

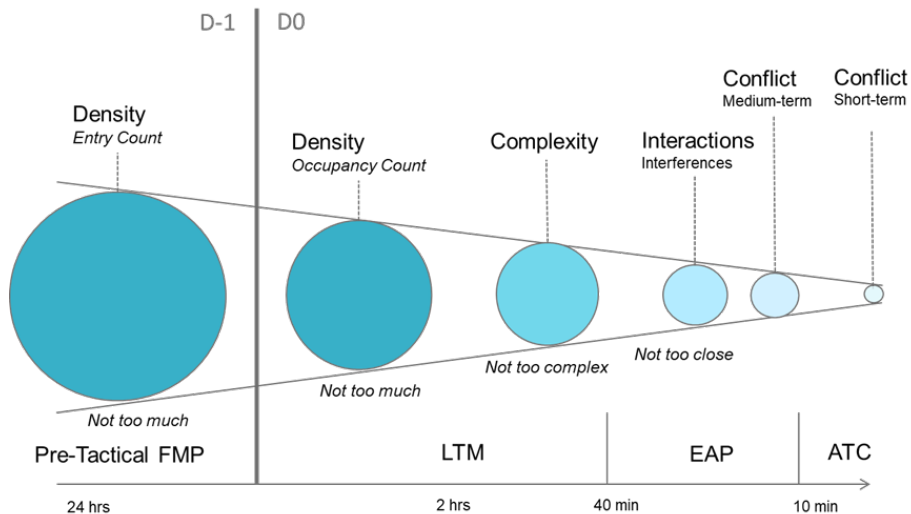
# Modélisation et Résolution des Interactions au sein de l'Espace Aérien par Intelligence Artificielle

Loïc Shi-Garrier

Encadré par

Daniel Delahaye (ENAC) et Nidhal C. Bouaynaya (Rowan University)

20 juin 2022



Extended ATC Planning : concept opérationnel visant à **lier** responsabilités **stratégiques** (équilibrer la demande avec la capacité) et **tactiques** (séparer les avions).

# Extended ATC Planning

Le principal objectif de la fonction EAP est de lier les responsabilités ATFCM et ATC en s'appuyant sur :

- Des outils d'**automatisation**
- Une amélioration de la **communication** entre FMPiste et contrôleur
- Une fourniture d'information pour **aider à la prise de mesures anticipées** avant que le trafic n'entre dans les secteurs surchargés

La fonction EAP vise à réduire les délais, réduire le nombre de régulations ATFCM, et améliorer la sécurité.

Objectif n°1 : utiliser l'**apprentissage machine** pour **prédire** la **complexité** de l'espace aérien.

→ **Predicting Air Traffic Congested Areas with Long Short-Term Memory Networks**. 14th USA/Europe ATM R&D Seminar, Sep 2021.

Prédire la congestion de l'espace aérien dans un horizon temporel de 40 minutes en combinant :

- Des méthodes de **prédiction de flux de trafic aérien** (ATFP)
- Une définition de la congestion fondée sur une **métrique de complexité intrinsèque**
- Des techniques d'**apprentissage machine** (Encodeur-Décodeur LSTM)

Il existe trois méthodes pour la **prédiction de la congestion** :

- Outils de **détection de conflits** : nécessite une incertitude suffisamment faible sur les trajectoires
- **Air Traffic Flow Prediction** : s'appuie sur des modèles agrégés plutôt que sur des trajectoires individuelles. Les implémentations les plus récentes reposent sur l'apprentissage machine pour prédire le nombre d'avions.
- **Métriques de complexité**

