

# Rapport de synthèse équipe IA Voltron: AP/HP de Paris et risque infectieux dans les blocs opératoires

## Introduction

---

Les évolutions amenées par l'intelligence artificielle transforment petit à petit différents secteurs. Les métiers de la santé ont su parfaitement prendre le train en marche de cette révolution et implémenter différentes solutions de machine learning pour répondre à différents types de problématique.

Dans le but d'automatiser et faciliter la gestion des conditions sanitaire des salles d'opérations au sein de l'AP-HP, notre client

## La qualité de l'air dans un bloc opératoire et le risque infectieux

---

Les conditions d'hygiène doivent être impeccables pour toutes les interventions, quelle qu'elles soient. Le but étant que les opérations se passent dans des conditions sanitaires optimales. Pour cela, nos blocs opératoires doivent suivre les directives nous permettant d'obtenir ***une salle blanche/propres de classe ISO 5 au moins.***

### Salle propre: Définition

Salle dans laquelle la concentration des particules en suspension dans l'air est maîtrisée et qui est construite et utilisée de façon à minimiser l'introduction, la production et la rétention des particules à l'intérieur de la pièce et dans laquelle d'autres paramètres pertinents, tels que la température, l'humidité et la pression sont maîtrisés comme il convient

### Salle propre de classification ISO 5:

Pour définir si un bloc opératoire est classé comme salle blanche de classe ISO 5 il faut donc pouvoir mesurer et contrôler le nombre de particules par m<sup>3</sup>. Pour se faire il nous faut des capteurs permettant de mesurer la concentration en particules de tailles différentes. Pour connaître les concentrations maximales admissibles (en particules/m<sup>3</sup> d'air) par tailles de particules, il est nécessaire de se référer à la

[classification ISO 14644-1](#):

<b>0,1<math>\mu</math></b>	<b>0,2<math>\mu</math></b>	<b>0,3<math>\mu</math></b>	<b>0,5<math>\mu</math></b>	<b>1<math>\mu</math></b>	<b>5<math>\mu</math></b>
100 000	23 700	10 200	3 520	832	-

Par exemple pour qu'un bloc opératoire soit classer comme salle propre, il faut que lors de la mesure on retrouve moins de 100 000 particules d'une taille supérieur ou égale à 0,1  $\mu$  mètre.

## La température d'un bloc opératoire

La température du bloc opératoire doit être comprise en 19°C et 26°C (Norme hospitalière : NFS 90-351 d'Avril 2013).

## Les zones à risques (par niveau de risque)

<b>Niveau</b>	<b>Risque infectieux</b>
4	Très haut risque infectieux
3	Haut risque infectieux
2	Risque infectieux moyen
1	Risque infectieux nul

## Indicateurs qualité de l'air en fonction de la classe de risque

<b>Classe de risque</b>	<b>Classification ISO 14644-1</b>	<b>Pression différentielle</b>	<b>Plage de température</b>
4	ISO 5	15 Pa (+ 5 Pa)	19 à 26 C
3	ISO 6	15 Pa (+ 5 Pa)	19 à 26 C
2	ISO 7	15 Pa (+ 5 Pa)	19 à 26 C

## Résumé

Pour qu'un bloc opératoire soit considéré comme salle propre:

- On doit pouvoir mesurer et contrôler le nombre de micro-particules par m<sup>3</sup>
- Cette concentration doit être en dessous des concentrations maximales admissibles par taille de particules ( de 0,1 à 5  $\mu$  )
- La pression différentielle doit être de 15 Pa (+ Pa)
- La température du bloc doit être comprise en tre 19 et 26°

- Si ces conditions sont remplies, on peut dire que le niveau de risque infectieux est égal ou inférieur à 4 (Norme NFS -90-351)

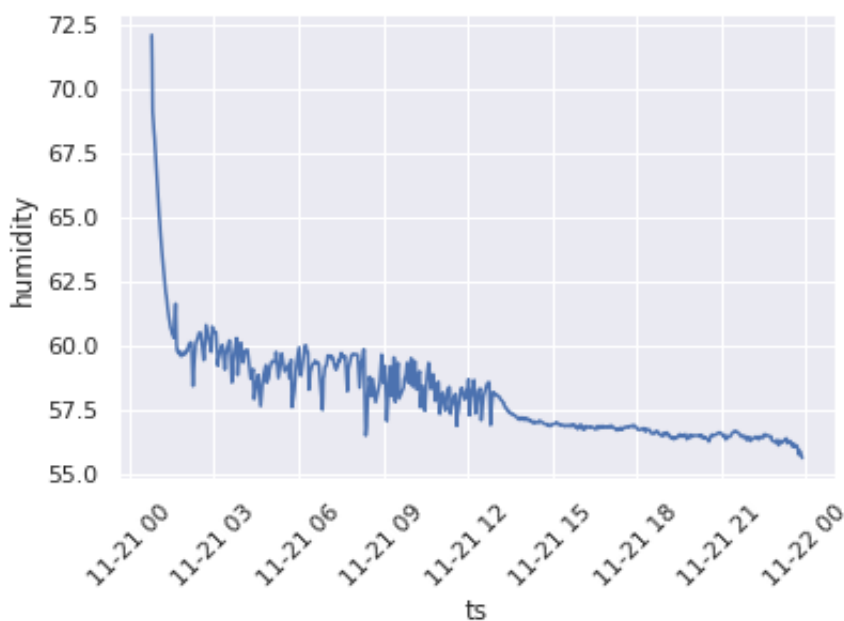
# Partie 1: Les sources de données séquentielles et prédiction temporelle

