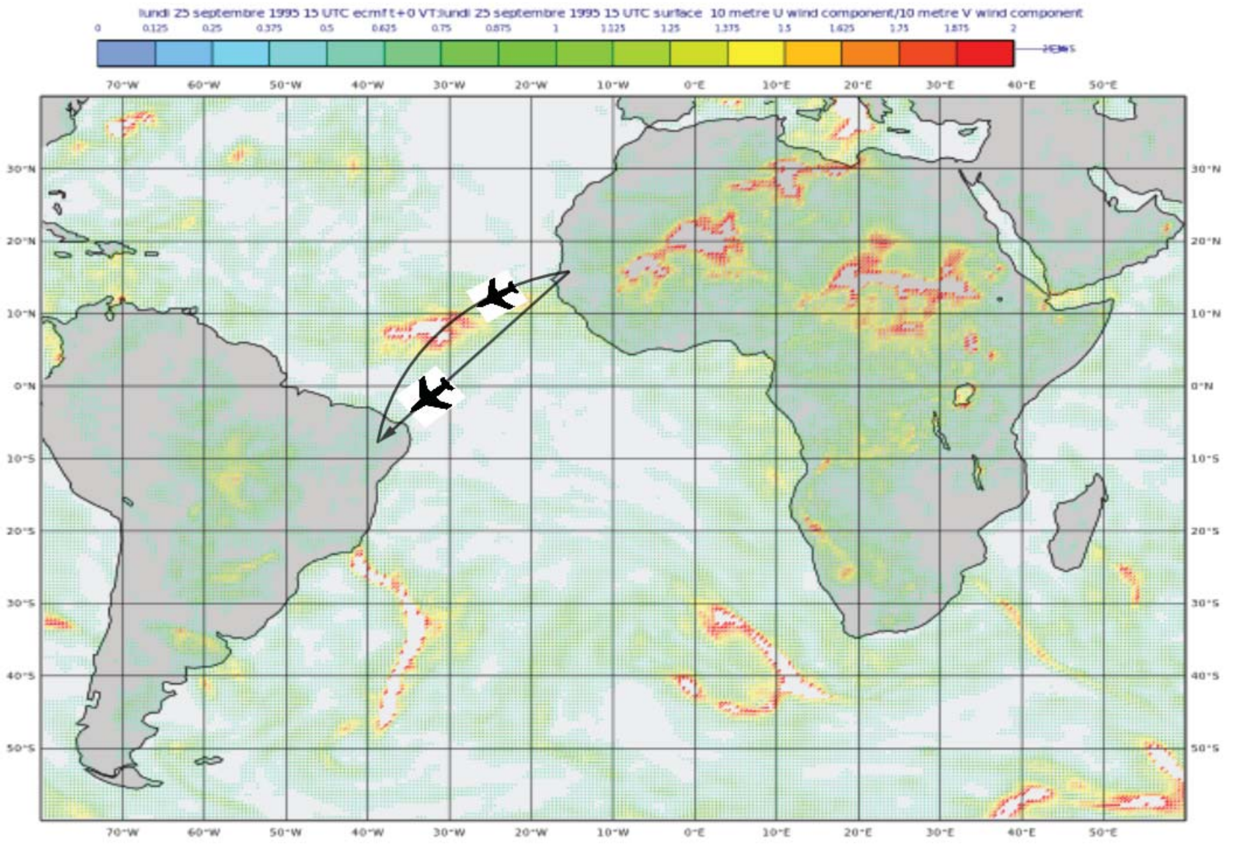

Fig. 11 Illustration of the best trajectory choice problem; The neural network indicates that T_2 is the best trajectory choice

success of the UAV crossing the South Atlantic. If the wind direction contradicts the UAV trajectory, this can lead to excessive fuel consumption and potential loss. In addition, the implemented neural network can predict whether the average and maximum wind uncertainty exceeds certain thresholds. The wind uncertainty problem poses a risk if the UAV crosses an area, significantly compromising the UAV crossing the South Atlantic. Based on the wind uncertainty problem, the neural network can predict the safest UAV trajectory. The results obtained for the three classification problems show that the neural network can help in making decision for the safest UAV trajectory.