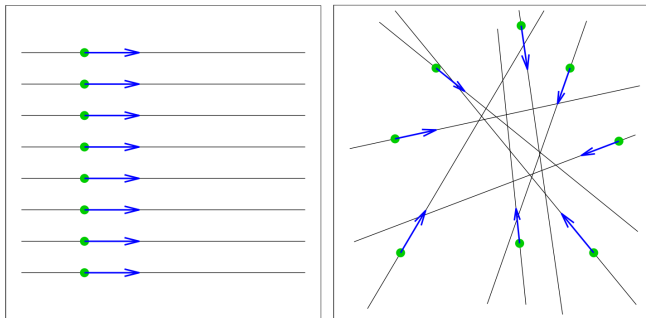


Figure 1: Complexités différentes pour un même nombre d'avions

## Métrique de complexité

La **congestion** est définie comme une situation dans laquelle un ensemble de trajectoires interagissent fortement dans une zone donnée et sur une durée donnée, conduisant à de potentiels conflits et nécessitant une surveillance accrue de la part des contrôleurs.

La **complexité** du trafic aérien est une mesure de la difficulté qu'une situation de trafic donnée présente à un système de contrôle aérien.

# Métrique de complexité

La **congestion** est définie comme une situation dans laquelle un ensemble de trajectoires interagissent fortement dans une zone donnée et sur une durée donnée, conduisant à de potentiels conflits et nécessitant une surveillance accrue de la part des contrôleurs.

La **complexité** du trafic aérien est une mesure de la difficulté qu'une situation de trafic donnée présente à un système de contrôle aérien.

Objectif: quantifier la "complexité" d'une situation de trafic

La complexité est liée :

- A la charge de travail du contrôleur
  - A la probabilité d'occurrence d'un conflit
  - A la **géométrie du trafic** (convergence/divergence) et de l'espace aérien
- ▶ Nous utilisons une métrique de complexité basée sur **les systèmes dynamiques linéaires**

## Modèles séquentiels

- Les **réseaux de neurones récurrents** sont bien adaptés au traitement de séries temporelles de longueur variable, par exemple des trajectoires
- Cellule RNN basique:  $y_t = \phi(W \cdot [x_t^T; y_{t-1}^T]^T + b)$
- Nombreuses variantes, notamment le réseau **Long Short-Term Memory (LSTM)**

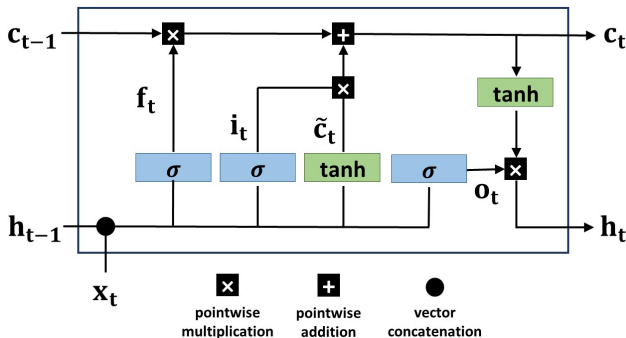


Figure 2: LSTM



# Données

Une trajectoire est une suite d'états de la forme :

$\text{état}_i = [\text{date}, \text{aircraft ID}, \text{latitude}, \text{longitude}, \text{altitude}, \text{vitesse sol}, \text{cap}, \text{taux de montée}]$ .

- 8,011 trajectoires simulées
- 3,025 pas de temps (1 pas de temps = 15s)
- 2,434 suites de longueur  $t_{in} = 160$  pour l'entraînement
- 271 suites de longueur  $t_{pred} = 160$  pour la validation