Dans notre cas nos données représentent les mesures de différents paramètres relatifs à la qualité de l'air dans un bloc opératoire en fonction d'un timestamp (une date). C'est donc ce que l'on nomme des données séquentielles.

## Les series temporelles ou Time Series

---

Comme nous allons pouvoir le voir sur les différentes visualisations, nos données représentent l'évolution d'une mesure (caractéristiques de la qualité de l'air en fonction du temps).



Ce que l'on voudrait faire avec notre modèle de machine learning c'est de la prédiction de serie temporelle, c'est à dire apprendre à partir d'information historique pour pouvoir prévoir ou prédire des informations futures inconnues.

Dans notre cas nous voulons prévoir qu'elles vont être pour une date future les conditions de qualité de l'air pour un bloc opératoire donné. Nous avons donc besoin de choisir un modèle pour ces capacités de forecasting (prévision) sur des séries temporelles.

## Les réseaux de neurones récurrents

---

Notre objectifs va être donc de concevoir un réseau de neurones qui va être capable d'apprendre d'un historique de mesure pour pouvoir prédire les prochaines mesures probables dans le futur.

Il faut que l'on assure que notre réseau soit alimenté avec suffisamment avec assez de séquence d'information pour pouvoir saisir toute tendance dans nos données (ex: Existe il une tendance dans les mesures de l'humidité qui vont nous permettre de pouvoir déterminer la prochaine mesure à  $t+1$ ).

## La problématique des RNNs et des prédictions à long terme

---

Le problème que l'on peut rencontrer avec les réseaux de neurones récurrents sur des prédictions visant une période de temps à plus long terme, est que le modèle va devoir se baser sur les prédictions à court terme pour pouvoir faire des prédictions sur les dates plus lointaines. C'est donc des prédictions basées sur des prédictions et la moindre erreur mineure peut être amplifiée sur les prédictions à long terme.

Nous avons donc décidé que les prédictions à court terme seraient suffisantes pour notre cas car il n'est pas nécessaire de devoir anticiper les conditions de qualité de l'air du bloc opératoire sur le long terme. Cependant une prédiction à court terme peut permettre de prendre des décisions efficaces pour améliorer la gestion des conditions sanitaires des salles d'opération et permettre d'avoir le plus de salles opérationnelles à un instant  $T$ .

L'objectif va être de concevoir un réseau de neurone récurrent qui va nous permettre de prédire quelles seront les conditions de qualité de l'air pour l'heure à venir.

## Traitement des données et mise en place du RNN

---

### Traitement des données

En amont il est nécessaire de traiter les données afin de répondre à la problématique de prédire les conditions de qualité de l'air pour l'heure à venir.

Tout d'abord nous avons limité notre dataset à une journée pour que notre modèle ne soit pas trop gourmand en ressource.

Nous avons déterminé que pour délimiter une heure en terme de point de donnée il est nécessaire d'avoir 60 points de données. Pour une journée il nous faut 700 points de données.

Ces données sont nécessaires pour pouvoir déterminer la taille de notre dataset d'entraînement et de notre dataset de test. Notre dataset d'entraînement devra se limiter aux données d'une heure pour pouvoir prédire les conditions à l'heure suivante

### Datasets

Le dataset d'entrainement est composé des points de données auquel on va soustraire les points de données du dataset de test. Et le dataset de test va donc être composé des points de données restant (1 heure de mesure).

