**Cadrage**

**(deadline 29/03)**

[**DEFINITION du projet**](#_22aq1ouwzbxs) **3**

[La problématique](#_nxx8zk3h6hy0) 3

[Le cadre / hors cadre du projet](#_98cc78vprzj3) 3

[Logigramme:](#_5bxpyyxoaqs2) 4

[Les données à disposition](#_nlfociwnudq) 4

[Benchmark de l’existant, La stratégie de mesure et de contrôle des objectifs](#_xhwd9bt4g9t5) 5

[**MESURE**](#_ad7o8hxfgjoy) **6**

[Jeux de donnée à disposition](#_32dso6s2552m) 6

[Premières écoutes](#_1gkjhom7e9ue) 6

[Caractérisation fréquentielle uni temporelle des données](#_2if6x0r65vz1) 7

[Caractérisation fréquentielle temporelle des données](#_uo21gr8bnq11) 8

[Utilisation du LogMel](#_x3xkoexzas1e) 8

[**ANALYSE**](#_k89skwespx9v) **9**

[Etat de l’art et recherches dans le domaine](#_otedehwzu9w4) 9

[Premiers essais avec un modèle de classification](#_keckiv8sd6zj) 9

[Premiers essais avec un Auto Encoder](#_kozvk7ehlb1v) 10

[**INNOVATION**](#_x8xis0wi6mlb) **11**

[Définition de la stratégie](#_bdr9u3naui3d) 11

[Pre-processing](#_c8id02uy8i1l) 11

[Générateur pour augmenter les données :](#_ujejjvkv4vd4) 11

[**CONTROLE**](#_f6yg5s81zlq) **12**

[Essais sur Auto-Encoder](#_n383wsf00tz2) 12

[Essais Classificateur d’IDs](#_xmr49ycezs0n) 28

## **DEFINITION du projet**

### **La problématique**

La robotisation et l’automatisation des machines et des biens est un enjeux du 21 ème siècle. A cette fin, de plus en plus d’industries investissent pour rendre totalement autonome des parties de leur process ou des tâches qui nécessitent habituellement la présence constante d’un opérateur. Les exemples les plus représentatifs dans l’industrie sont les robots de manutention et les chaînes d’assemblage automatiques. Concernant le quotidien, la course à la voiture autonome et aux drones de surveillance / livraison en sont des exemples les plus visibles.

La problématique associée à ces enjeux est l’automaintenance des machines; tâche de surveillance réalisée dans le passé par l’opérateur, son absence induit un contrôle moins fréquent des paramètres et donc un risque de panne plus élevé. Dans cet objectif, l’enjeu du projet est de pouvoir détecter une anomalie dans une machine grâce à sa signature audio.

Le livrable principal du projet sera un ou plusieurs algorithmes de contrôle et d’alerte implémentables.

Une piste d’ouverture pourrait être un dispositif (type Raspberry Pi équipé d’un micro) configuré pour réaliser cette tâche.

### Le cadre / hors cadre du projet

Plusieurs types d'anomalies / problèmes peuvent survenir sur une machine, aussi dans le cadre de ce projet nous adopterons le découpage suivant :

Type d’anomalie à détecter :

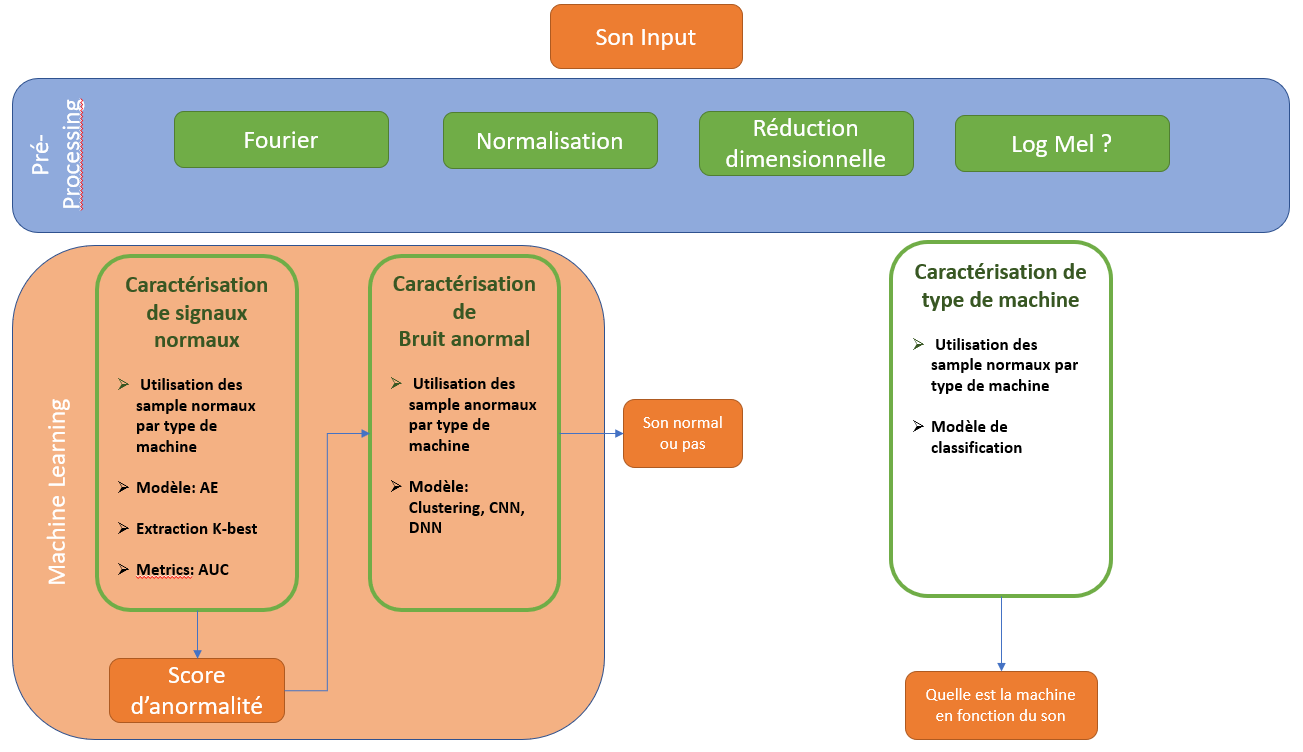
* Les bruits redondants : défauts liés à une usure ou casse qui entraîne un son particulier à une fréquence f et une périodicité multiple de la fréquence k de travail de la machine. Exemple : Un ventilateur tournant à 100 tours par minutes (fréquence k) pourra émettre un bruit de frottement à [100/k,100\*k] période, avec une signature audio de fréquence f.
* Les bruits singuliers identifiés par le modèle: une anomalie pour laquelle un enregistrement audio a déjà été analysé et implémenté dans le modèle.

Type d’anomalie exclu :

* Les anomalies n'émettant pas de son: Problème électrique / électronique
* Les anomalies singulières passagères : La chute d’un objet sur la machine, une rupture nette amenant l’arrêt complet de la machine
* Les anomalies singulières inconnues (non référencées dans les bruits anomalie)

### 

### **Logigramme:**



### **Les données à dispositio**n

Nombre de fichiers à disposition

|  |  |  |  |
| --- | --- | --- | --- |
| **Machine** | **Fichiers audio Total** | **Fichiers Anomalie** | **Fichiers Normaux** |
| ToyConveyor | 6509 | 1059 | 5400 |
| ToyCar | 6459 | 1110 | 5399 |
| fan | 5550 | 1475 | 4075 |
| pump | 4205 | 456 | 3749 |
| valve | 4170 | 890 | 3204 |
| slider | 4094 | 479 | 3691 |
| **Total** | **30987** | **5469** | **25518** |

### **Benchmark de l’existant, La stratégie de mesure et de contrôle des objectifs**

**1/ Modèles utilisés en ASD (état de l’art)**

La revue [arXiv:2102.07820](https://arxiv.org/abs/2102.07820) qui recense une trentaine de publications supposées être les plus pertinentes dans le domaine du Machine Learning appliqué à l’ASD (Anomaly Sound Detection) a permis de mettre en avant les modèles les plus fréquemment utilisés pour répondre à ce type de problématique: il s’agit principalement de réseaux de neurones de type CNN (réseaux de neurones convolutionnels) et AE (AutoEncoder). Les jeux de données utilisés dans ces études sont en majorité les datasets ToyADMOS et MIMII qui sont inclus dans notre dataset DCASE2020-Task2. Les études sur ToyADMOS et MIMII ont toutes utilisé la courbe ROC et la métrique associée AUC pour évaluer leur modèle.

**2/ Modèles utilisés spécifiquement dans le cadre du challenge DCASE2020-Task2**

D’après la revue consacrée spécifiquement aux études ayant développé les modèles les plus performants sur notre jeu de données DCASE2020-Task2 (*Koyzumi & al, 2020*) certaines se sont démarquées pour avoir utilisé une approche de classification tandis que les autres se sont focalisées sur la détection d'outliers par rapport à une “baseline”.

Le système de baseline repose sur un AutoEncoder qui permet de calculer un score d’anomalie. L’AE est un réseau de neurones de type FCNN ( Fully Connected Neural Network) qui prend en input un spectrogramme log-mel. Il est entraîné sur des données “normales” et son optimisation consiste à minimiser au maximum l’erreur associée à la reconstruction du signal normal. Finalement le score d’anomalie est calculé à partir de cette erreur. Autrement dit l’AE est optimisé pour minimiser le score d’anomalie associé aux fichiers de label “normal” sur lesquels il est entraîné, dans le but de pouvoir détecter l’anormalité sur des données test de label “anormal”.

Dans le cas où les participants du Challenge ont adopté une méthode de classification via un DNN (Deep neural network), ils ont utilisé, pour une machine donnée, les fichiers de label “normal” d’autres machines comme des labels anormaux.

Enfin une dernière approche innovante, basée sur un AutoEncoder, a consisté à prévenir un potentiel effet indésirable de l’AE: à savoir associer des scores d’anormalité élevés à un ID de machine différent de celui avec lequel il a été entraîné et qui est pourtant de label “normal”. Pour pallier ce problème deux équipes ont eu recours à deux stratégies différentes: la première a entraîné son AE avec un label “anormal” pour entraîner l’AE à ne pas reconstruire de signal “normal” dans ce cas (*Daniluk & al., 2020*). L’autre stratégie a consisté à inclure l’ID de la machine aux features à extraire ainsi le modèle ne reconstruit pas seulement les features du spectre audio mais aussi l’ID machine (*Hayashi & al., 2020)*.

## **MESURE**

### **Jeux de donnée à disposition**

La taille du jeu de données est de 9 Go répartis en 30987 fichiers soit une taille moyenne de 290 Ko par fichier.

Notre dataset comprend 5469 fichiers d’anomalie soit environ 17% du jeu de données, ce qui est proportion non négligeable mais faible, nous privilégierons donc dans un premier temps des approches non supervisées ou semi supervisées.

### **Premières écoutes**

Les fichiers sont classés par type de machine (pompe, ventilateur, convoyeur) et par ID de machine, nous pouvons donc supposer que chaque ID correspond à une machine unique. Le jeu de données par machine est scindé en trois : une partie normale, une partie normale pour réaliser un test et une partie contenant des anomalies. Toutefois, il n’est pas indiqué dans le jeu de données le nombre d'anomalies différentes enregistrées par machine et leur répartition dans les enregistrements. Il faudra donc générer une étape de classification non supervisée sur ces données pour essayer de les caractériser.

**Pump**

Les premières écoutes sont assez complexes car si certains problèmes ressortent assez franchement, comme des frottements ou des cliquetis, certains enregistrements ne semblent présenter aucune différence entre le normal et l’anomalie notamment dans les enregistrements de pump ID06 et 04. De même le bruit de l’écoulement de l’eau est très présent dans certains enregistrements, et il faudra donc décider si ce son est un indicateur de bon/mauvais fonctionnement ou un parasite.

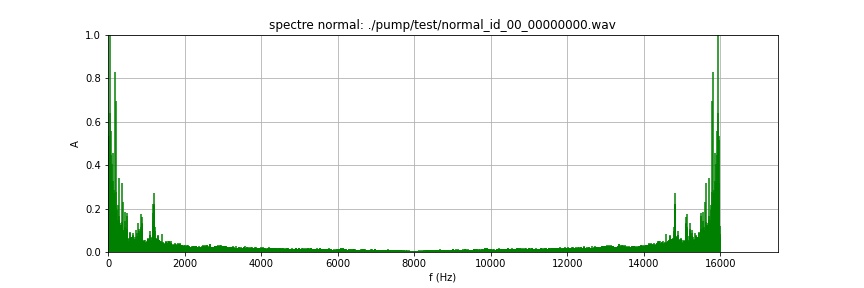
**Slider**

De manière générale, les enregistrements de slider sont peu bruyants. De même certains enregistrements présentent des bruits d’anomalie très présents, et qui semblent revenir sur plusieurs enregistrements différents. Ce jeu de données peut être un bon début pour tester des algorithmes.

### **Caractérisation fréquentielle uni temporelle des données**

La première piste suivie est d’essayer d’extraire une signature fréquentielle d’une machine et de la comparer à la même machine avec anomalie.

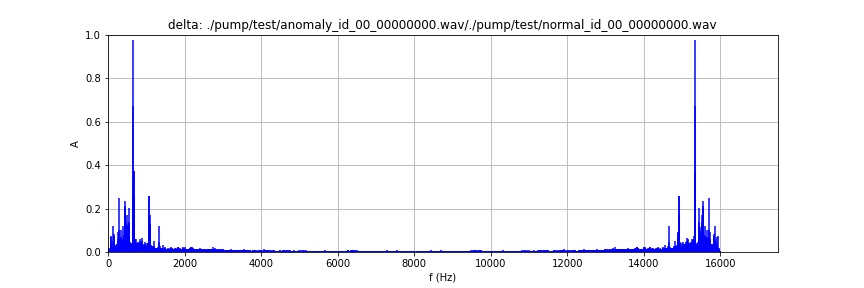
La première chose que nous pouvons voir sur les graphiques sont la fréquence d’échantillonnage à 16Khz, au-delà de laquelle il n’y a plus de fréquence présente dans l’analyse.

