Évaluation of air traffic complexity metrics using neural networks and sector status

David Gianazza, Kevin Guittet

Evaluation of air traffic complexity metrics using neural networks and sector status

David Gianazza
Laboratoire d'Optimisation Globale

Kévin Guittet
Laboratoire d'Economie et d'Econométrie de l'Aérien

DSNA/DTI/SDER (former CENA)
7, avenue Edouard Belin 31055 Toulouse Cedex
Email : {lastname}@recherche.enac.fr

***Abstract*— This paper presents an original method to evaluate air traffic complexity metrics. Several complexity indicators, found in the litterature, were implemented and computed, using recorded radar data as input. A principal component analysis (PCA) provides some results on the correlations between these indicators. Neural networks are then used to find a relationship between complexity indicators and the actual sector configurations. Assuming that the decisions to group or split sectors are somewhat related to the controllers workload, this method allows to identify which types of complexity indicators are significantly related to the actual workload.**

I. Introduction

Much research has been conducted over the last decade to help understanding air traffic complexity and controller workload. The inadequacy of the *aircraft count* to appropriately reflect the traffic complexity has now been acknowledged for a long-time, and complementary indicators such as "traffic mix", "number of potential conflicts" and others, have been (and still are) designed. A linear combination of these variables, often referred to as *dynamic density*, is likely to better fit traffic complexity than individual indicators. It is used throughout most studies, where the correllation of a set of indicators with a quantifiable variable, assumed to represent the actual traffic complexity, is maximized.

A possible shortcoming of this methodology is that potentially non-linear relations between indicators are missed (see [1] and the concern of Eurocontrol when writing calls for proposals). But, more importantly, the choice of the dependent variable is crucial to determine how well complexity is actually measured. Indeed, physical activity, as used in [2] and [3], miss the important cognitive part of the controller activity. On the other hand, physiological indicators ([4], [5]) seem difficult to exploit and how well they relate to traffic complexity is unclear. Finally, widely used subjective ratings ([6], [7]) provide high quality data (as they obviously relate to the kind of complexity investigated), but are often seen as subject to biases (such as the recency effect denounced in [5], and the possibility of raters errors in the case of "over-the shoulder workload ratings" [8]). In all of these cases, data are very expensive to collect, as they require the active participation of controllers. Databases are often small and might exhibit low variability, which may in turn harm the statistical relevance of the results. This phenomenom is acknowledged in [7], where the overfitting of data is clearly a consequence of a lack of observations rather than a misspecification of the neural network. Finally, as these complexity metrics may be used to design computer-assisted control tools or traffic management tools, and to help organizing airspace, it is surprising to notice that the question of the relevance of the complexity measured to the final goal is scarcely discussed. The question really is to understand which complexity is measured and how well it relates to the foreseen application (benchmarking, improvements in airspace organization, design of new tools...).

This paper is motivated by former studies on optimal airspace sector configurations ([9], [10]) and intends to improve the criterion used therein to evaluate sector configurations. The basic idea, introduced in [11], is that the decisions to split a sector, mostly taken when the controller is close to overload, are linked to traffic complexity and may therefore provide an acceptable dependent variable. Interestingly enough, collecting data on sector configurations does not require controllers active participation, as current outcomes from control centers can be used, while related flight informations are available from recorded radar tracks. As such, raw data needed in our study are noticeably cheap to collect and might be produced in large quantities. The price to pay is that these data are noisy, as we may not be sure that a sector splitting (resp. merging) decision is directly related to overload (resp. underload). Other factors might distort data, such as training of unexperienced controllers, meteorological hazards, military airspace use... However, we will assume that the impact of these phenomena on the accuracy of the results is limited, particularly because

of the kind of complexity we are looking at here. Indeed, this work is conducted in the perspective of future pre-tactical applications (e.g. sector planning) and thus does not ask for as much details as studies of instantaneous workload would (on the opposite, benchmarking of ATC centers would require an even coarser granularity, as indicators are averaged on wide temporal and geographical horizons [12], [13]). To investigate the link between complexity indicators and sector configurations, we use neural networks, as non-linear interactions are suspected.

The paper is organized as follows. Section II briefly describes the indicators used throughout the study, while section III presents the raw data from which the final database is built. A Principal Component Analysis (PCA) is then performed in section IV to restrain the dimensionality of the data. Neural networks are introduced in section V and their results are presented and discussed in section VI. Section VII concludes.

II. Air traffic complexity indicators

The accuracy of the results of a study related to air traffic complexity is strongly dependent of the diversity and quality of the chosen individual complexity indicators. Many have been suggested to help describe the controllers workload, and it is hardly possible to implement the entire pool. In order to limit the number of variables to be (re)programmed and present indicators that are representative of the *dynamic density* litterature, we focused on the ones selected by Kopardekar [6] in its unified complexity metric[1]. These indicators, such as references of studies where they were used and where definitions may be found, are presented in Table I. We also implemented several indicators inspired by studies conducted elsewhere in the SDER (former CENA). Definitions are indicated in appendix[2]. Finally, we also used incoming flows as explanatory variables, as they may be a significant factor in the decision to split (or merge) a sector.

III. Input data

The indicators are computed every round minute of the day, using recorded radar data, environnement data (sector description), and recorded sector configurations of the five french ATC centers. The sector configurations are recorded every round minute of the day, which explains our choice concerning the frequency at which we compute the indicators.

[1]Though we were not always able to find an explicit formula, and thus missed seamingly important indicators like, e.g., "MET_airspace structure". Note that this difficulty to get clear definitions is also reported by Eurocontrol in [14].

[2]Further informations and discussions about indicators are to be found in the internal note [15].