On définit une **tâche de régression par apprentissage supervisé**

# Réseau Encodeur-Décodeur

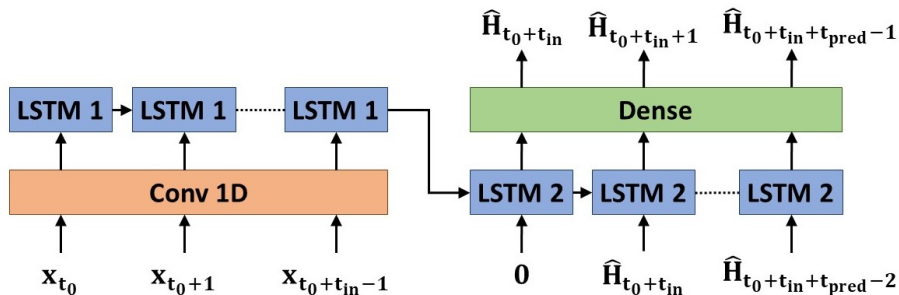
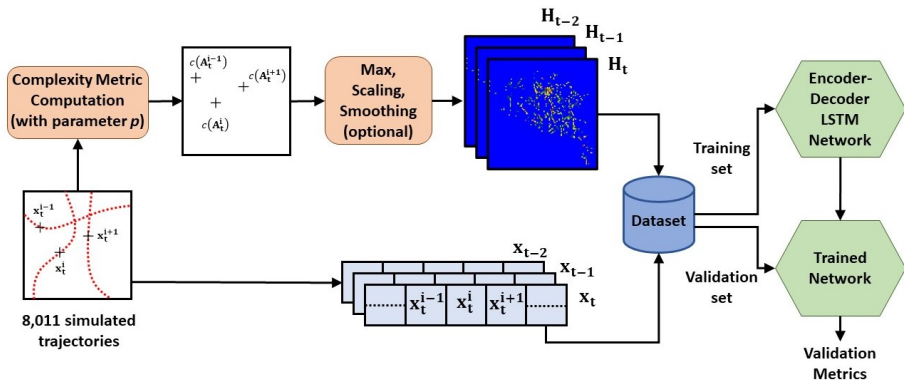


Figure 3: Modèle Encodeur-Décodeur avec "teacher forcing"

# Méthodologie complète



## Exemple de résultat

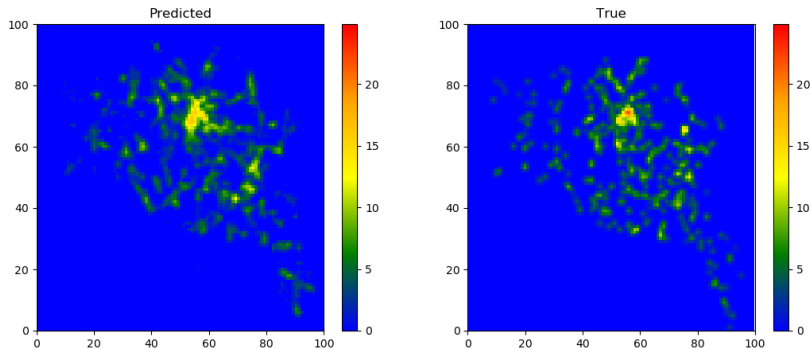


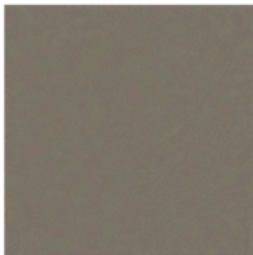
Figure 4: Un exemple issu du jeu de validation (avec données lissées). A gauche : la matrice prédite  $\hat{H}_t$ . A droite : la vraie matrice  $H_t$ .

Problème : l'apprentissage machine n'est pas **robuste** (au bruit, aux attaques, à des données issues d'une autre distribution ...). Le modèle est incapable de **quantifier son incertitude**.



stop sign  
Confidence: 0.9153

+



Adversarial perturbation

=

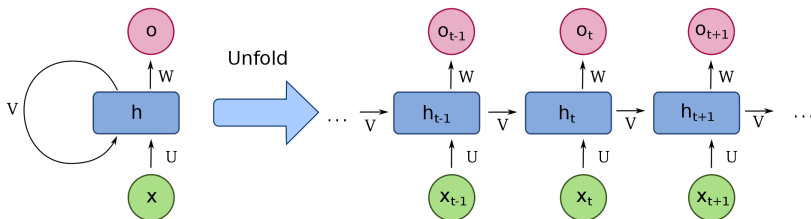


flowerpot  
Confidence: 0.8374

→ **Excès de confiance**, donc problème de **fiabilité** et d'**acceptabilité** auprès de l'utilisateur final, en particulier dans le cadre de **systèmes critiques** comme le contrôle aérien.