Pump :

Nous pouvons voir que la machine “pump” a une signature très marquée qui se caractérise par une amplitude élevée pour des fréquences respectivement inférieure à 2 kHz et comprise entre 14 et 16 kHz. A l’inverse on observe une amplitude quasiment nulle entre 2Khz et 14khz. De plus le spectre est tronqué à 16 kHz, la fréquence d'échantillonnage. Cela pourrait engendrer des problématiques de modélisation par la suite.



Le spectre de l’anomalie est semblable à celui de la machine saine en termes de répartition des fréquences: il n’y a pas de signal marqué dans l’intervalle 2-14khz. En revanche, à une fréquence légèrement inférieure à 1 kHz et à 15.5 kHz respectivement, on observe un pic unique d’amplitude maximale. Tandis que sur le spectre “normal” on observe plusieurs pics d'amplitude élevée et décroissante entre zéro kHz et un peu moins de 1 kHz: la fréquence du pic unique sur le spectre anormal. De même que l’on dénote plusieurs pics d'amplitude élevée et croissante entre 15 (la fréquence du pic unique du spectre “anormal”) et 16 kHz.



Le pattern obtenu suite à la soustraction des fréquences du spectre “normal” à celles du spectre “anormal” est très similaire au pattern “anormal”: on retrouve 2 pics uniques un peu en dessous de 1 kHz et à 15 kHz.

Nous pouvons poser une hypothèse : ces deux pics sont peut-être la signature sonore de l’anomalie constatée.

### **Caractérisation fréquentielle temporelle des données**

### **Utilisation du LogMel**

### 

## ANALYSE

### **Etat de l’art et recherches dans le domaine**

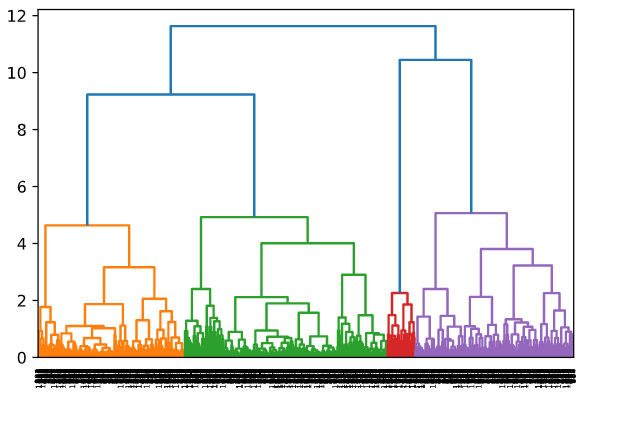
### **Premiers essais avec un modèle de classification**

Dendrogramme sur la machine Fan avec que des données normales.

Nous pouvons voir 4 groupes se constituer assez nettement, nous avions 4 machines ID nous pouvons donc supposer que ces groupes correspondent aux ID machines.

### 

En ajoutant les anomalies :



### **Premiers essais avec un Auto Encoder**

## INNOVATION

### Définition de la stratégie

### Pre-processing

### Générateur pour augmenter les données :

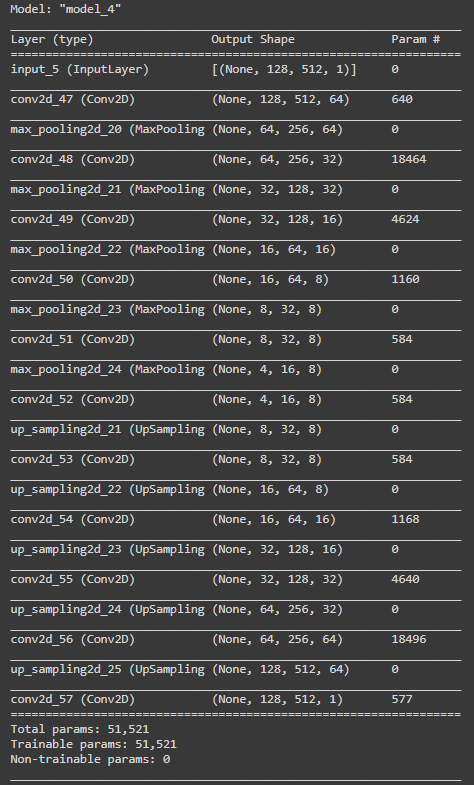
L’objectif est de générer de nouvelles données par augmentation légère des données de base. Sur les spectres ci-dessous, nous avons généré une homothétie de l’amplitude des logmel de 10% aléatoirement. Vous pouvez-constater que sur le premier et le dernier spectre les valeurs s’étalent de 0.2 à 0.8 au lieu de 0.2 à 1.

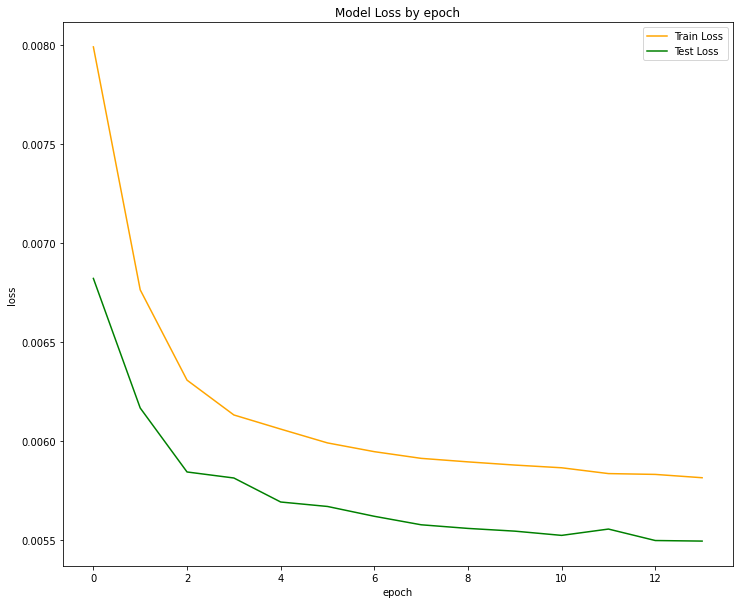
### 

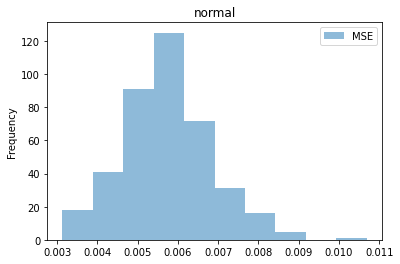
## CONTROLE

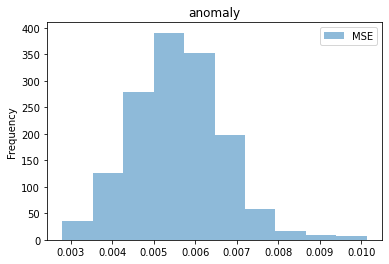
### Essais sur Auto-Encoder

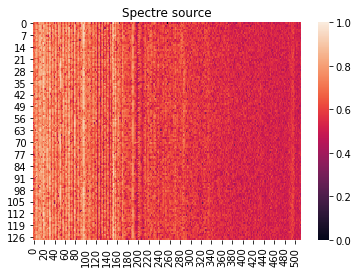
Sorties du premier modèle dt=128 freq=512

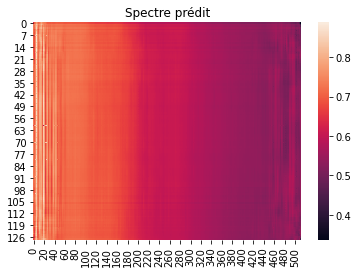




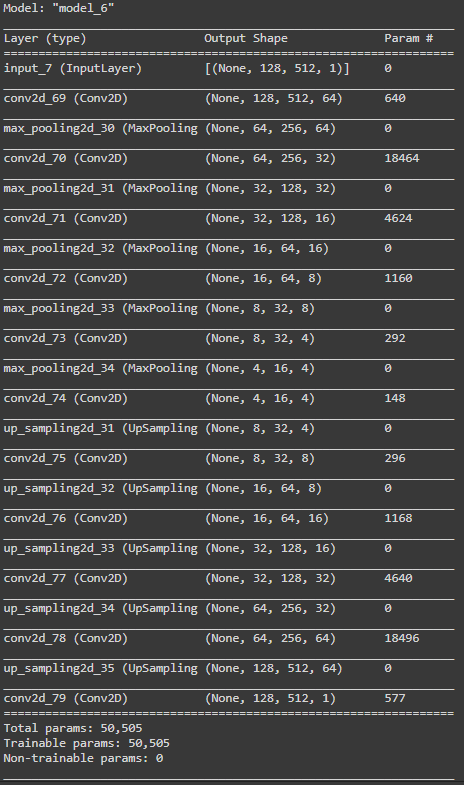




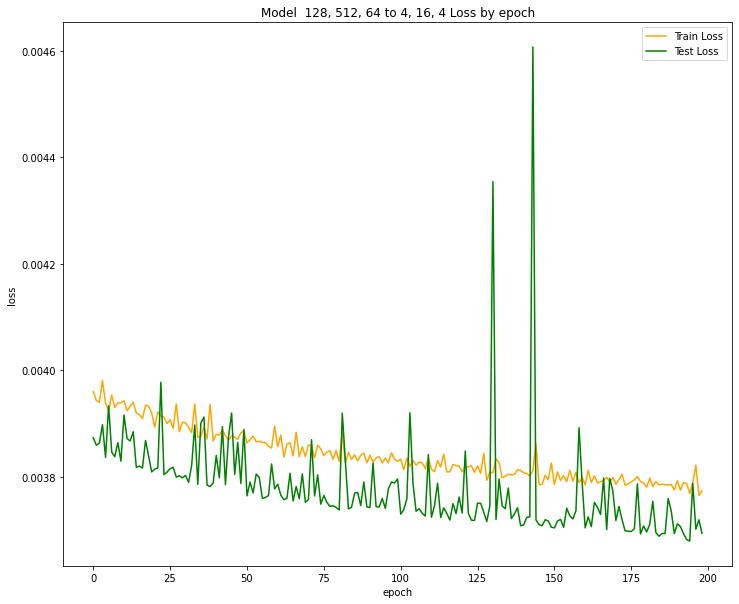
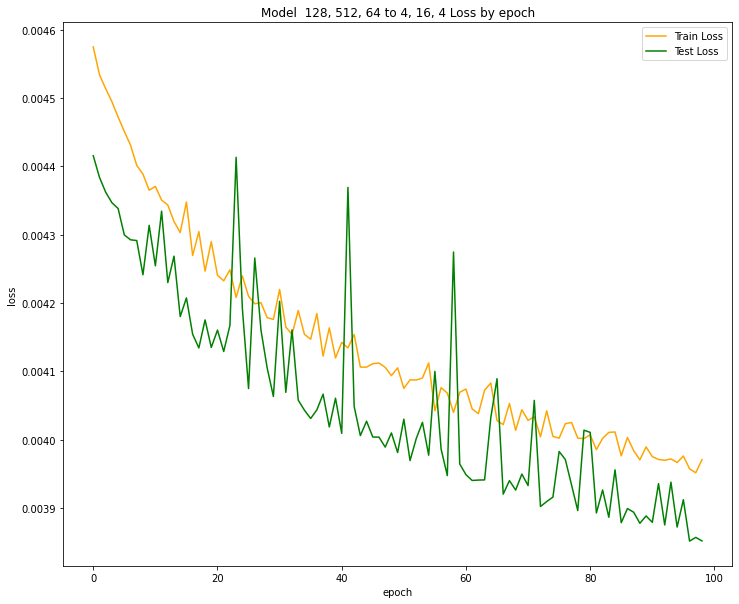


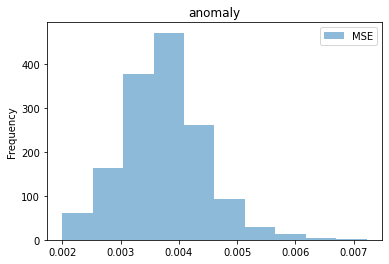
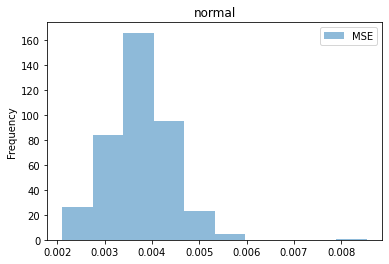
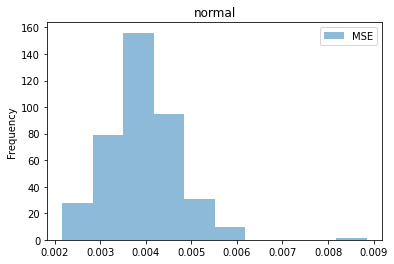
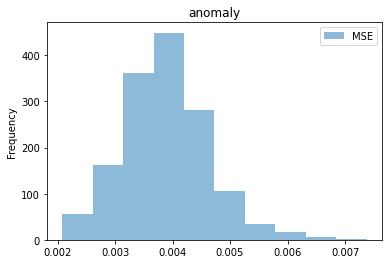


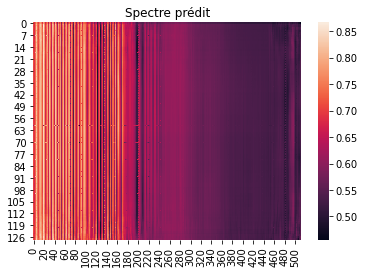
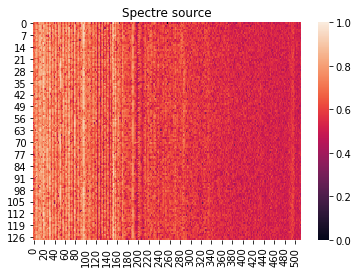
Sorties du second modèle dt=128 freq=512