Indicator	Definition	Used in
Nb	Number of aircraft	[16] [7] [17] [6]
Nb^2	Squared number of aircraft	[17] [6]
σ^2_{gs}	Variance of ground speed	[7] [6]
N_{ds}	Number of descending aircraft	[2] [16] [7] [6]
N_{cl}	Number of climbing aircraft	[6] [2] [16] [7]
$\frac{\sigma^2_{gs}}{\overline{gs}}$	Ratio of standard deviation of speed to average speed	[7] [6]
F_5	Incoming flow (hozizon 5mn)	[11]
F_{15}	Incoming flow (hozizon 15mn)	[11]
F_{30}	Incoming flow (hozizon 30mn)	[11]
F_{60}	Incoming flow (hozizon 60mn)	[11]
vprox_1	Vertical proximity	[7] [17] [6]
vprox_2	*See appendix*	[7] [17] [6]
hprox_1	Horizontal proximity	[7] [17] [6]
Dens		[18]
track_disorder		[18]
speed_disorder		[18]
Div		[18]
Conv	*See appendix*	[18]
sensi_d		[18]
insen_d		[18]
sensi_c		[18]
insen_c		[18]
inter_vert		[13]
avg_vs	*See appendix*	[13]
inter_hori		[13]
creed_ok	*See appendix*	[19], [20]
creed_pb		[19], [20]

TABLE I
Chosen subset of air traffic complexity indicators

Radar data is available in several forms: records made by each center, with one position every twelve seconds, in average, and a global record of the five centers, with one position every three minutes. Several months of global records were available, whereas the centers local records were not readily available, at least for a sufficiently long period of time. So we used the global records (made by the IMAGE system), and interpolated the aircraft positions in order to get one position per minute. As many trajectory changes may occur within three minutes of flight, the computed positions are not highly accurate, and this may introduce a bias in the indicators values. However, this bias is most probably of small importance in our problem: we just want to predict when a sector will be merged into another one, or split in several smaller sectors. We are not considering the instant workload, which may require a very high level of accuracy on the aircraft position, speed, and so on. To be sure that this bias is small, we should compare the computed positions, and maybe also the indicators values, using local centers records, and global records, on small data samples. This is left for future work.

Several months of recorded traffic are available. However, considering the volume of data, it would be tedious to run several experimentations on very large

data samples. So, we have restricted our choice, at least for the moment, to one day of traffic (1st june, 2003). Once we have found the most significative complexity indicators, it will be possible to re-train the neural network on larger data samples.

On the chosen day, 103 different sectors were armed. The term "sector" means here either an elementary sector, or a set of elementary sectors merged together, and handled on a single controller's working position. The air traffic complexity indicators were computed for each of these sectors, every minute of the day, together with the sector status (*merged*, *armed*, or *split*). This data was split into two sets : about sixty percent was randomly selected in order to *train* the neural network, and the rest was used to *test* the trained network on fresh data.

This single day of traffic already provides a big volume of data, as detailed in table II, with a great diversity of geographic sectors, and with enough data in each class of sector status.

	Total	Merged	Armed	Split
Train	71270	46.6%	27.0%	26.4%
Test	47513	46.4%	27.0%	26.6%

TABLE II
NUMBER OF MEASURES AVAILABLE, ON THE 1ST JUNE OF 2003.

Before applying the neural network to complexity indicators and sector statuses, let us first discuss the correlations between the indicators, using a *principal component analysis method*.

IV. PRINCIPAL COMPONENT ANALYSIS

Including incoming flows, we end up with 27 complexity indicators. Given the neural network greediness in numbers of parameters and the high multicollinearity of the data, we will use principal components rather than individual indicators for our experiments. 6 main components were identified (corresponding to eigenvalues greater than 1), that covered more than 76 % of the variance of the data set. These components are interpreted below.

C1: Eigenvalue 12.6, 46.7 % of the variance of the data set. Appart from *vprox_1* and *vprox_2*, all variables are strongly (and positively) correlated with this component, explaining its high associated eigenvalue. This component may be seen as a "size factor", and we follow [6] on the term "Overall monitoring". This component is strongly representative of the *aircraft count*.

C2: Eigenvalue 2.78, 10 % of the variance of the data set. This component is strongly correlated with *avg_vs*, σ_{gs}^2 and $\sigma_{gs}/\overline{gs}$ (resp. 0.70, 0.69 and 0.68). Accounting for the impact on ground speed of vertical evolution of aircraft, this component may be seen as related to the *ground speed variance*, and the *aircraft vertical evolutions*.

C3: Eigenvalue 1.96, 7.3 % of the variance of the data set. This component is mainly correlated with *incoming flows*.

C4: Eigenvalue 1.25, 4.6 % of the variance of the data set. Appart from variables directly related to the traffic volume (N, *inter_hori*), the correlation on this component is high with *insen_c* and CREED indicators, and thus might be related to *converging flows* and *anticipation of conflicts*.

C5: Eigenvalue 1.06, 3.9 % of the variance of the data set. This component is strongly correlated with Div and *insen_p*, and seems therefore mostly linked with *divergent flows*.

C6: Eigenvalue 1.03, 3.8 % of the variance of the data set. This component is strongly correlated with the vertical proximity measures ([7]), and could stand for the *monitoring of vertical separation* (near the minimas).

Notice that we extracted only 6 components, thus significatively less than the 12 components (briefly) described in [6]. This might be explained by the lack, at the point of the project, of indicators related to the sector geometry.

V. NEURAL NETWORKS

A. General presentation

Artificial neural networks are algorithms inspired from the biological neurons and synaptic links. An artificial neural network is a graph, with vertices (neurons, or units) and edges (connections) between vertices. There are many types of such networks, associated to a wide range of applications: pattern recognition (see [21] and [22]), control theory,...

Beyond the similarities with the biological model, an artificial neural network may be viewed as a statistical processor, making probabilistic assumptions about data ([23]). Some *train* data is used to determine a statistical model of the process which produced this data. Once correctly trained, the neural network uses this model to make predictions on new data.