V. Conclusions

In this paper, we focus on the feasibility of a UAV trajectory. The feasibility of a UAV trajectory is based on the average wind direction to a UAV trajectory and the wind uncertainty levels along this trajectory. We were inspired by the Mermoz project, which involved a UAV crossing the South Atlantic.

We extract from ERA5 two datasets: the ensemble members and the ensemble spread datasets. We make a UAV trajectory between Dakar and Natal from the two datasets and send it to each spatial grid. We assign labels to each covariance spatial grid corresponding to the three classification problems. The UAV trajectory, the covariance spatial grid creation, and the assignment labels are detailed in Section II.

We then implement a specific neural network capable of classifying covariance spatial grids. To tackle these classification problems, we use a novel approach. The specificity of this neural network is based on the implementation of data pre-processing step, new convolution, new max-pooling, and new flatten layers able to process images composed of covariance matrix pixels. The neural network is detailed in Section III.

During the testing step, the neural network demonstrates its prowess by correctly classifying between 91% and 96.75% of covariance spatial grids. For a more detailed analysis of these impressive results, readers should refer to Section IV. The implemented neural network can identify the average wind direction to a UAV trajectory, which can train a critical fuel consumption and a loss of UAV. It can also identify if the UAV trajectory represents a risk based on the wind uncertainty area (average wind uncertainty and maximum wind uncertainty). Finally, he can identify the best trajectory based on the wind uncertainty area. The combination of this information could help engineers and scientists make decisions for the UAV crossing the South Atlantic.

It would be interesting to create more than two trajectories to see if the neural network can still identify the best UAV trajectory. One possible future application is linked to the research carried out as part of the Mermoz project described in the introduction. One of these projects led to developing and implementing a Python module called Dabry, which computes the optimised trajectory between two points [11]. For example, given the departure date, the Python module Dabry computes the optimised trajectory between Dakar (in Senegal) and Natal (in Brazil). The module also computes the time necessary to achieve the optimal trajectory. With knowledge of the spatial grid and its evolution over time, we can use our neural network to determine if the optimal path is achievable by associating it with the wind uncertainty.

Acknowledgment

This work was supported by the convention CIFRE 2019/1343 Siemens Healthineers and CerCo CNRS.

References

[1] Johannes Jisse Duistermaat and Johan AC Kolk. *Lie groups*. Springer Science & Business Media, 2012.

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[3] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[4] Simon Haykin. *Neural networks and learning machines, 3/E*. Pearson Education India, 2010.

[5] Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., and Thépaut, J-N. ERA5 hourly data on pressure levels from 1940 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS).

[6] Guosheng Hu, Yongxin Yang, Dong Yi, Josef Kittler, William Christmas, Stan Z Li, and Timothy Hospedales. When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 142–150, 2015.

[7] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[9] Yiheng Liu, Honglun Wang, Jiaxuan Fan, Jianfa Wu, and Tiancai Wu. Control-oriented uav highly feasible trajectory planning: A deep learning method. *Aerospace Science and Technology*, 110:106435, 2021.

[10] Yuxin Peng, Xin Huang, Jinwei Qi, Junjie Zhao, Junchao Zhang, Yunzhen Zhao, Yuxin Yuan, Xiangteng He, and Jian Zhang. Pku-icst at trecvid 2015: Instance search task. In *TRECVID*, 2015.

[11] Bastien Schnitzler. https://github.com/dabry-route/dabry, 2023.

[12] Jonas Teuwen and Nikita Moriakov. Convolutional neural networks. In *Handbook of medical image computing and computer assisted intervention*, pages 481–501. Elsevier, 2020.

[13] Rui Wang, Xiao-Jun Wu, Ziheng Chen, Tianyang Xu, and Josef Kittler. Learning a discriminative spd manifold neural network for image set classification. *Neural networks*, 151:94–110, 2022.

[14] Yunhong Yang, Xingzhong Xiong, and Yuehao Yan. Uav formation trajectory planning algorithms: A review. *Drones*, 7(1):62, 2023.