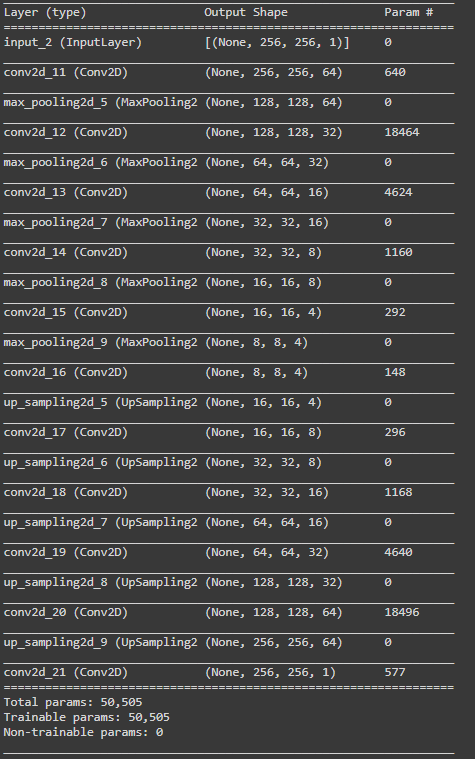
Après 100 EPOCHS puis après 200 EPOCHS supplémentaires



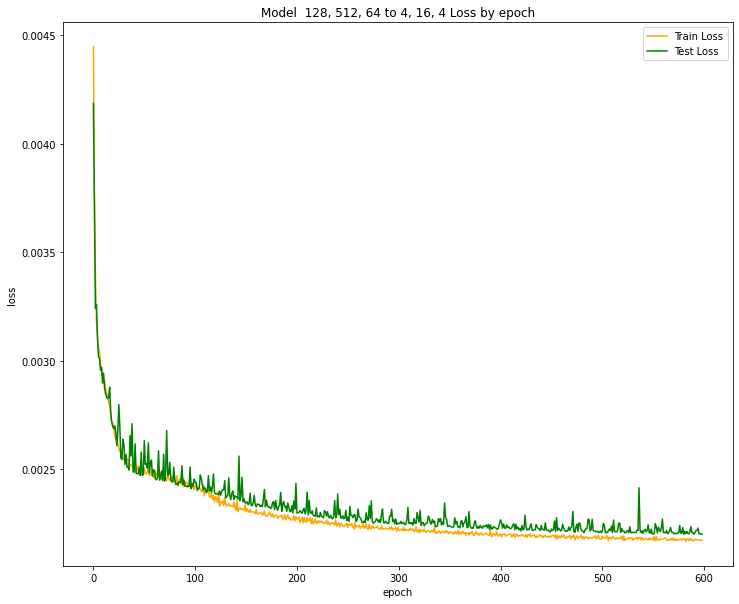




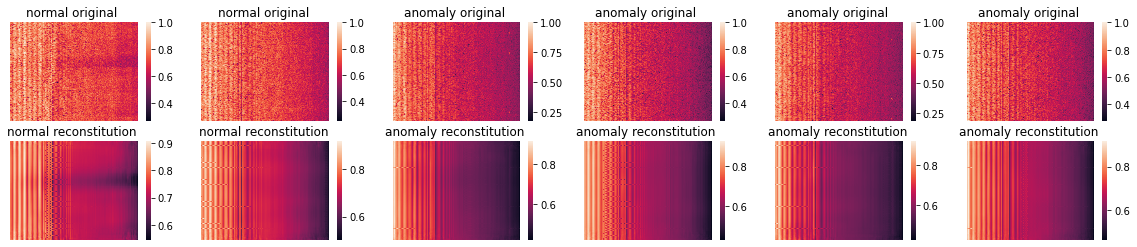
Troisième modèle dt=256 freq=256 goulot à 8x8x4



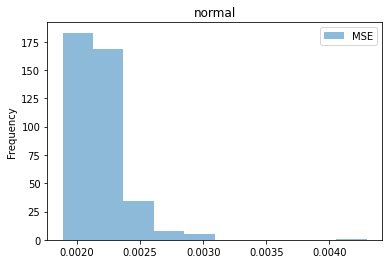
Entraînement et prédiction sur Fan sur 600EPOCHS

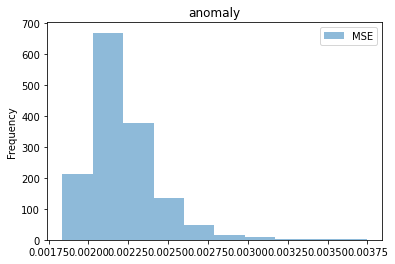


Nous pouvons voir que les reconstitutions sont plutôt fidèles et peu bruitées.

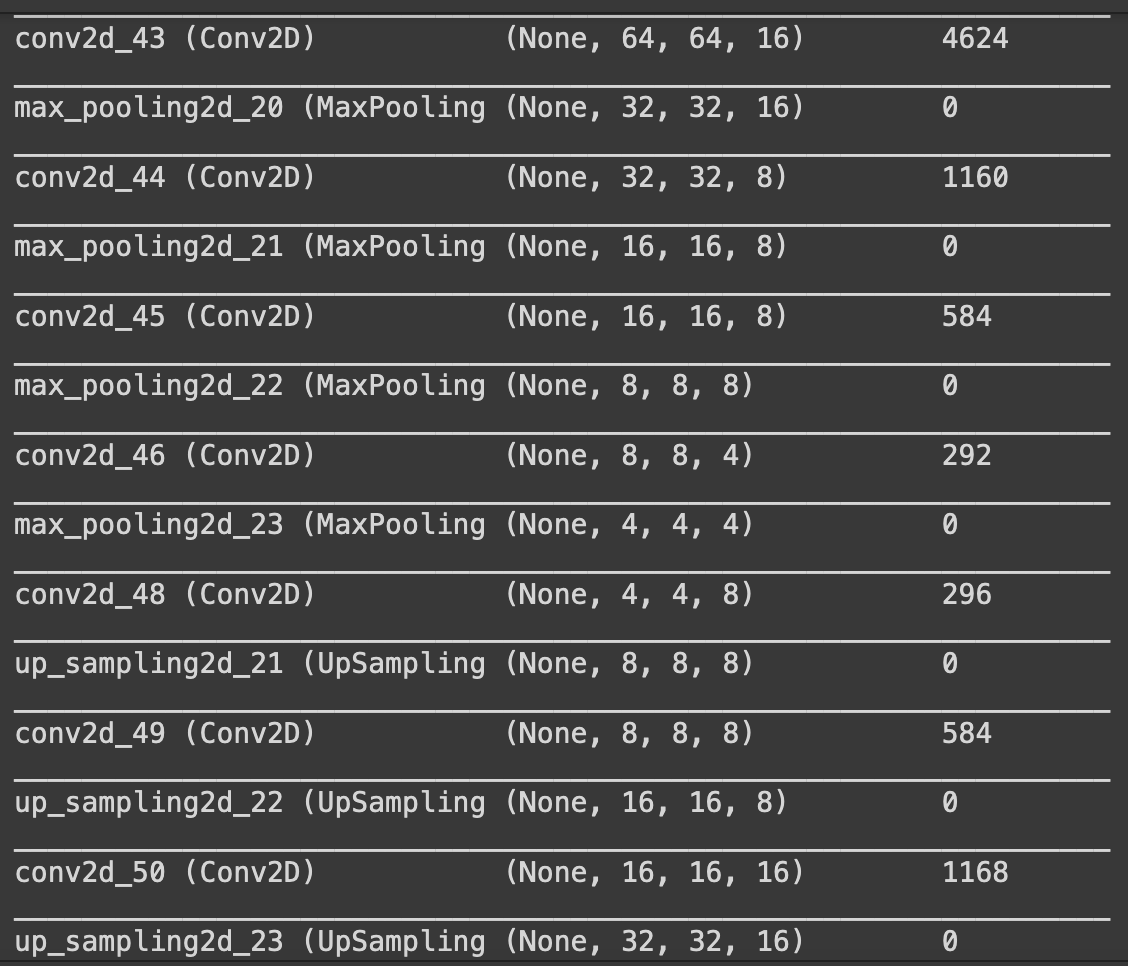


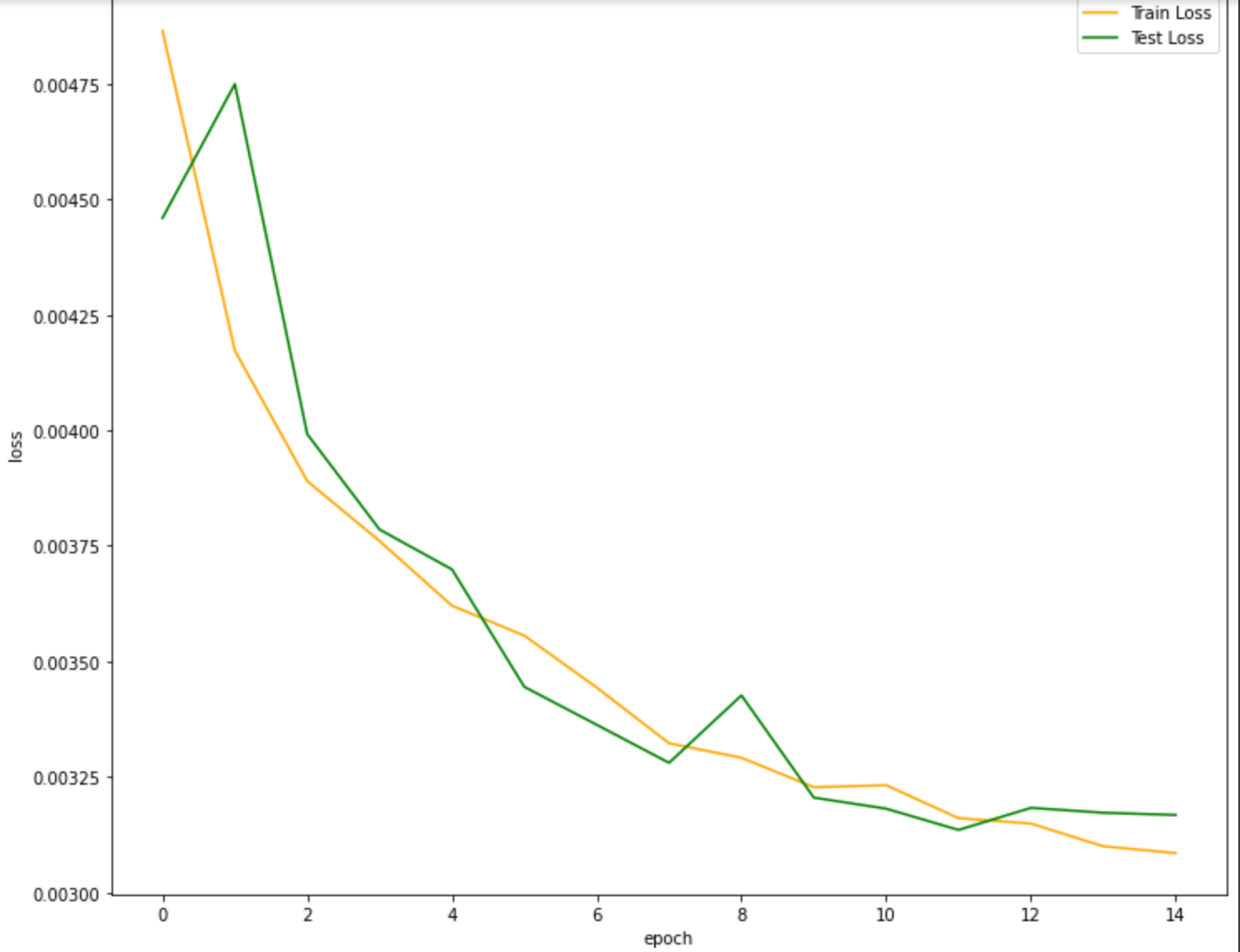
Toutefois les écarts de MSE ne permettent en aucun cas une identification normal/ anormal sur ce seul critère.