## Le RNN

```
[266] from tensorflow.keras.models import Sequential
      from tensorflow.keras.layers import Dense, LSTM
```

```
[267] # We use only one feature (humidity)
      n_features = 1
```

```
[268] # We define the model
      model = Sequential()
      model.add(LSTM(100, activation='relu', input_shape=(length, n_features)))
      model.add(Dense(1))
      model.compile(optimizer='adam', loss='mse')
```

```
[272] model.summary()
```

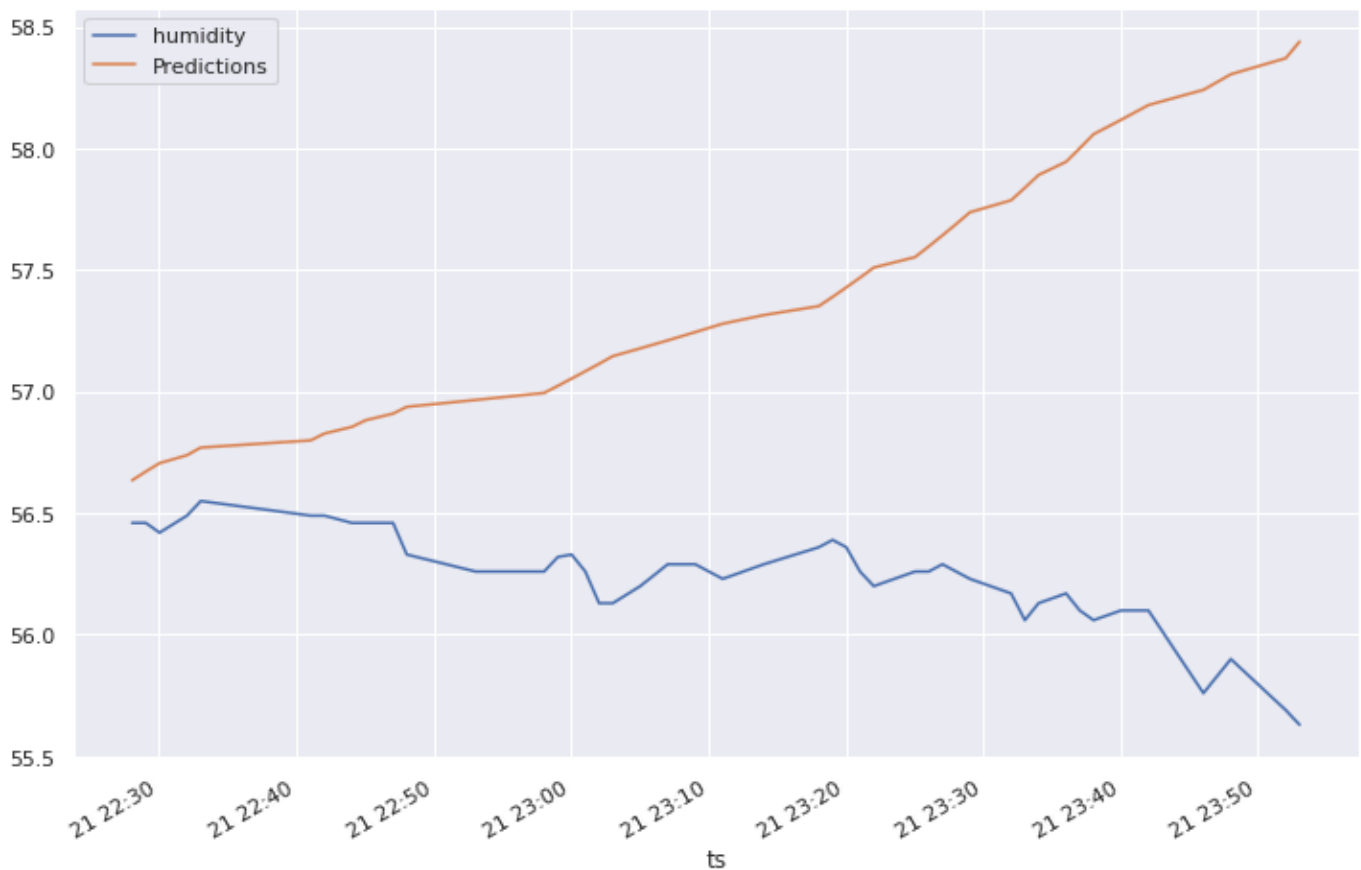
Model: "sequential\_2"

Layer (type)	Output Shape	Param #
=====		
lstm_2 (LSTM)	(None, 100)	40800
=====		
dense_2 (Dense)	(None, 1)	101
=====		