Objets d'étude : trajectoires, plans de vol, flux etc. → **séries temporelles**.

Objectif n°2 : étudier la **robustesse** des modèles d'apprentissage machine manipulant des séries temporelles, e.g., **réseaux de neurones récurrents**, **transformers**.



### Première approche : **Réseaux de neurones bayésiens**

- Les **paramètres** du modèle sont des **distributions de probabilités**.
- Les **distributions a posteriori** ne peuvent pas être calculées exactement → Variational Inference, Markov Chain Monte Carlo etc.

### Seconde approche : **Méthodes probabilistes non bayésiennes**

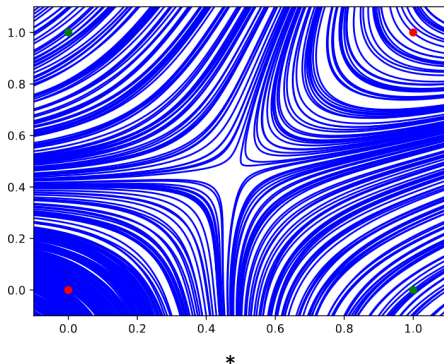
- Méthodes ensemblistes: plusieurs modèles sont entraînés.
- Autres méthodes fondées sur l'optimisation robuste.

### Troisième voie : **Approche géométrique**

- Suivre une distribution de probabilité à travers le modèle est difficile.
- En revanche, suivre la **géométrie** des distributions est plus facile.
- La **sortie** du modèle est vue comme une **variable aléatoire**  $y$  dont la distribution appartient à une **famille paramétrique**  $p(y|\theta)$ .
- Cette famille est munie de la **métrique de l'information de Fisher**  $G_\theta = \mathbb{E}_\theta[\partial_{\theta i} \log p(y|\theta) \partial_{\theta j} \log p(y|\theta)]$  qu'on peut calculer explicitement.
- Le paramètre  $\theta$  est fourni par le modèle  $\theta = N_\omega(x)$ . On peut "rétropropager la géométrie" en calculant la **métrique pullback**  $G_x = N_\omega^* G_\theta$ .
- Le pullback  $N_\omega^*$  ne dépend que de la dérivée de  $N_\omega$  par rapport à  $x$  et peut donc être calculé par **rétropropagation**.
- La métrique  $G_x$  permet d'étudier **comment le modèle lie l'entrée  $x$  à la sortie  $y$** .



- La métrique pullback  $G_x = N_\omega^* G_\theta$  est en général **dégénérée** i.e., son noyau est non trivial.
- L'ensemble des noyaux  $\ker G_x$  pour chaque  $x$  de l'espace d'entrée forme une distribution intégrable  $\rightarrow$  on obtient un **feuilletage**  $\ker G$  sur l'espace d'entrée.
- Ce feuilletage est intimement lié à la **robustesse** du modèle.



---

\*Source: Tron et al. Canonical foliations of neural networks: application to robustness.

- Supposons que la sortie  $y = (y_i)$  soit une **série temporelle** de distribution  $p(y|\theta)$  avec  $\theta = (\theta_1, \dots, \theta_m)$ .
  - D'après le **théorème de Takens**, tout système dynamique  $s_{t+1} = \phi(s_t) \in \mathbb{R}^n$  peut être reconstruit à partir d'une seule mesure  $\theta_t = f(s_t) \in \mathbb{R}$  suffisamment répétée  $(\theta_{t_1}, \dots, \theta_{t_{2n+1}}) = \tilde{\phi}(\theta_{t_0}, \dots, \theta_{t_{2n}})$ .
  - Le modèle a pour but de **reconstruire cette dynamique inconnue**  $N_\omega \approx \tilde{\phi}$ .
  - Ainsi, l'ensemble des trajectoires du système  $\tilde{\phi}$  forme un second **feuilletage**.
- L'étude de l'interaction des deux feuilletages permet de caractériser la **robustesse** pour les modèles travaillant sur des **séries temporelles**.

# Questions