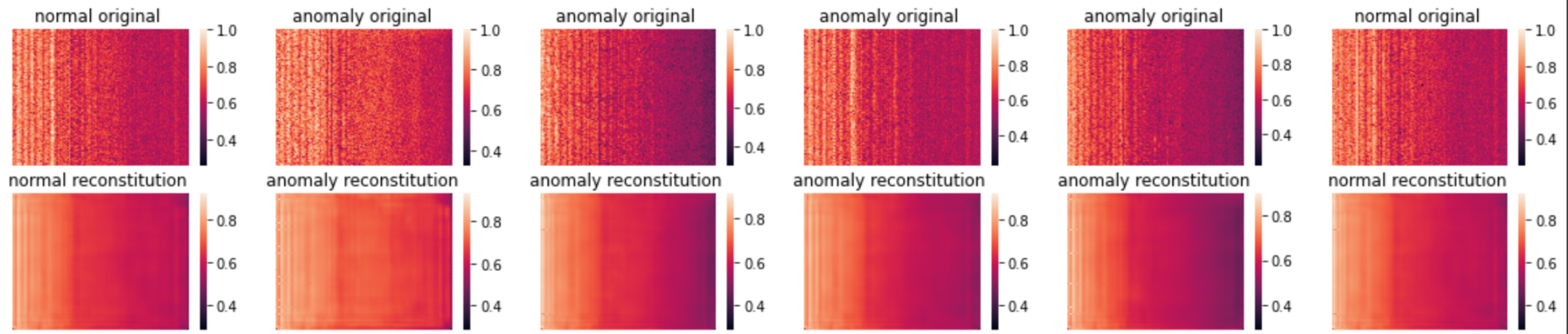


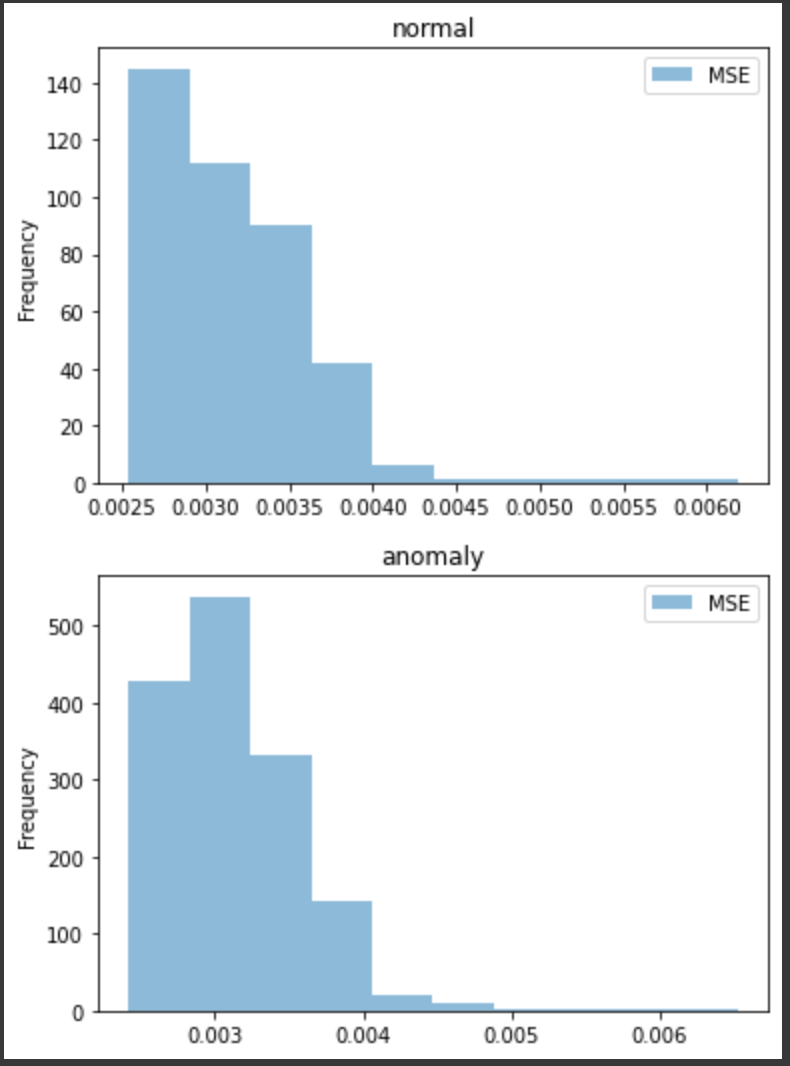
**Test d’un goulot d’étranglement plus resserré: compression [4, 4]**





Le modèle a convergé au bout de 16 epochs :





Ajout de Batch normalization

Layer (type) Output Shape Param #

=================================================================

conv2d\_37 (Conv2D) (None, 256, 256, 32) 320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_17 (Batc (None, 256, 256, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_12 (MaxPooling (None, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_38 (Conv2D) (None, 128, 128, 16) 4624

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_18 (Batc (None, 128, 128, 16) 64

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_13 (MaxPooling (None, 64, 64, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_39 (Conv2D) (None, 64, 64, 8) 1160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_19 (Batc (None, 64, 64, 8) 32

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_14 (MaxPooling (None, 32, 32, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_40 (Conv2D) (None, 32, 32, 8) 584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_20 (Batc (None, 32, 32, 8) 32

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_15 (MaxPooling (None, 16, 16, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_41 (Conv2D) (None, 16, 16, 8) 584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_7 (UpSampling2 (None, 32, 32, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_42 (Conv2D) (None, 32, 32, 8) 584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_8 (UpSampling2 (None, 64, 64, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_43 (Conv2D) (None, 64, 64, 16) 1168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_9 (UpSampling2 (None, 128, 128, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_44 (Conv2D) (None, 128, 128, 16) 2320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_10 (UpSampling (None, 256, 256, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_45 (Conv2D) (None, 256, 256, 1) 145

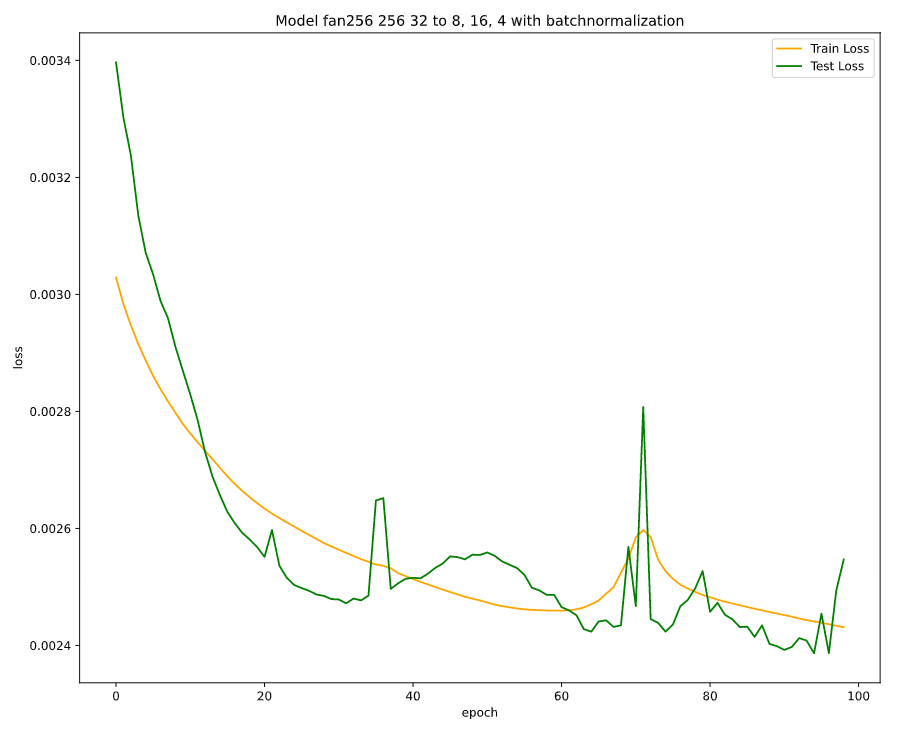
=================================================================

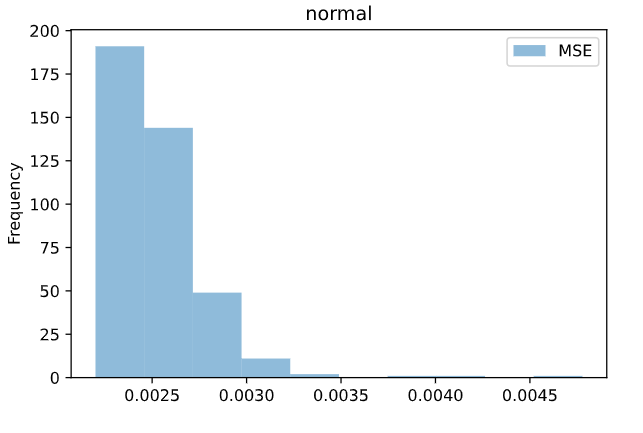
Total params: 11,745

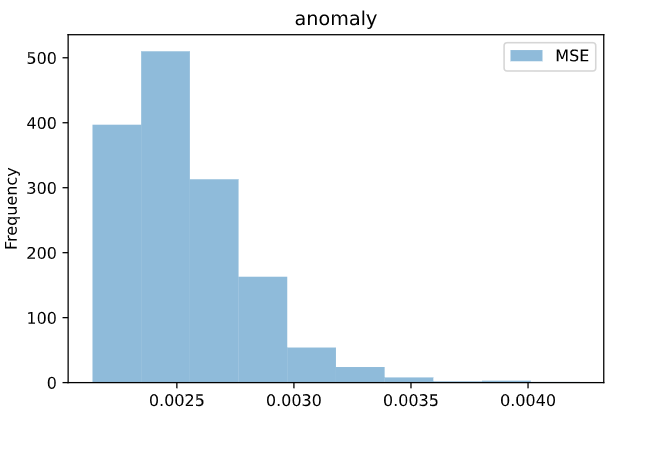
Trainable params: 11,617

Non-trainable params: 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_







Layer (type) Output Shape Param #

=================================================================

conv2d\_78 (Conv2D) (None, 256, 256, 64) 1664

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_79 (Conv2D) (None, 256, 256, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_36 (Batc (None, 256, 256, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_36 (MaxPooling (None, 128, 128, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_80 (Conv2D) (None, 128, 128, 32) 18464

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_37 (Batc (None, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_37 (MaxPooling (None, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_81 (Conv2D) (None, 64, 64, 16) 4624

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_38 (Batc (None, 64, 64, 16) 64

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_38 (MaxPooling (None, 32, 32, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_82 (Conv2D) (None, 32, 32, 8) 1160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_39 (Batc (None, 32, 32, 8) 32

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_39 (MaxPooling (None, 16, 16, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_83 (Conv2D) (None, 16, 16, 4) 292

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_40 (Batc (None, 16, 16, 4) 16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_40 (MaxPooling (None, 8, 8, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_84 (Conv2D) (None, 8, 8, 4) 148

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_41 (Batc (None, 8, 8, 4) 16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_41 (MaxPooling (None, 4, 4, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_85 (Conv2D) (None, 4, 4, 4) 148

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_36 (UpSampling (None, 8, 8, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_86 (Conv2D) (None, 8, 8, 4) 148

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_37 (UpSampling (None, 16, 16, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_87 (Conv2D) (None, 16, 16, 8) 296

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_38 (UpSampling (None, 32, 32, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_88 (Conv2D) (None, 32, 32, 16) 1168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_39 (UpSampling (None, 64, 64, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_89 (Conv2D) (None, 64, 64, 16) 2320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_40 (UpSampling (None, 128, 128, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_90 (Conv2D) (None, 128, 128, 32) 4640

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_41 (UpSampling (None, 256, 256, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_91 (Conv2D) (None, 256, 256, 1) 289

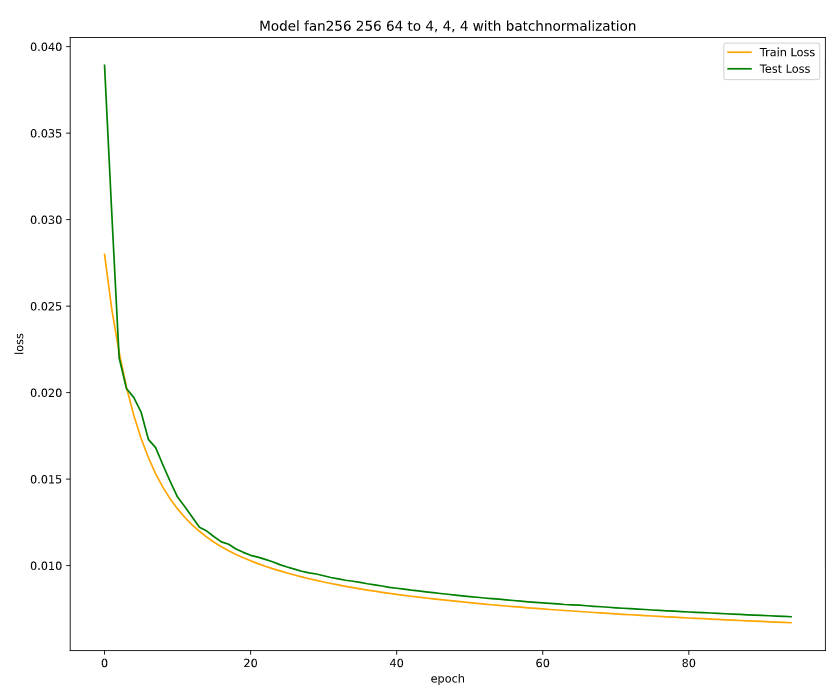
=================================================================

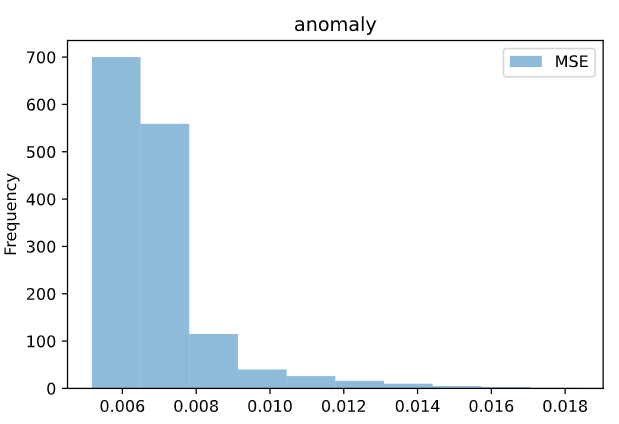
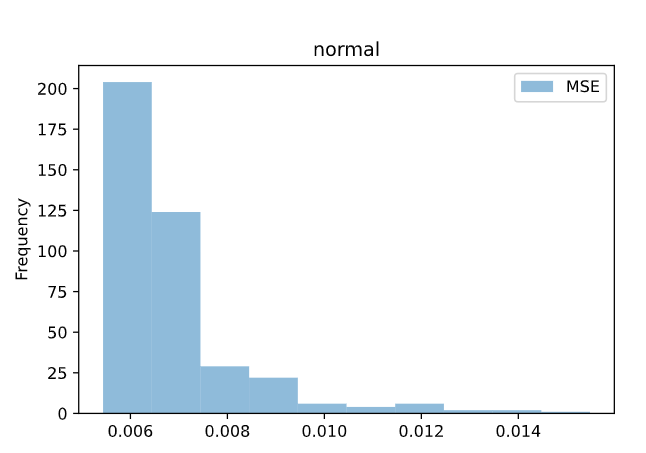
Total params: 72,801

Trainable params: 72,545

Non-trainable params: 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_





Nouveau test en forçant le modèle à avoir des poids positifs afin de ne pas occulter certaines données

Layer (type) Output Shape Param #

=================================================================

conv2d\_106 (Conv2D) (None, 256, 256, 64) 1664

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_107 (Conv2D) (None, 256, 256, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_48 (Batc (None, 256, 256, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_48 (MaxPooling (None, 128, 128, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_108 (Conv2D) (None, 128, 128, 32) 18464