Neural networks are closely related to the Bayesian probabilities. They may be used for unsupervised learning (*density estimation* problems), and, mainly, supervised learning problems (*regression*, *classification*). Density estimation is not in the scope of our paper, so we will not detail it. The aim of *regression* is to find a statistical model producing an *output* y from *input variables* (let us denote them by x), so that the output y is as close as possible to a *target variable*, which we shall denote by t. In the case of *classification* problems, the target variables represent

class labels, and the aim is to assign each input vector x to a class.

We will use a specific class of neural networks, referred to as feed-forward networks, or multi-layer perceptrons (when the activation function is logistic). In such networks, the units (neurons) are arranged in fully-connected layers: an *input layer*, one or several *hidden layers*, and an *output layer*. Figure 1 shows an example of such a network.

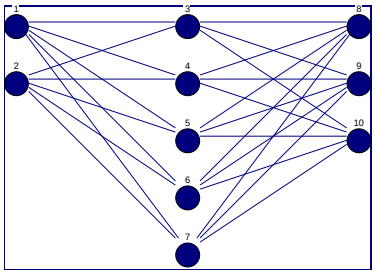

Fig. 1. Example of a feed-forward network with one hidden layer

For a network with one hidden layer, the output vector $y = (y_1, ..., y_k, ..., y_q)^T$ is expressed as a function of the input vector $x = (x_1, ..., x_i, ..., x_p)^T$ as follows:

$$y_k = \Psi(\sum_{j=1}^{q} w_{jk}\Phi(\sum_{i=1}^{p} w_{ij}x_i + w_{0j}) + w_{0k}) \qquad (1)$$

where the w_{ij} and w_{jk} are weights assigned to the connections between the input layer and the hidden layer, and between the hidden layer and the output layer, respectively, and where w_0j and w_0k are biases (or threshold values in the activation of a unit). Φ is an *activation function*, applied to the weighted output of the preceding layer (in that case, the input layer), and Ψ is a function applied, by each output unit, to the weighted sum of the activations of the hidden layer. This expression can be generalized to networks with several hidden layers.

The output error – *i.e.* the difference between the target values t and the output y computed by the network – will depend on the parameters w (weights and biases). The *training* aims at choosing these parameters, so as to minimize a chosen function of the output error.

In the case of *regression*, the minimized function is the sum of quadratic errors. For classification problems, it is best to consider a log-likelihood function. The network is then designed with one output unit per class. When the output of such an unit is 1, the input x is assigned to the class corresponding to the unit, and when the output is 0, it is not. Let us consider a problem with C classes. The log-likelihood function minimized during training is the following, known as *cross entropy*:

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{C} t_k^{(n)} ln(y_k^{(n)}) \qquad (2)$$

where $t^{(n)}$ and $y^{(n)}$ are the n^{th} target and output vectors, respectively.

Several optimization methods may be used to minimize $E(w)$, when training the network. Let us cite *backpropagation*, which consists in successive modifications of the weights assigned to the connections between the layers, starting with the output layer. These modifications take account of the relative importance of each weight in the output error variations. Other local optimizations using the gradient of the error (BFGS, conjugate gradients, for example) are also widely used. A variety of global optimization methods (simulated annealing, evolutionary algorithms) are also proposed in the literature. These global methods usually perform better than the local methods when there are many local minima for the error function, but they are generally much slower.

B. Neural networks applied to our problem

For our problem, we have chosen three-layers feed-forward networks, denoted $I_\alpha H_\beta O_\gamma$ in the rest of the paper, with α units in the input layer, β units in the hidden layer, and γ units in the output layer. The input variables are normalized, by substracting the mean value and dividing by the standard deviation.

There are many possible choices for the functions Ψ and Φ, depending on the problem being adressed. A common choice for Φ is the logistic function :

$$\Phi(z) = \frac{1}{1 + e^{-z}} \qquad (3)$$

This is the activation function that was used in our experiments. As we adress a classification problem – assign each input vector (a list of complexity indicators values) to a class representing the sector status (*merged*, *armed*, or *split*) – we have chosen to minimize the *cross entropy* function. Therefore, the transfer function Ψ applied to the output layer, must be the *softmax* function:

$$\Psi(z_k) = \frac{e^{z_k}}{\sum_{m=1}^{C} e^{z_m}} \qquad (4)$$

The `nnet` package of the `R` language was used (see `http://www.r-project.org/` for details on the `R` language and environnment). In this package, developped by Pr B. D. Ripley, a quasi-newton minimization method (BFGS) is used for the network's learning. The parameters of the `nnet` tool are the range parameter (default 0.5), defining the range into

which the initial random weights are chosen, the parameter for weight decay (default 0), and the maximum number of iterations. The training stops either if the fit criterion (the *cross entropy* in our case) falls below a chosen parameter $abstol$ (default 1.0e-4), or if the improvement of the fit criterion is less than $1 - reltol$ (the default value for $reltol$ is 1.0e-8).

Several combinations of air traffic complexity indicators, or of principal components, will be tested. The number of input units of the network will be chosen equal to the cardinal of the evaluated set of indicators (or components). The number of hidden units is 15 (this choice is discussed later). The output layer is made of three units, one for each class (*merged*, *armed*, or *split*).

So a target vector $t^{(n)}$ with value $(1, 0, 0)$ means that the considered sector was merged with other sectors when the n^{th} measure of the vector of complexity indicators was made. Armed sectors will be represented by $(0, 1, 0)$. A value of $(0, 0, 1)$ will mean that the sector was previously split in two or more sectors at the time x was measured. Of course, the actual output of the neural network will not be exact values 0 or 1. It will be triples (a, b, c) of floating-point values between 0 and 1, each value being the probability to belong to a class. The input vector $x^{(n)}$ will be assigned to the class of highest probability.