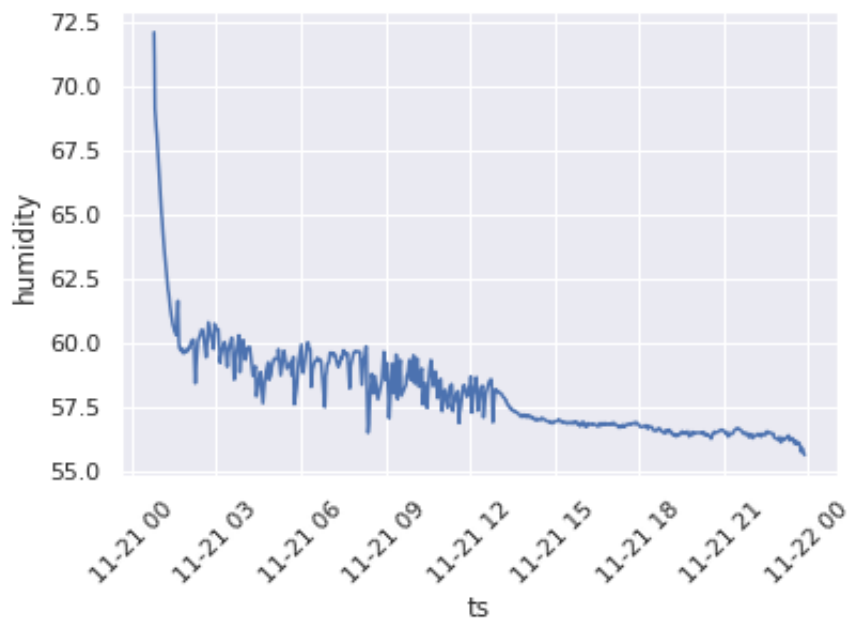
Total params: 40,901  
Trainable params: 40,901  
Non-trainable params: 0

## Résultats et interprétation

---



Comme on peut le voir sur ce graphique, les prédictions ne se rapproche pas de celle de l'heure précédente. En effet comme notre modèle se base sur les heures précédentes et comme nous pouvons le voir sur les graphique d'une journée il n'existe pas de saisonnalité sur une journée qui nous permettent de dégager un pattern de récurrence sur lequel se baser pour tirer une conclusion sur la précision du modèle.



## Partie 2: Classification de données labélisées

L'objectif de ce modèle va être de pouvoir catégoriser une salle d'opération selon si elle est au norme recommandée pour un risque infectieux toléré. Pour ce faire il est nécessaire dans une première phase de pouvoir labéliser nos données en fonction des seuils définies par la norme ISO 5 relative aux "salles blanches"/"salles propres".

## Preprocessing et Labellisation

---

### Contexte

Nous avons à attendre que l'équipe IOT mette en place ses capteurs, récupère les différentes métriques, les transmette à l'équipe Big Data qui nous aurait nettoyé le jeu de données avant de nous le transmettre.

Afin de gagner du temps et dans le contexte d'un POC, nous avons cherché à mocker les données qui arriveraient dans notre système. Nous avons trouvé un jeu de données comprenant toutes les métriques que nous souhaitions exploitées sur [Kaggle](#).

Cependant, les données récupérées ne provenaient pas de d'un bloc opératoire. Il a donc été question de les ajuster afin qu'elles respectent plus ou moins les spécifications que nous cherchons à suivre.

- ppm25: < 0.56
- ppm10: < 1.76
- temperature: 19 - 26 °C (ISO5)
- humidité: 45 - 65 % (ISO5)
- co2: 300 – 380 ppm

Notre objectif à terme est labelliser notre jeu de données de manière binaire en fonction des différents paramètres. Soit toutes les conditions sont respectées et on obtient *True*, sinon on obtient *False*.

Nous avons utilisé pandas et numpy pour effectuer différentes opérations mathématiques sur chaque élément des colonnes désirées afin de réduire l'écart-type, ajuster la moyenne et faire en sorte d'avoir une suffisamment de labels *True* pour que ce soit exploitable.

### Résultats

	co2	humidity	pm10	pm25	temperature	
mean	354	50%	1.6	0.63	23°C	
25%	311	46%	1.42	0.25	21°C	
50%	320	50%	1.65	0.55	23°C	
75%	382	55%	1.87	0.94	25°C	
std	64.3	6.8%	0.31	0.45	2.1°C	
True	44%	52%	63%	64%	58%	Global : 26%

## Détermination du label

Dans cette partie, il a été question de déterminer le label de la salle étudiée en fonction des paramètres. Dans notre cas, l'étape de labellisation suffit, mais il est possible qu'une norme produite par ce type de modèle permette plus de finesse.

Une fois notre jeu de données labellisé, nous l'avons normalisé, afin que toutes les valeurs des paramètres observés soient comprises entre 0 et 1. Ensuite, nous avons séparé le jeu de données en deux sous-ensembles, pour l'entraînement et le test. Les différents datasets ont permis d'entraîner notre modèle. Nous avons utilisé la librairie Keras pour le construire, en créant un modèle séquentiel dans lequel nous avons fait se suivre des couches LSTM ainsi qu'une couche dense afin de restructurer. En observant la perte de notre modèle, on remarque qu'il est très précis ce qui est logique, du fait que l'on labellise suivant des règles très précises.