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_49 (Batc (None, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_49 (MaxPooling (None, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_109 (Conv2D) (None, 64, 64, 16) 4624

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_50 (Batc (None, 64, 64, 16) 64

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_50 (MaxPooling (None, 32, 32, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_110 (Conv2D) (None, 32, 32, 8) 1160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_51 (Batc (None, 32, 32, 8) 32

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_51 (MaxPooling (None, 16, 16, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_111 (Conv2D) (None, 16, 16, 4) 292

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_52 (Batc (None, 16, 16, 4) 16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_52 (MaxPooling (None, 8, 8, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_112 (Conv2D) (None, 8, 8, 4) 148

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_53 (Batc (None, 8, 8, 4) 16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_53 (MaxPooling (None, 4, 4, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_113 (Conv2D) (None, 4, 4, 4) 148

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_48 (UpSampling (None, 8, 8, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_114 (Conv2D) (None, 8, 8, 4) 148

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_49 (UpSampling (None, 16, 16, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_115 (Conv2D) (None, 16, 16, 8) 296

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_50 (UpSampling (None, 32, 32, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_116 (Conv2D) (None, 32, 32, 16) 1168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_51 (UpSampling (None, 64, 64, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_117 (Conv2D) (None, 64, 64, 16) 2320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_52 (UpSampling (None, 128, 128, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_118 (Conv2D) (None, 128, 128, 32) 4640

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_53 (UpSampling (None, 256, 256, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_119 (Conv2D) (None, 256, 256, 1) 289

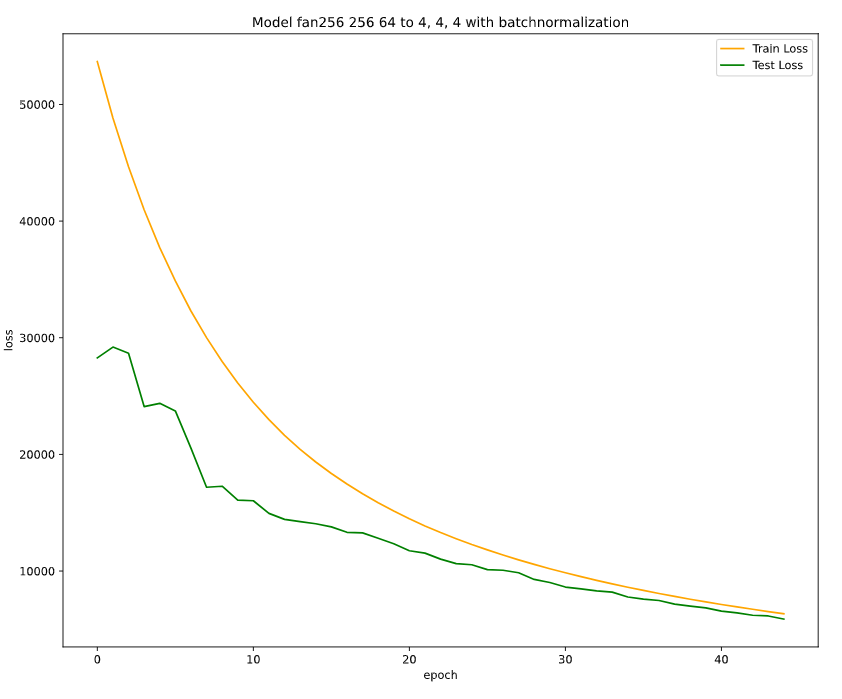
=================================================================

Total params: 72,801

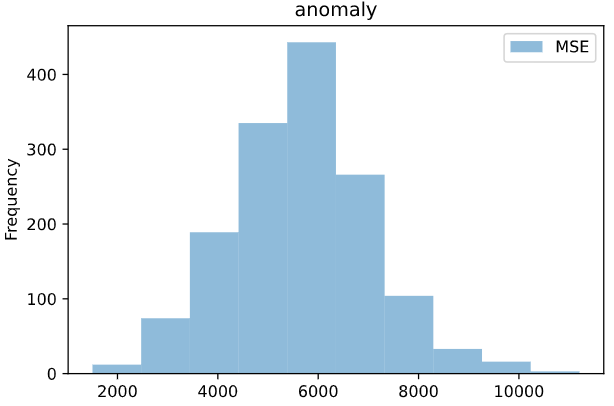
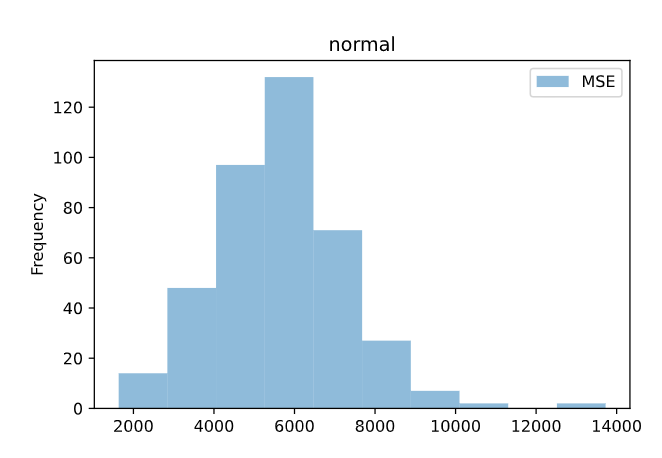
Trainable params: 72,545

Non-trainable params: 256

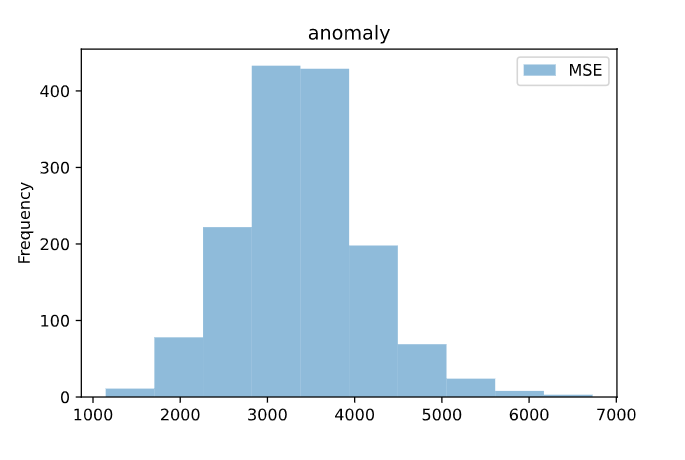
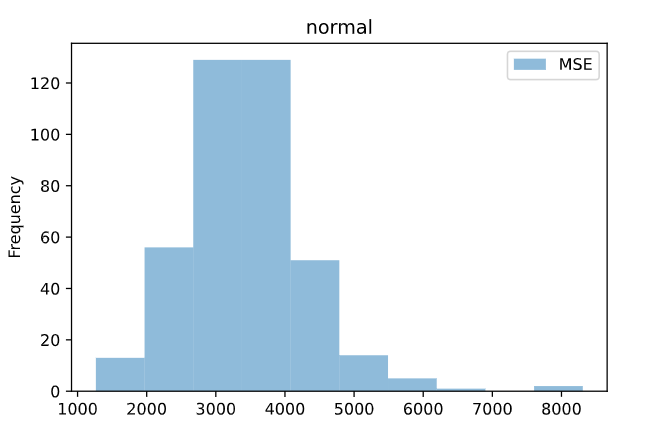
Nous pouvons constater un impact direct sur la MSE (Loss) qui passe d’un score <0 à une valeur de plus de 10 000.



Toutefois si l’effet est direct sur la Loss, il n’y a aucun effet sur les résultats, les valeurs des MSE étant toujours réparties de manière identique entre les normaux et anormaux.



Après 20%(1h) d'entraînement supplémentaire:



Le temps d'entraînement affecte toujours de manière presque égale les MSE normales et anormales, nous pouvons donc conclure que le type de modèle utilisé n’est pas adapté à notre objectif

Layer (type) Output Shape Param #

=================================================================

conv2d\_130 (Conv2D) (None, 256, 256, 4) 104

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_131 (Conv2D) (None, 256, 256, 4) 148

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_58 (Batc (None, 256, 256, 4) 16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_58 (MaxPooling (None, 128, 128, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_132 (Conv2D) (None, 128, 128, 8) 296

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_133 (Conv2D) (None, 128, 128, 8) 584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_59 (Batc (None, 128, 128, 8) 32

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_59 (MaxPooling (None, 64, 64, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_134 (Conv2D) (None, 64, 64, 16) 1168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_135 (Conv2D) (None, 64, 64, 16) 2320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_60 (Batc (None, 64, 64, 16) 64

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_60 (MaxPooling (None, 32, 32, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_136 (Conv2D) (None, 32, 32, 32) 4640

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_61 (Batc (None, 32, 32, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_61 (MaxPooling (None, 16, 16, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_137 (Conv2D) (None, 16, 16, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_62 (Batc (None, 16, 16, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_62 (MaxPooling (None, 8, 8, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_138 (Conv2D) (None, 8, 8, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_58 (UpSampling (None, 16, 16, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_139 (Conv2D) (None, 16, 16, 32) 18464

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_59 (UpSampling (None, 32, 32, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_140 (Conv2D) (None, 32, 32, 16) 4624

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_60 (UpSampling (None, 64, 64, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_141 (Conv2D) (None, 64, 64, 8) 1160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_61 (UpSampling (None, 128, 128, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_142 (Conv2D) (None, 128, 128, 4) 292

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_62 (UpSampling (None, 256, 256, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

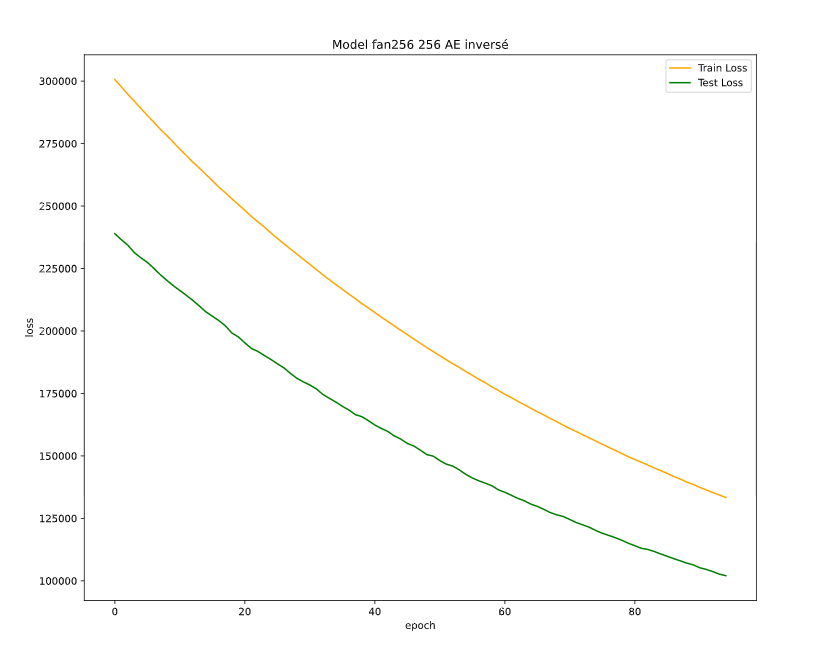
conv2d\_143 (Conv2D) (None, 256, 256, 1) 37

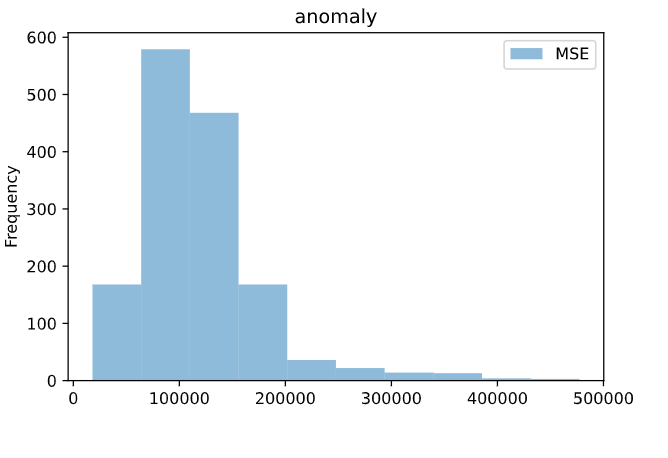
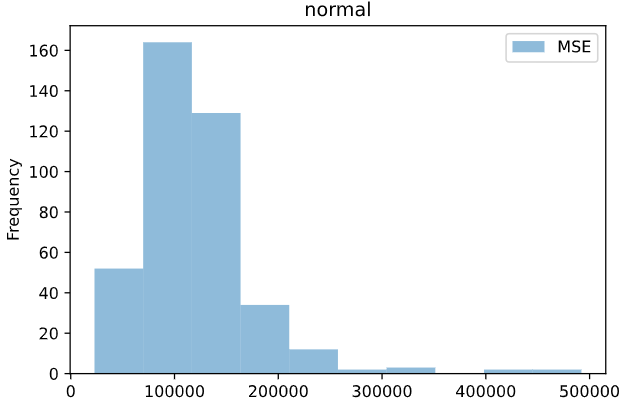
=================================================================