C. Evaluation of the neural network's outputs

A well-known problem, when using neural networks (or other regression methods), is *overfitting*: with enough parameters and enough training cycles, it is always possible to find a good fit for a given data set. So one may find a perfect fit for a chosen data sample, and then feel disappointed when the trained network makes wrong predictions on fresh data. So, we will systematically proceed as follows: train the network on a randomly chosen data sample (called *train*), then check the results, first on the same data sample, and second on a fresh data sample (called *test*), that was not used for the training.

In order to evaluate the outputs of several different models, we have to compare the neural networks predictions to the actual target values. We may use the fit criterion (*cross entropy*) but it does not reflect the influence of the number of weights (and biases) in the neural network. It is known (see [23]) that a network with too few weights may not be able to capture all the variations of the response to the input x, whereas a network with too many weights will more likely be subject to *overfitting*. In the next sections, we will compare several sets of input variables, of various sizes. Consequently, *the number of weights in the network will not remain constant*, and this variation will bias the results.

We will therefore use the *Akaike information criterion* ([24]): $AIC = 2\lambda - 2ln(L)$, where λ is the number of unadjusted parameters of the model (i.e. the number of weights and biases of the network), and $ln(L)$ is the log-likelihood. This criterion is strongly related to information theory, and more specifically to the Kullback-Leibler distance (K-L) between a candidate model and the "true" model. In our case, the AIC is written as follows:

$$AIC = 2\lambda - 2\sum_{n=1}^{N}\sum_{k=1}^{C} t_k^{(n)} ln(y_k^{(n)}) \qquad (5)$$

One should be aware that AIC is a *relative* criterion, which can only be used to compare a set of candidate models relative to a same "true" model: as this true model is unknown, the corresponding term in the K-L distance was considered as a constant and dropped, in the AIC. As we would like to compare predictions made on the *train* test and on the *test* set, which are of different size, we will divide the AIC by N, the number of data items, and use: $AIC_{avg} = \frac{AIC}{N}$.

In addition to the numerical results provided by the *Akaike information criterion*, we shall also consider the global proportion of correctly classified input vectors, and also the percentage of correct classifications for each class. One must be aware, however, that the rate of correct classifications *is not* the criterion being maximized by the neural network, so we should remain cautious when comparing the different classification rates. However, these percentages are easily understandable and may allow us to make some interesting statements on the results.

VI. Results

A. Preliminary results, discussion on the parameters

Before making statements about how significant the indicators (or combinations of indicators) are, let us first check if the chosen network is efficient on our problem, and consider how to choose the network and training parameters.

Figure 2 shows the evolution of the cross entropy criterion during the network's training. The input variables were the six main components found by the PCA, and also the sector's volume, so the input layer of the network had 7 units. We have chosen a hidden layer with 15 units. The output layer had 3 units, each one representing a class, as explained in section V. The maximum number of cycles was set to 1500, but the training actually stopped at 1010 cycles, because the algorithm was unable to significantly improve the cross entropy criterion.

Several questions may arise, concerning the parameters choices, and their influence on the results. In this

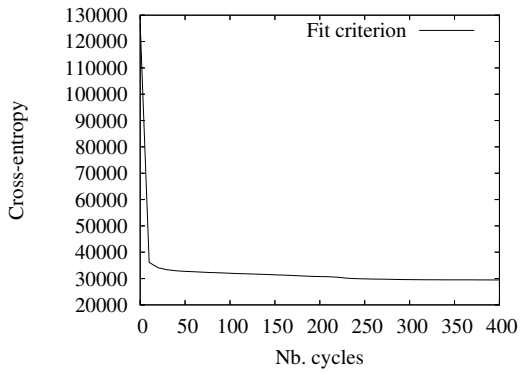

Fig. 2. Evolution of the cross entropy criterion with the number of training cycles, using the main 6 components and the volume as input (network $I_2H_{15}O_3$)

study, we will use the default values for the following `nnet` parameters: the stop criteria $abstol$ and $reltol$, and the weight decay parameter, set to $decay = 0$. The range in which the network's weights and biases are initialized depends on the range of the input variables, that we have normalized. A parametric study with all components plus the sector volume as input and with 15 units in the hidden layer shows that the best results, averaged on two runs for each value of the range parameter, were obtained with a range value of 0.4. So we shall use 0.4 for the `nnet` range parameter.

It may have been useful to make another parametric study, trying to find the optimum number of hidden units by minimizing the AIC, but this is rather time-consuming (it should be repeated for each set of input variables). So we have chosen, after a few trials, a hidden layer of 15 units.

B. Model selection

Let us now select the best model among several candidate models. A *model* is a trained neural network and set of input variables that we expect to provide a good explanation of the sector status (*merged*, *armed*, or *split*). We use the PCA components plus the sector volume as input variables. The sector volume was not analyzed in the PCA, but as we have not implemeted any indicator using the sector geometry, we will use the volume as a proxy for metrics such as "space available around conflicts" or "distance to sector boundary".

An iterative approach is used: we shall first use component C_1 (representative of the number of aircraft) as input to the neural network, then add the volume, and continue with the five other main components, successively added in the order found by the PCA. At last, we will use all the 27 components, and the sector volume, as input. The AIC_{avg} criterion (see section V-C) is used to select the best model.

For each set of input variables, five training runs are made. The reason for this is the following: the training method of `nnet` is a local optimization method which starts at a randomly chosen point (the initial weights), and which follows the steepest descent of the error function being optimized. This error function may have several local minima, so choosing different initial weight values may lead to different results. Although these local minima are often fairly close, several runs will comfort our results.

Figure 3 shows, for *train* and *test* data, how the AIC_{avg} criterion evolves when adding components to the set of input variables. The mean values, averaged on the five runs, are presented. This figure should be interpreted as follows: when the AIC significantly decreases when introducing a new variable in the model, this means that this variable improves the prediction of the sector status. When the AIC increases or remains constant, this means that the benefit provided by the additional variables is offset by the complexity it implies on the model (increase in the number of parameters)[3].