Total params: 89,757

Trainable params: 89,509

Non-trainable params: 248





### Essais Classificateur d’IDs

**slider**

Layer (type) Output Shape Param #

=================================================================

conv2d\_31 (Conv2D) (None, 256, 256, 64) 640

max\_pooling2d\_14 (MaxPooling (None, 128, 128, 64) 0

conv2d\_32 (Conv2D) (None, 128, 128, 32) 18464

max\_pooling2d\_15 (MaxPooling (None, 64, 64, 32) 0

conv2d\_33 (Conv2D) (None, 64, 64, 16) 4624

max\_pooling2d\_16 (MaxPooling (None, 32, 32, 16) 0

conv2d\_34 (Conv2D) (None, 32, 32, 8) 1160

max\_pooling2d\_17 (MaxPooling (None, 16, 16, 8) 0

conv2d\_35 (Conv2D) (None, 16, 16, 4) 292

max\_pooling2d\_18 (MaxPooling (None, 8, 8, 4) 0

flatten\_4 (Flatten) (None, 256) 0

dense\_12 (Dense) (None, 256) 65792

dropout\_8 (Dropout) (None, 256) 0

leaky\_re\_lu\_8 (LeakyReLU) (None, 256) 0

dense\_13 (Dense) (None, 256) 65792

dropout\_9 (Dropout) (None, 256) 0

leaky\_re\_lu\_9 (LeakyReLU) (None, 256) 0

dense\_14 (Dense) (None, 4) 1028

=================================================================

Total params: 157,792

Trainable params: 132,612

Non-trainable params: 25,180

74 erreurs sur la classe normal soit 18.50 %

528 erreurs sur la classe anomaly soit 59.33 %

détails fichiers normaux

precision recall f1-score support

0 0.89 0.33 0.48 100

1 0.60 0.96 0.74 100

2 0.98 0.98 0.98 100

3 0.97 0.99 0.98 100

accuracy 0.81 400

macro avg 0.86 0.81 0.79 400

weighted avg 0.86 0.81 0.79 400

détails fichiers anomaly

precision recall f1-score support

0 0.06 0.01 0.01 356

1 0.68 0.37 0.48 267

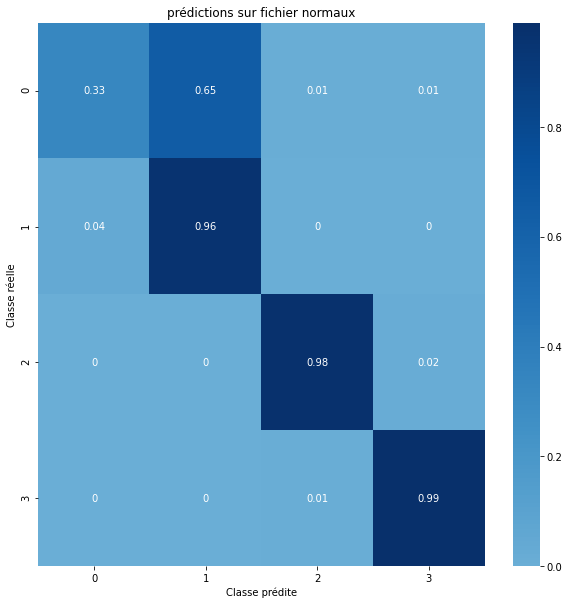
2 0.70 0.97 0.81 178

3 0.19 0.98 0.32 89

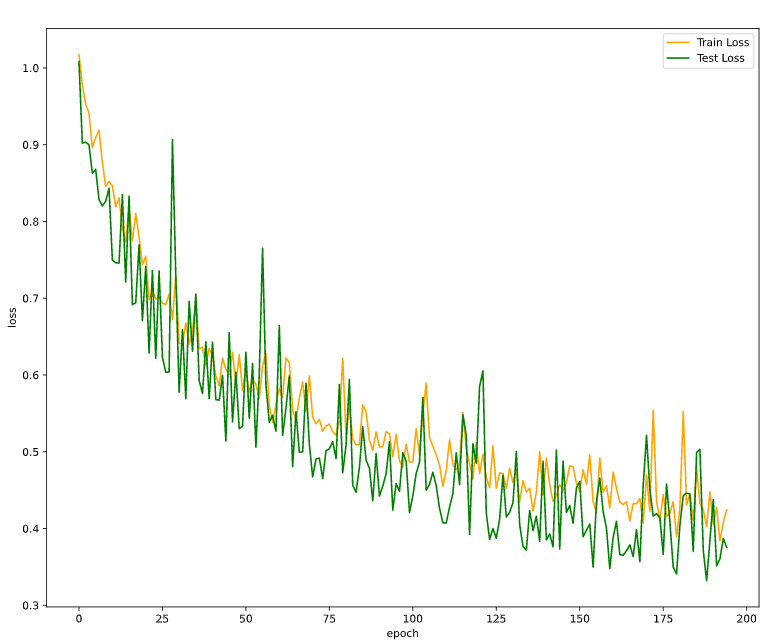
accuracy 0.41 890

macro avg 0.41 0.58 0.41 890

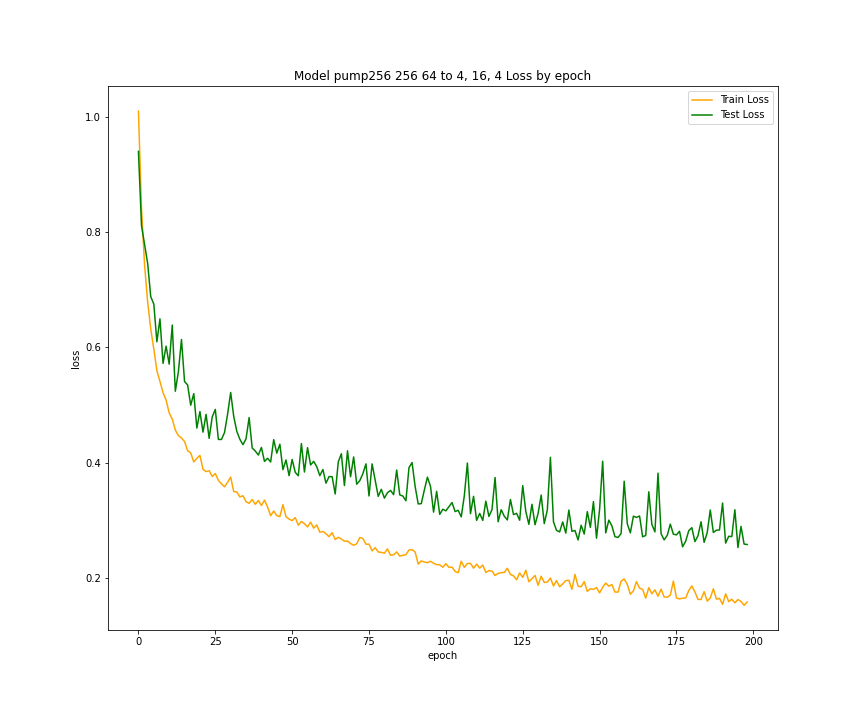
weighted avg 0.39 0.41 0.34 890

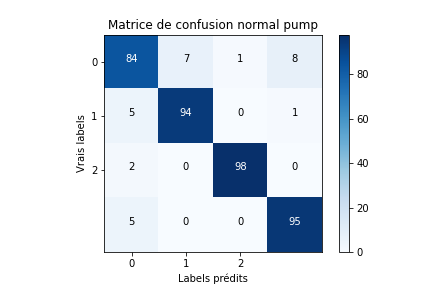
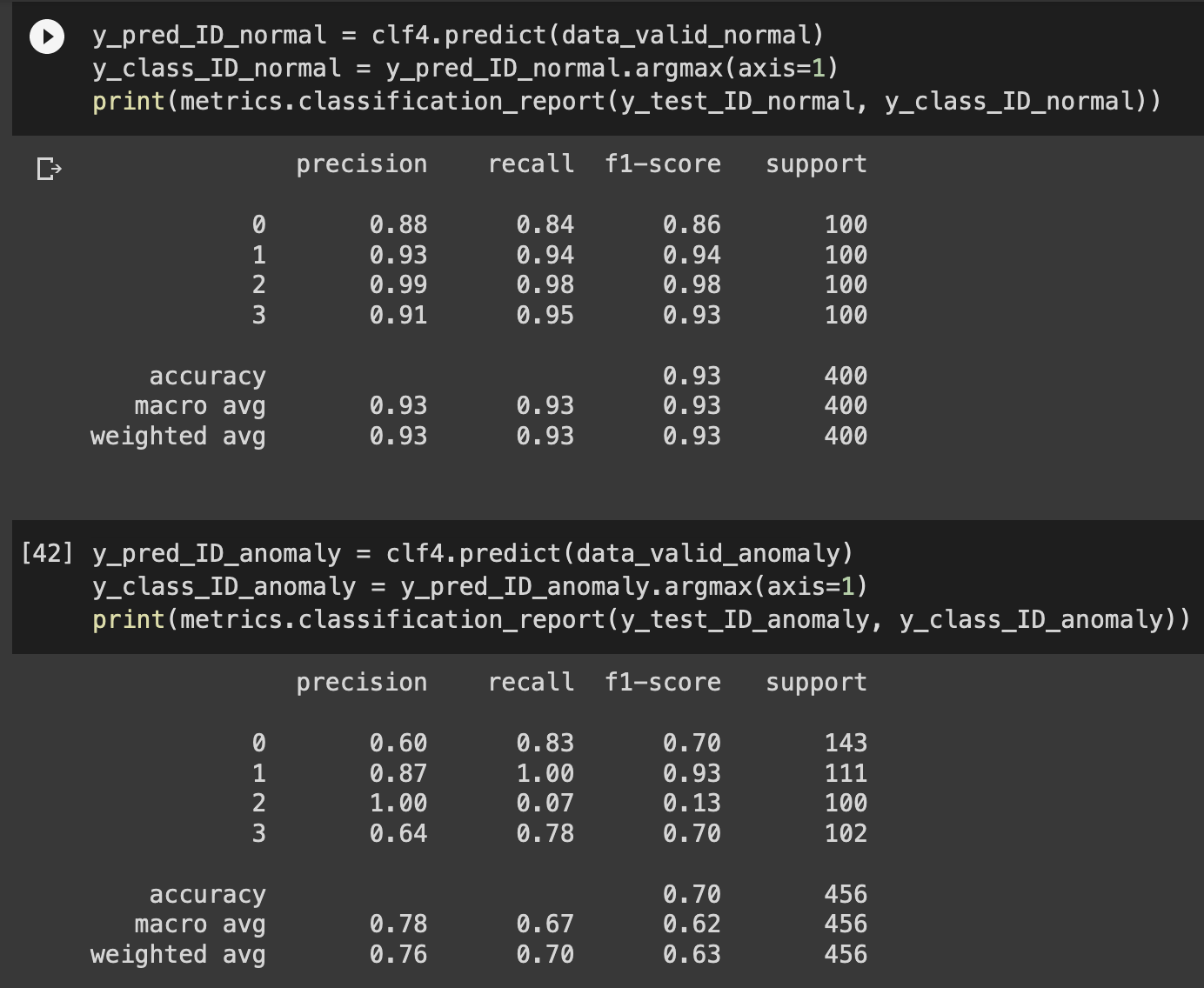


On peut voir que le modèle distingue bien les fichiers normaux des classes 1/2/3, mais a plus de difficultés sur la classe 0. Toutefois il a tellement de facilité à distinguer les classes ⅔ qu’il ne se trompe pas non plus sur les anomalies de ces classes.



Avec 200 EPOCHS supplémentaires le modèle se sur-entraine et se renforce sur ses points forts

**PUMP:**



### 

### 

### 

Essais en LTSM sur Slider

Layer (type) Output Shape Param #

=================================================================

conv2d\_31 (Conv2D) (None, 256, 256, 64) 640

max\_pooling2d\_14 (MaxPooling (None, 128, 128, 64) 0

conv2d\_32 (Conv2D) (None, 128, 128, 32) 18464

max\_pooling2d\_15 (MaxPooling (None, 64, 64, 32) 0

conv2d\_33 (Conv2D) (None, 64, 64, 16) 4624

max\_pooling2d\_16 (MaxPooling (None, 32, 32, 16) 0

conv2d\_34 (Conv2D) (None, 32, 32, 8) 1160

max\_pooling2d\_17 (MaxPooling (None, 16, 16, 8) 0

conv2d\_35 (Conv2D) (None, 16, 16, 4) 292

max\_pooling2d\_18 (MaxPooling (None, 8, 8, 4) 0

conv2d\_1 (Conv2D) (None, 8, 8, 1) 37

reshape\_1 (Reshape) (None, 8, 8) 0

lstm\_3 (LSTM) (None, 8, 64) 18688

dropout\_10 (Dropout) (None, 8, 64) 0

leaky\_re\_lu\_10 (LeakyReLU) (None, 8, 64) 0

lstm\_4 (LSTM) (None, 256) 328704

dropout\_11 (Dropout) (None, 256) 0

leaky\_re\_lu\_11 (LeakyReLU) (None, 256) 0

dense\_12 (Dense) (None, 64) 16448

dropout\_12 (Dropout) (None, 64) 0

leaky\_re\_lu\_12 (LeakyReLU) (None, 64) 0

dense\_13 (Dense) (None, 256) 16640

dropout\_13 (Dropout) (None, 256) 0

leaky\_re\_lu\_13 (LeakyReLU) (None, 256) 0

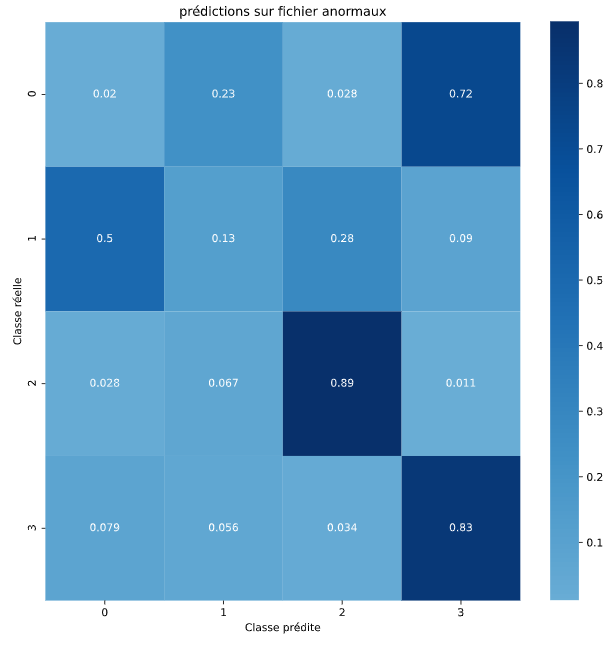
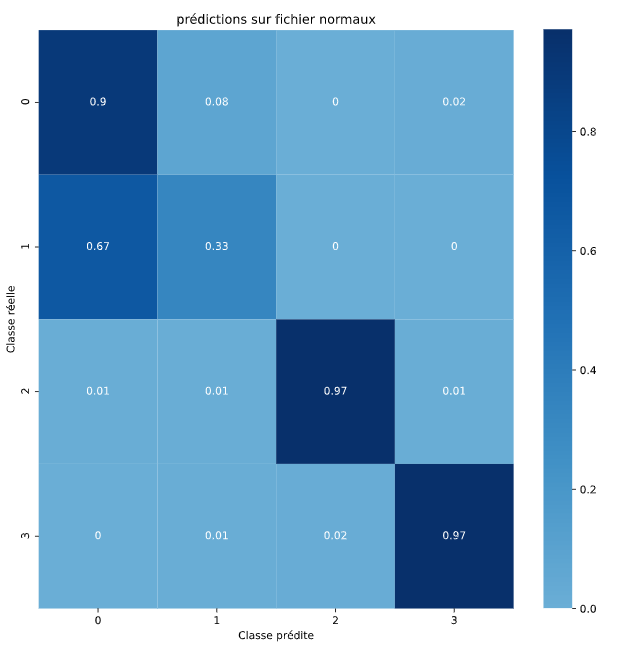
dense\_14 (Dense) (None, 4) 1028

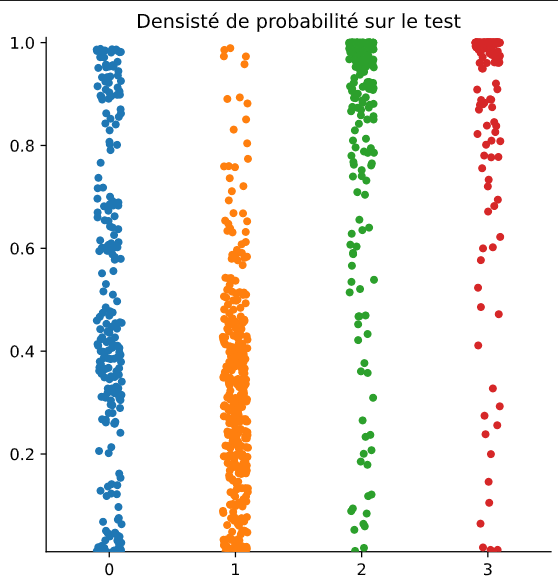
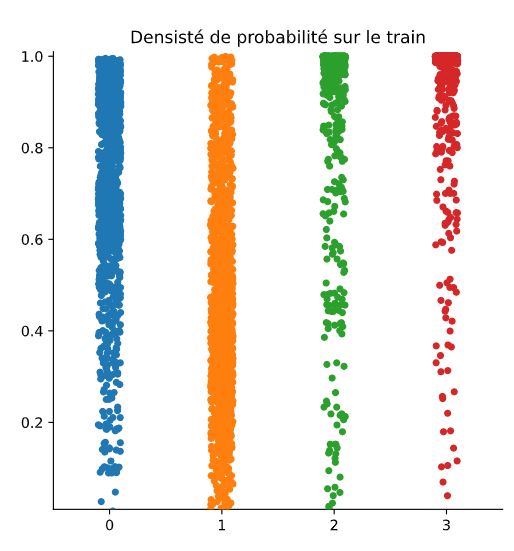
=================================================================

Total params: 406,725

Trainable params: 381,545

Non-trainable params: 25,180





Layer (type) Output Shape Param #

=================================================================

Encoder

conv2d\_3 (Conv2D) (None, 8, 8, 1) 37

reshape\_3 (Reshape) (None, 8, 8) 0

lstm\_7 (LSTM) (None, 8, 64) 18688

batch\_normalization (BatchNo (None, 8, 64) 256

leaky\_re\_lu\_17 (LeakyReLU) (None, 8, 64) 0

lstm\_8 (LSTM) (None, 256) 328704

batch\_normalization\_1 (Batch (None, 256) 1024

leaky\_re\_lu\_18 (LeakyReLU) (None, 256) 0

dense\_17 (Dense) (None, 256) 65792

dropout\_17 (Dropout) (None, 256) 0

leaky\_re\_lu\_19 (LeakyReLU) (None, 256) 0

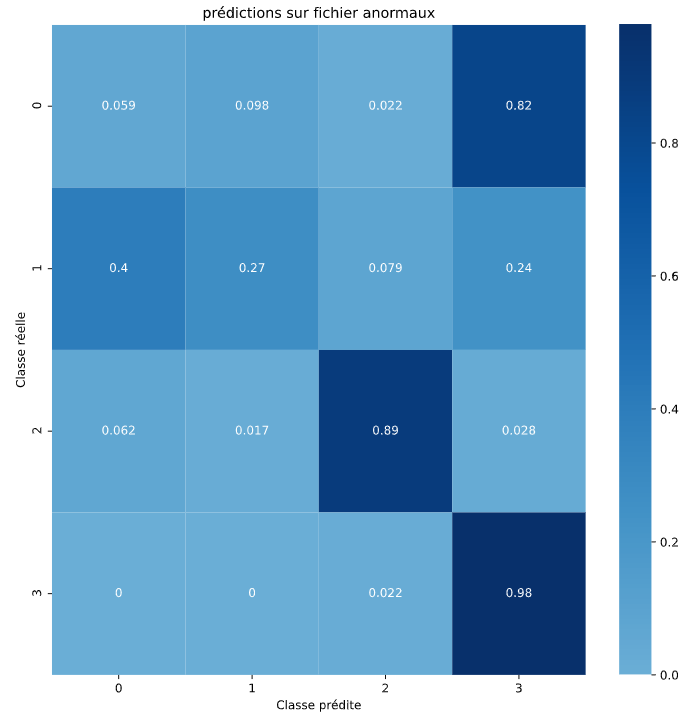
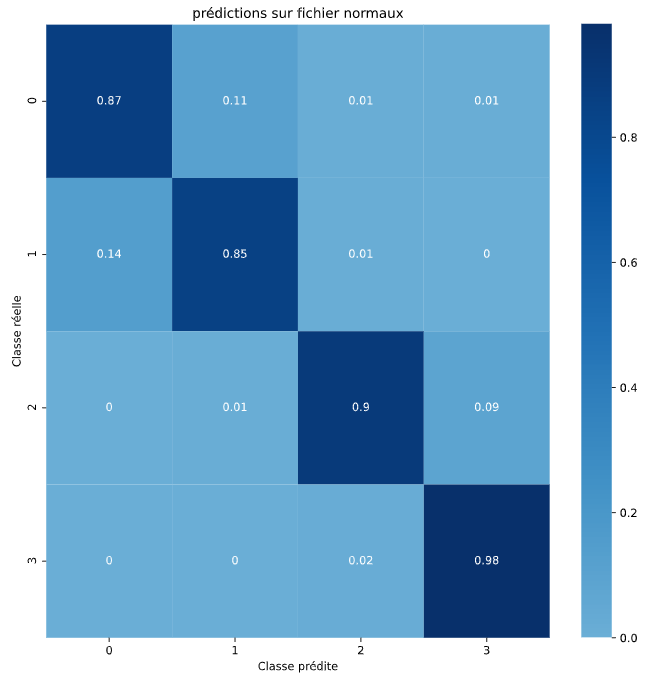
dense\_18 (Dense) (None, 4) 1028

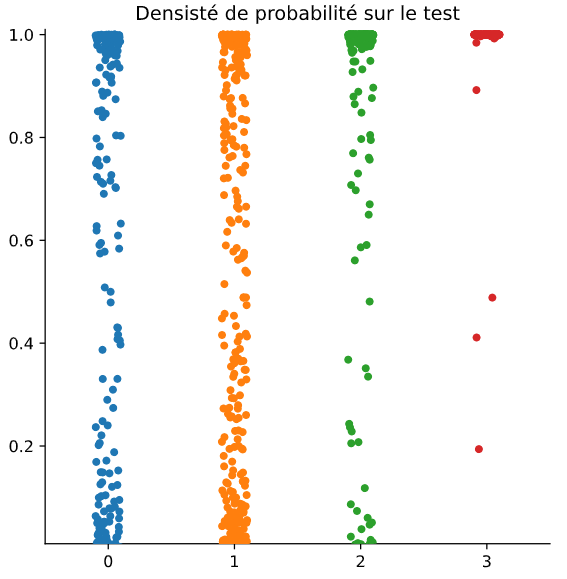
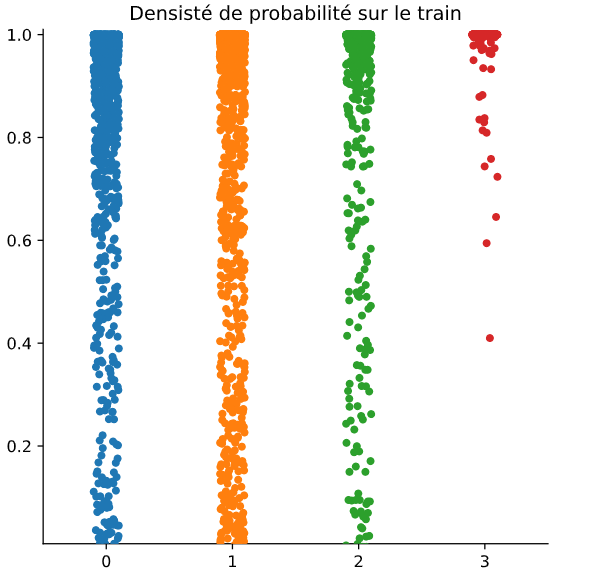
=================================================================

Total params: 440,709

Trainable params: 414,889

Non-trainable params: 25,820





LTSM sans AE

Layer (type) Output Shape Param #

=================================================================

lstm\_38 (LSTM) (None, 256, 64) 82176

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_33 (Batc (None, 256, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

leaky\_re\_lu\_48 (LeakyReLU) (None, 256, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_39 (LSTM) (None, 64) 33024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_34 (Batc (None, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

leaky\_re\_lu\_49 (LeakyReLU) (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_26 (Dense) (None, 64) 4160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_13 (Dropout) (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

leaky\_re\_lu\_50 (LeakyReLU) (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_27 (Dense) (None, 4) 260

=================================================================

Total params: 120,132

Trainable params: 119,876

Non-trainable params: 256

68 erreurs sur la classe normal soit 17.00 %

677 erreurs sur la classe anomaly soit 76.07 %

détails fichiers normaux

precision recall f1-score support

0 0.86 0.94 0.90 100

1 0.65 0.98 0.78 100

2 1.00 0.74 0.85 100

3 1.00 0.66 0.80 100

accuracy 0.83 400

macro avg 0.88 0.83 0.83 400

weighted avg 0.88 0.83 0.83 400

détails fichiers anomaly

precision recall f1-score support

0 0.16 0.11 0.13 356

1 0.26 0.63 0.37 267

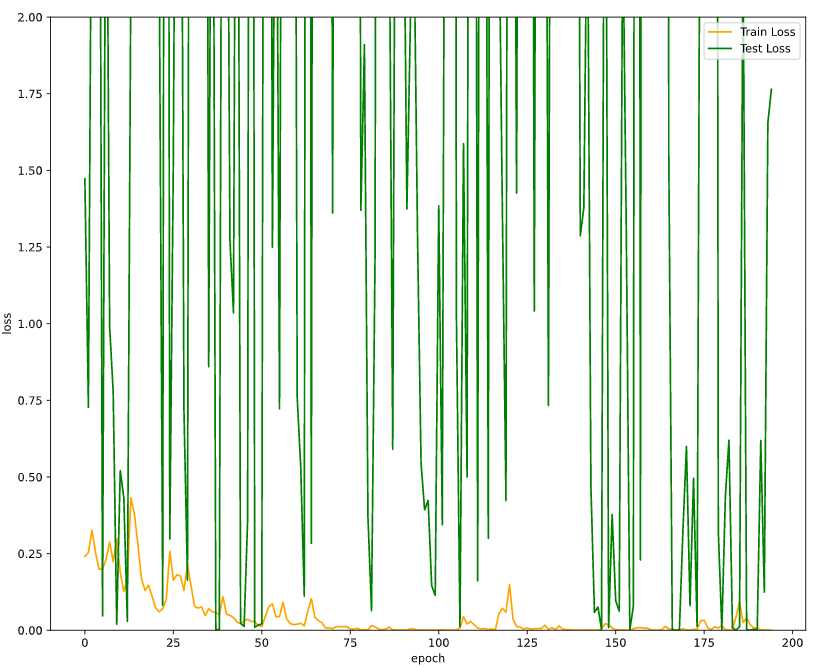
2 0.00 0.00 0.00 178

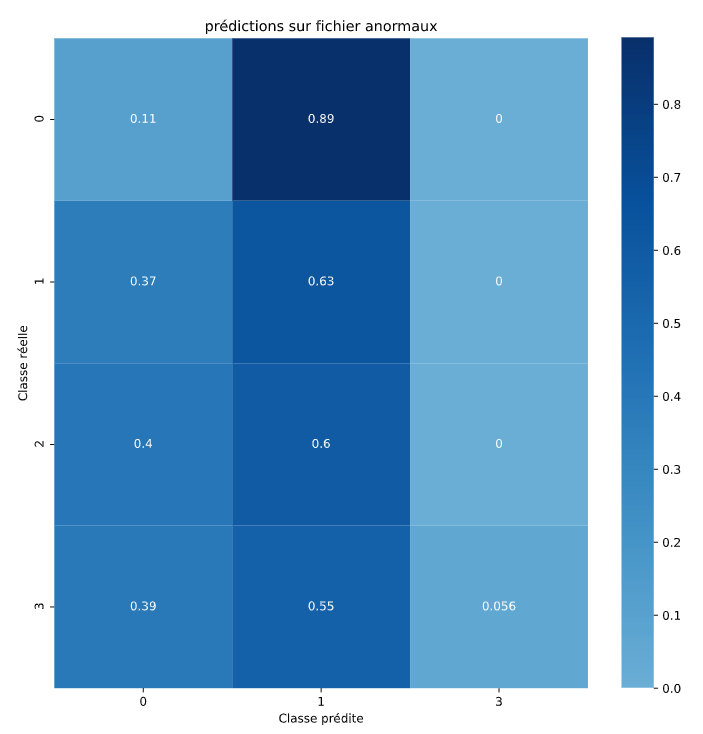
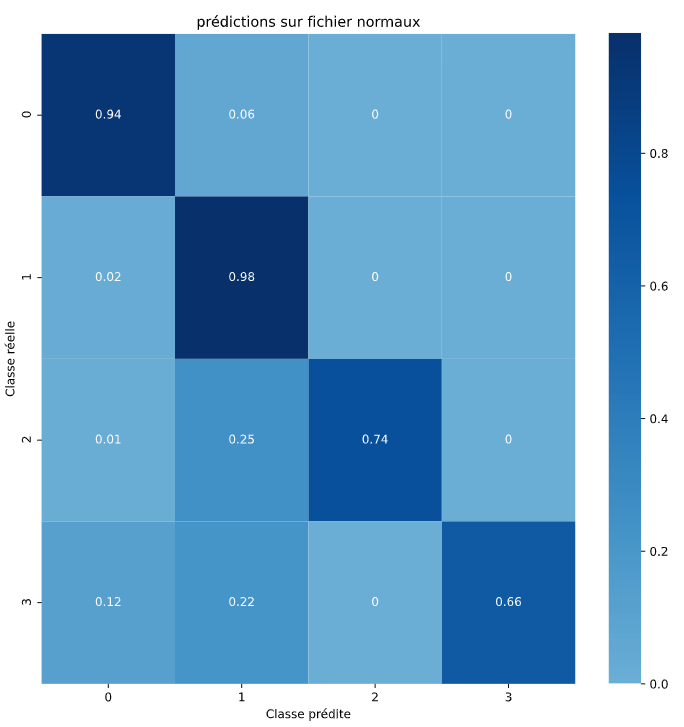
3 1.00 0.06 0.11 89