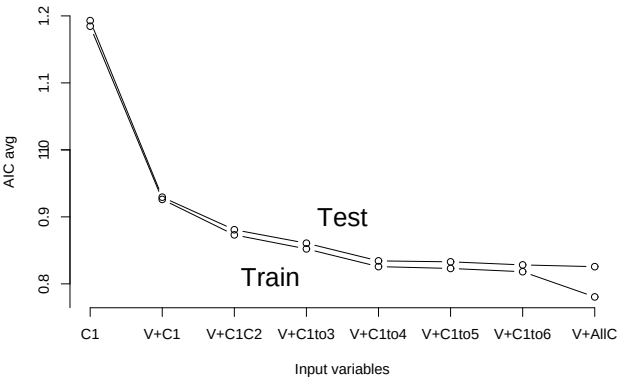

Fig. 3. Values of the AIC_{avg} criterion for the different sets of input variables

The two curves, *train* and *test*, are fairly close, except at the last point ($V + AllC$), which corresponds to the model $\{V; C_1; ...; C_{27}\}$. For this last point, the AIC criterion is improved on *train* data, whereas it remains nearly the same on *test* data, which shows a little overfitting of the neural network on *train* data. Otherwise, the results on *train* and *test* data are quite consistent, so the neural networks are able to generalize efficiently on fresh data.

The AIC criterion is significantly improved when adding the *sector volume* to the component C_1 (*aircraft count*). Component C_2, which is mostly related the

[3]We could have used the Schwartz BIC criterion instead of the AIC. As it assigns a heavier penalty to additionnal parameters, it may have been easier to interpret.

speed variance and the *aircraft vertical evolutions* also improves the criterion, although much less than the sector volume. Small improvements are brought by component C_3 (*incoming flows*), and also by C_4, mostly related to the *converging flows* and the *conflict anticipation*. The components C_5 (*diverging flows*) and C_6 (*monitoring of vertical proximity*) bring no significant improvement. The use of the other components does not improve the prediction, as shown by the last point of the *test* curve.

It might look surprising that the aircraft proximity (horizontal or vertical) is of little influence on the sector status explanation. But, as was already stated in [7], the aircraft have already been separated (in the other dimension) before the proximity situation occurs, which explains why it is not very useful. The *anticipation* of future aircraft proximity (component C_4), is more significant.

C. Classification rates

Tables III and IV give the proportions of correct classifications made by the neural networks (the best of the five runs), using *train* data as input, or *test* data, respectively. The second column of each table shows the global rate of correct classifications. The three last columns detail the results for each class (*merged*, *armed*, or *split*). As previously stated on figure 3, the results on *test* data are quite consistent with the results on *train* data.

Set	Global	Merged	Armed	Split
$\{C_1\}$	72.91%	81.92%	44.90%	85.61%
$\{V; C_1\}$	79.34%	86.26%	59.01%	87.89%
$\{V; C_1; C_2\}$	80.70%	86.93%	60.79%	90.04%
$\{V; C_1; C_2; C_3\}$	80.84%	87.32%	60.23%	90.43%
$\{V; C_1; ...; C_4\}$	82.03%	87.87%	63.02%	91.15%
$\{V; C_1; ...; C_5\}$	81.80%	88.23%	61.77%	90.88%
$\{V; C_1; ...; C_6\}$	81.83%	88.09%	62.16%	90.85%
$\{V; C_1; ...; C_{27}\}$	83.36%	88.63%	65.61%	92.19%

TABLE III
CORRECT CLASSIFICATIONS ON TRAIN DATA

Set	Global	Merged	Armed	Split
$\{C_1\}$	72.61%	81.64%	44.41%	85.40%
$\{V; C_1\}$	79.12%	85.56%	59.14%	88.12%
$\{V; C_1; C_2\}$	80.48%	86.58%	60.74%	89.82%
$\{V; C_1; C_2; C_3\}$	80.42%	86.71%	60.04%	90.08%
$\{V; C_1; ...; C_4\}$	81.82%	87.78%	62.77%	90.74%
$\{V; C_1; ...; C_5\}$	81.59%	88.19%	61.44%	90.49%
$\{V; C_1; ...; C_6\}$	81.65%	87.98%	61.88%	90.64%
$\{V; C_1; ...; C_{27}\}$	82.67%	88.04%	64.51%	91.68%

TABLE IV
CORRECT CLASSIFICATIONS ON TEST DATA

When using only the component C_1 (*aircraft count*) as input, the neural network already makes more then 72% of correct predictions. However, the rate for the *armed* class is less than 45%, which is not very good. Once again, we see that the use of the sector volume (denoted V) greatly improves the prediction of the sector status, with more than 79% of correct classifications, globally, and a rate of about 59% for the *armed* class. The classification rates when adding C_2 climb to 81% for the global rate, and about 60% for the *armed* class. Adding C_3 does not increase the rates, although it improved the AIC. C_4 slightly improves the results, whereas C_5 and C_6 provide no improvement. At most, when considering the sector volume and the 4 first components, we have about 82% of correct classifications, and about 63% of correct classifications for the *armed* class.

The last line of both tables, with all components and the volume, shows the best classification rates, whereas the AIC showed no improvement in the same case. The apparently good results of the last line are certainly due to the higher number of network's parameters than in the previous models. They do not mean that the model with all components is the best one.

In order to verify this assumption, we have tried another network for the $\{V; C_1; ...; C_4\}$ model, with about the same number of parameters than the $\{V; C_1; ...; C_{27}\}$ model. This last model was assessed with a network $I_{28}H_{15}O_3$, where there were 483 weights and biases. Let us take a $I_5H_{53}O_3$ network, with 53 units in the hidden layer, for the $\{V; C_1; ...; C_4\}$ model. This network has 480 parameters. Table V shows that the classification results, with this network, are better than the ones obtained with all components and the volume in table III and IV.

Set	Global	Merged	Armed	Split
$\{V; C_1; ...; C_4\}_{train}$	84.34%	89.74%	67.61%	91.88%
$\{V; C_1; ...; C_4\}_{test}$	83.84%	89.32%	67.26%	91.09%

TABLE V
CORRECT CLASSIFICATION RATES FOR THE $\{V; C_1; ...; C_4\}$ MODEL, WITH A $I_5H_{53}O_3$ NETWORK

When considering the detailed results, for each class, on the three tables, we see that the *merged* and *split* classes have better classification rates than the *armed* class. This is not a surprise, as the cloud of points representing the measures of the *armed class*, in the variables space, is located "between" the clouds representing the two other classes. The neural network aims at finding the frontiers between these clouds, and this is more difficult with two frontiers instead of one.

To conclude this section, we have seen that the best model is able to classify correctly about 84% of the input vectors. We may still slightly improve these results by running a parametric study on the number

of hidden units, for this model, still minimizing the AIC criterion. But this is rather time-consuming, and our results are already quite good, considering that our data is rather noisy. Let us remind that we have assumed that the decisions to split or to merge sectors had a single cause, that is the workload. This may not always be the case in reality, where there could be other reasons: controllers training, hardware failures, and so on. In a next step of our research, we may try to improve our results by filtering the data – at least the sector *split* decisions – using an approximation of the workload (the number of aircraft for example).

D. Discussion on the results, comparison with other works

To summarize the previous results, the AIC criterion (figure3) allowed us to select the best model among the ones we have tested. This model uses a subset of only 5 input variables $\{V; C_1; ...; C_4\}$, among the 28 that we have considered.

The proportions of correct classifications (tables III and IV) show that the use of the sector volume significantly improves the network's prediction, by 6.5%, when compared to the sole *aircraft count* C_1. There is a relatively small improvement (around 3%), between the *sector volume* + *aircraft count* model and the best model. The other components bring no significant improvement.

Other works, like [7] and [6] for example, have already stated that the *aircraft count* model is not very efficient in predicting the controller's workload. In these studies, the improvement brought by other complexity indicators was much higher than what is shown in our results. There may be several explanations to this.

In [7], Chatterji and Sridhar show workload prediction rates (on test data) ranging from less than 16% to 54%, for their *medium* workload class, and from 0% to 100% for their *high* workload class. However, their results with *test* data were not consistent with the ones obtained with *train* data. They honestly state that they were unable to draw reasonable conclusions from these results, as they had too few measures in the medium and high workload classes.

In [6], Kopardekar and Magyarits apply a linear regression on subjective complexity ratings. The results were not given in percentages of correct classifications. The R^2 criterion of the regression was used to compare the candidate models. The results presented in [6] show some significant differences of R^2 values between the *aircraft count* model and the model based on *dynamic density* (which is viewed as a linear combination of several indicators). Kopardekar and Magyarits also lacked measures, but for very low and high workload traffic.

Both studies ([7], [6]) use subjective complexity ratings, provided by air traffic controllers who assessed the traffic complexity of several traffic samples, only during periods when the sector was armed[4]. In our study, we use the actual sector status, recorded for several elementary sectors and groups of sectors, assuming that the sector status is related to the controllers workload. So we have only three levels of workload: *low*, when the sector is merged with other sectors, *normal use*, when the sector is armed, and *too high* when the sector has been split into smaller sectors. We have the feeling that the studies [7] and [6] focus on the *normal use* load interval, and that the use of complexity ratings within this domain magnifies the observed phenomenons.

Aside from these considerations, another, and more straightforward, explanation of the relatively small differences between our candidate models – when comparing the classification rates – resides in the nature of the data used in our study. We simply have a great number of measures for which there is no doubt as to which class they belong to, even when using the worst possible model. For example, when there are 2 aircraft in a given sector, it is most likely merged with other sectors. On the opposite, when there are 60 aircraft in a sector, it has certainly been split into several sectors. We may have exhibited much higher variations in the results, by computing the classification rates using only measures collected around the times when the sector configurations changed.

VII. Conclusion

In conclusion, we were able to select the best model, among several candidate models, establishing a functional relationship (equation V-A with the weights of the trained network) between the air traffic complexity indicators and the sector status. Assuming that the decisions to merge or to split a sector are statistically related to the controllers workload, this original method provides an objective way to validate the complexity metrics. Our method also has the advantage, in comparison to other methods, to use widely available data (sector configurations and radar tracks recorded by the ATC centers), noticeably cheap to collect, as the active participation of air traffic controllers is not required.

Neural networks, minimizing the *cross entropy* function of the output error, showed good results, consistent on *train* and *test* data. The *Akaike information criterion* proved useful in selecting the best model, avoiding the bias due to the different number of parameters in the candidate models. In the iterative approach that

[4]This is not explicitly stated in the papers, but we assume so, as they lacked measures for high workload traffic situation.

was used, the highest improvement, when comparing to the *aircraft count* model, was brought when introducing the *sector volume* as a new input variable. Smaller improvements were provided by components C_2 (*speed variance* and *aircraft vertical evolutions*), C_3 (*incoming flows*), and C_4 (*converging flows* and *conflict anticipation*).

So far, we have only considered the PCA components in our study. The next step of our research may be to select a subset of individual indicators, issued from the components of the best model, and re-iterate the approach presented in this paper. This would provide a more direct and simple relationship between the indicators and the sector status, by avoiding to compute the components from all the indicators.

The neural network approach used in this study seems appropriate for the granularity we are interested in and the foreseen applications, either strategical (sector design) or pre-tactical (sector planning). We are also fairly confident that decisions to split or merge sectors may allow to assess the instantaneous workload as well, and could therefore be used to improve tactical tools (PRESAGE). To this end, other statistical methods should be investigated to take into account the serial correlation of sector status, looking closely at the sector splitting times. We plan to tackle this issue in a close future, using dynamic discrete choice models.

Finally, another issue that we intend to address, in relation to the complexity indicators, is the prediction of optimal sector configurations. Previous works ([9], [10]) proposed several algorithms to compute optimal sector configurations, using sector capacities and *incoming flows*. The output of the neural network is a triple of probabilities, allowing to decide when a sector should be split, or merged. We may derive a realistic workload indicator – and also threshold values – from these probabilities, which could be used to compute optimal sector configurations.

REFERENCES

[1] B. Hilburn and G. Flynn. Toward a non-linear approach to modeling air traffic complexity. In *2nd Human Performance Situation Awareness and Automation Conference*, 2004.

[2] I. V. Laudeman, S. G. Shelden, R. Branstrom, and C. L. Brasil. Dynamic density: An air traffic management metric. Technical report, 1999.

[3] A. Majumdar, W. Y. Ochieng, G. McAuley, J.M. Lenzi, and C. Lepadetu. The factors affecting airspace capacity in europe: A framework methodology based on cross sectional time-series analysis using simulated controller workload data. In *Proceedings of the 6th USA/Europe Air Traffic Management R & D Seminar*, 2005.

[4] J.H. Crump. Review of stress in air traffic control: Its measurement and effects. *Aviation, Space and Environmental Medecice*, 1979.

[5] P. Averty, S. Athènes, C. Collet, and A. Dittmar. Evaluating a new index of mental workload in real atc situation using psychological measures. Note cena nr02-763, CENA, 2002.

[6] P. Kopardekar and S. Magyarits. Measurement and prediction of dynamic density. In *Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar*, 2003.

[7] G.B. Chatterji and B. Sridhar. Measures for air traffic controller workload prediction. In *Proceedings of the First AIAA Aircraft Technology, Integration, and Operations Forum*, 2001.

[8] C. Mannings, S. Mill, C. Fox, E. Pfleiderer, and H. Mogilka. The relationship between air traffic control events and measures of controller taskload and workload. In *Proceedings of the 4th Air Traffic Management Research & Developpment Seminar*, 2001.

[9] D. Gianazza and J. M. Alliot. Optimization of air traffic control sector configurations using tree search methods and genetic algorithms. In *Proceedings of the 21st Digital Avionics Systems Conference*, 2002.

[10] D. Gianazza, J. M. Alliot, and G. Granger. Optimal combinations of air traffic control sectors using classical and stochastic methods. In *Proceedings of the 2002 International Conference on Artificial Intelligence*, 2002.

[11] D. Gianazza. *Optimisation des flux de trafic aérien*. PhD thesis, Institut National Polytechnique de Toulouse, 2004.

[12] G. M. Flynn, C. Leleu, and L. Zerrouki. Traffic complexity indicators and sector typology analysis of u.s. and european centres. Technical report, Eurocontrol, 2003.

[13] Note de synthèse sur l'indicateur de complexité pru. Technical report, DTI/SDER (ex CENA), 2005.

[14] Cognitive complexity in air traffic control, a litterature review. Technical report, Eurocontrol experimental centre, 2004.

[15] K. Guittet and D. Gianazza. Analyse descriptive des indicateurs de complexité du trafic aérien à partir des données image et courage. Note nr05-905, DSNA/DTI/SDER, Décembre 2005.

[16] B. Sridhar, K. S. Sheth, and S. Grabbe. Airspace complexity and its application in air traffic management. In *Proceedings of the 2nd USA/Europe Air trafic Management R&D Seminar*.

[17] P. Kopardekar. Dynamic density: A review of proposed variables. Faa wjhtc internal document. overall conclusions and recommendations, Federal Aviation Administration, 2000.

[18] D. Delahaye and S. Puechmorel. Air traffic complexity: towards intrinsic metrics. In *Proceedings of the third USA/Europe Air Traffic Management R & D Seminar*, 2000.

[19] P. Averty. Conflit perception by atcs admits doubt but not inconsistency. In *Proceedings of the 6th Air Traffic Management Research & Developpment Seminar*, 2005.

[20] P. Averty, K. Guittet, and P. Lezaud. *Work in progress, presented at an internal SDER seminar*. Technical report, DTI/SDER (former CENA), 2005.

[21] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1996. ISBN: 0-198-53864-2.

[22] B. D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996. ISBN: 0-521-46086-7.

[23] M. I. Jordan and C. Bishop. *Neural Networks*. CRC Press, 1997.

[24] W. H. Greene. *Econometrics analysis*. Prentice Hall International. ISBN: 0-13-015679-5.

[25] F. Chatton. Etudes de nouvelles métriques de complexité de la circulation aérienne. Master's thesis, Ecole Nationale de l'Aviation Civile (ENAC), 2001.

APPENDIX: COMPLEXITY METRICS

Delahaye and Puechmorel metrics

To present the geometrical indicators introduced in [18], we need to define several quantities:

- The vector representing the distance between two aircraft is denoted by $\overrightarrow{X_iX_j}$ where X_i (resp. X_j) stands for the location of aircraft i (resp. j).
- The "oblical" distance between two aircraft (i and j) is denoted by

$$d_{ij}^{ob} = \sqrt{< \overrightarrow{X_iX_j}, \overrightarrow{X_iX_j} >}\,, \qquad (6)$$

where $< ., . >$ stands for the appropriate scalar product.
- We denote by $\overrightarrow{v}_{ij} = \overrightarrow{v}_j - \overrightarrow{v}_i$ the speed difference between two aircraft.
- The derivative of the "oblical" distance between two aircraft is denoted by v_{ij} and writes

$$v_{ij} = \frac{< \overrightarrow{X_i X_j}, \overrightarrow{v}_{ij} >}{d_{ij}^{ob}} \ . \tag{7}$$

- We introduce a weighting function f. As suggested in [25], we used

$$f(d_{ij}^{ob}) = \frac{e^{-\alpha (d_{ij}^{ob})^2} + e^{-\beta d_{ij}^{ob}}}{2} \ , \tag{8}$$

with $\alpha = 0.002$, $\beta = 0.01$, the d^{ob} being expressed in nautical miles.

These indicators are defined pointwise. To get a value on the controlled airspace, they have to be averaged on the different aircraft. In [18], a density indicator is defined as follows

$$\textit{Dens}(i) = \sum_{j=1}^{N} b f(d_{ij}^{ob}) \ . \tag{9}$$

Two indicators are introduced to reflect the variability in headings (*track_disorder*) and speed (*speed_disorder*). There are defined as

$$\textit{track\_disorder}(i) = \sum_{j \neq i} |\theta_i - \theta_j| f(d_{ij}^{ob}) \ . \tag{10}$$

$$\textit{speed\_disorder}(i) = \sum_{j \neq i} \|\overrightarrow{v}_{ij}\| f(d_{ij}^{ob}) \ . \tag{11}$$

Indicators *Div* et *Conv* respectively describe convergency and divergency of the aircraft in the controlled sector.

$$\textit{Div}(i) = \sum_{\substack{j=1 \\ j \neq i}}^{Nb} 1_{\mathbb{R}^-}(v_{ij}) \, . \, |v_{ij}| \, f(d_{ij}^{ob}) \ , \tag{12}$$

$$\textit{Conv}(i) = \sum_{\substack{j=1 \\ j \neq i}}^{Nb} 1_{\mathbb{R}^+}(v_{ij}) \, . \, |v_{ij}| \, f(d_{ij}^{ob}) \ . \tag{13}$$

Indicators Sd_+ and Sd_- are designed to set a weight on potential conflicts that are difficult to solve. These "sensitivity" indicators are defined by

$$Sd_-(i) = \sum_{\substack{j=1 \\ j \neq i}}^{Nb} 1_{\mathbb{R}^-}(v_{ij}) \, \|\overrightarrow{\nabla} v_{ij}\| f(d_{ij}) \ , \tag{14}$$

$$Sd_+(i) = \sum_{\substack{j=1 \\ j \neq i}}^{Nb} 1_{\mathbb{R}^+}(v_{ij}) \, \|\overrightarrow{\nabla} v_{ij}\| f(d_{ij}) \ . \tag{15}$$

Note that components of the gradient are weighted so as to reflect the difficulty of the respective manoeuvers[5]. As observed in [18], a situation with a high "sensitivity" is easier to resolve for the air controller than one with a low "sensitivity". As these indicators "increase" with the number of aircraft, it is unclear whether they actually are "complexity" or "simplicity" indicators. We thus define a last pair of indicators, *insen_c* and *insen_d* as

$$\textit{insen\_c} = \frac{\textit{Conv}^2}{Sd_+} \quad \text{and} \quad \textit{insen\_d} = \frac{\textit{Div}^2}{Sd_-} \ . \tag{16}$$

Modified PRU metrics

The work conducted by SDER-RFM for the Performance Research Unit (citeRFM), though initially designed to compare ATC centers on a daily basis, inspired the following indicators :

- *inter_hori*: number of potential crossings (irrespective of the aircraft direction on their trajectories) with angle greater than 20 degrees.
- *inter_vert*: denote by n_1, n_2 et n_3 the numbers of stable/climbing/descending aircraft. The indicator is then defined as

$$\texttt{inter\_vert} = \frac{(n_1 n_2 + n_2 n_3 + n_1 n_3)}{(n_1 + n_2 + n_3)} \ . \tag{17}$$

- *avg_vs*: this is simply the average vertical speed of controlled aircraft.

Metrics inspired from the CREED project

The work of P. Averty on conflict detection [19] inspired a set of indicators. One of the ideas in [19] is that conflict perception is "plannar". The author thus defines for converging pairs of aircraft the following quantities

- *Ed* : minimum horizontal distance between aircraft.
- *Efl* : horizontal distance when the aircraft are vertically separated (after the crossing).
- *Da* : the "anticipation degree", i.e. the distance between the faster aircraft and the intersection of the aircraft trajectories (in the horizontal plan). We replace this variable to a modified *Da*, *DaC*, which stands for the greater distance between one of the aircraft and the point where, horizontally, the distance between aircraft is the smallest. For explanations about this substitution, we refer to [15].

Originally, these quantities are defined to describe conflict perception. To translate the idea of [19] in terms of traffic complexity, we assume that a conflict is all the more critic that the expected separation (*Ed* and *Efl*) and the anticipation (*DaC*) are small. We thus set

$$\texttt{creed} = \frac{1}{\alpha \ \texttt{Da} + (1-\alpha)(\beta \ \texttt{Ed} + (1-\beta) \ \texttt{Efl})} \ , \tag{18}$$

where α and β are parameters in $[0; 1]$[6]. Finally, aircraft pairs considered in [19] are such that vertical separation occurs prior to separation, as the converse situation is avoided as much as possible by controllers. Accordingly, the complexity associated with these latter pairs is likely to be greater and we distinguished the two kind of conflicts by summing the quantity fintroduced in (18) on both sets of aircraft, thus creating two distinct indicators, *creed_ok* ("good pairs") and *creed_pb* ("bad pairs").

[5]Reasonable weights were given by P. Averty and M. Tognoni.

[6]As for now, these parameters are set equal to 0.5, but are meant to be adjusted and possibly vary with *DaC* to reflect the results of ongoing research [20].