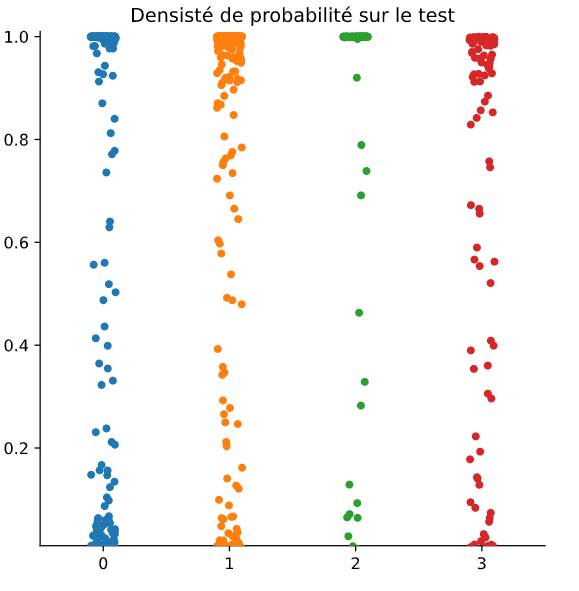
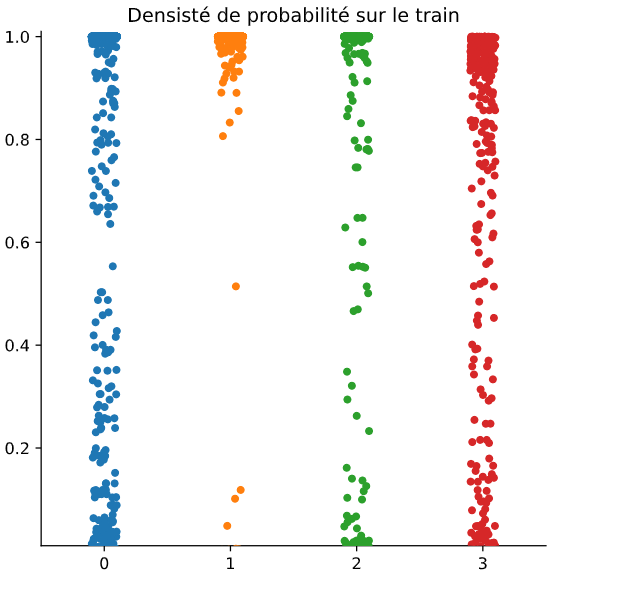
accuracy 0.24 890

macro avg 0.36 0.20 0.15 890

weighted avg 0.24 0.24 0.17 890









GRU sans AE

Layer (type) Output Shape Param #

=================================================================

gru (GRU) (None, 256, 32) 27840

leaky\_re\_lu\_57 (LeakyReLU) (None, 256, 32) 0

gru\_1 (GRU) (None, 32) 6336

leaky\_re\_lu\_58 (LeakyReLU) (None, 32) 0

dense\_32 (Dense) (None, 32) 1056

dropout\_16 (Dropout) (None, 32) 0

leaky\_re\_lu\_59 (LeakyReLU) (None, 32) 0

dense\_33 (Dense) (None, 4) 132

=================================================================

Total params: 35,364

Trainable params: 35,364

Non-trainable params: 0

12 erreurs sur la classe normal soit 3.00 %

447 erreurs sur la classe anomaly soit 50.22 %

détails fichiers normaux

precision recall f1-score support

0 0.94 0.94 0.94 100

1 0.94 0.94 0.94 100

2 1.00 1.00 1.00 100

3 1.00 1.00 1.00 100

accuracy 0.97 400

macro avg 0.97 0.97 0.97 400

weighted avg 0.97 0.97 0.97 400

détails fichiers anomaly

precision recall f1-score support

0 0.09 0.02 0.04 356

1 0.63 0.67 0.65 267

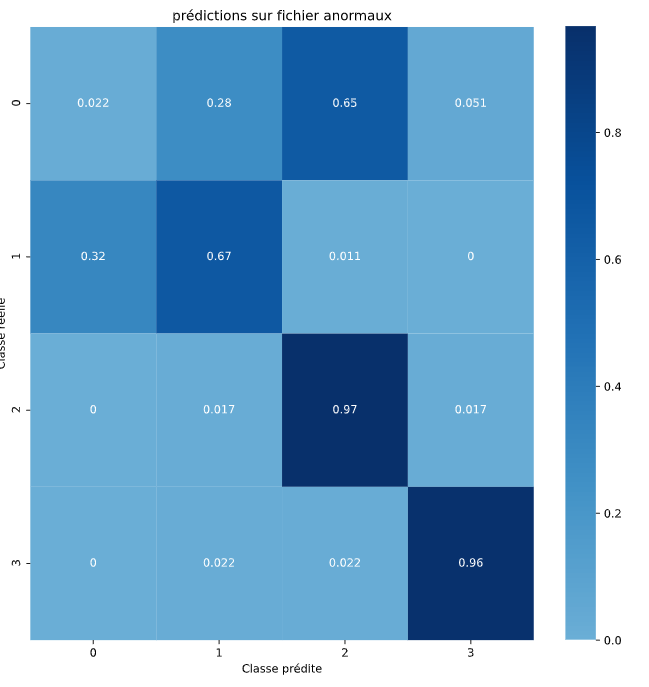
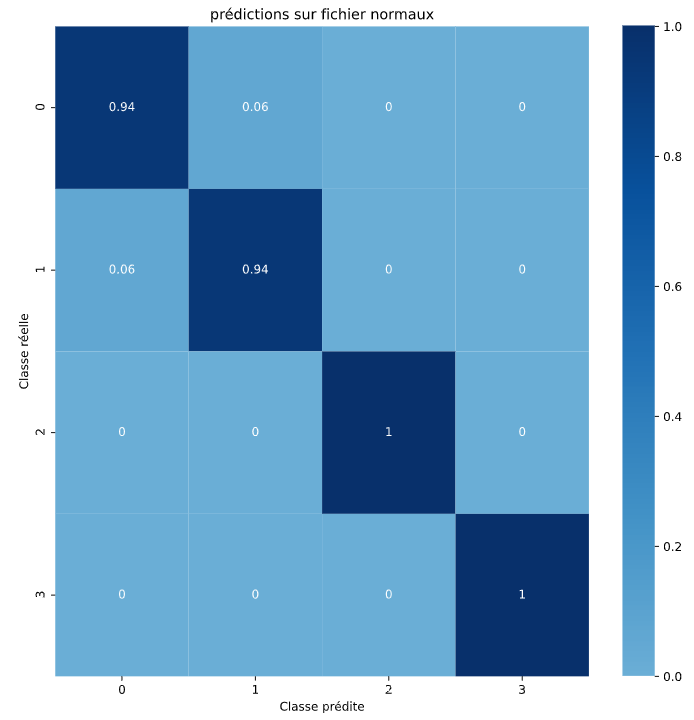
2 0.42 0.97 0.59 178

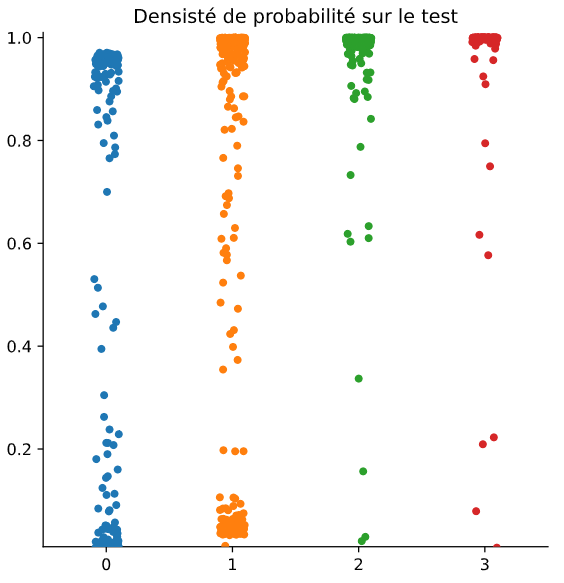
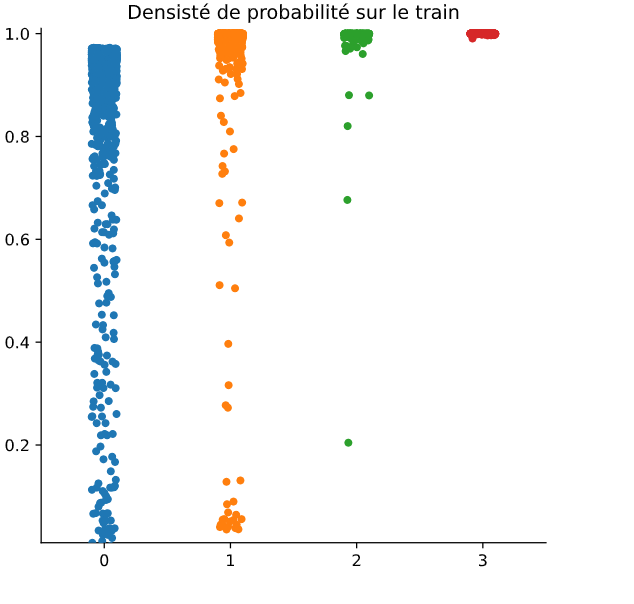
3 0.80 0.96 0.87 89

accuracy 0.50 890

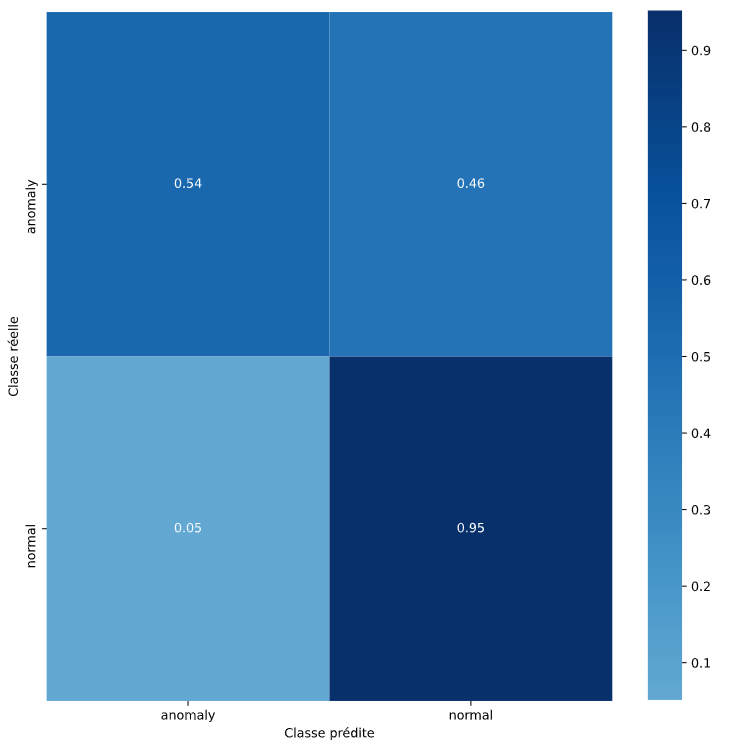
macro avg 0.49 0.65 0.54 890

weighted avg 0.39 0.50 0.41 890









GRU 64 sans AE

Layer (type) Output Shape Param #

=================================================================

gru (GRU) (None, 256, 64) 61824

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization (BatchNo (None, 256, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

leaky\_re\_lu (LeakyReLU) (None, 256, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

gru\_1 (GRU) (None, 64) 24960

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_1 (Batch (None, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

leaky\_re\_lu\_1 (LeakyReLU) (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 64) 4160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

leaky\_re\_lu\_2 (LeakyReLU) (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 4) 260

=================================================================

Total params: 91,716

Trainable params: 91,460

Non-trainable params: 256

2 erreurs sur la classe normal soit 0.50 %

506 erreurs sur la classe anomaly soit 56.85 %

precision recall f1-score support

0 0.99 0.99 0.99 100

1 0.99 0.99 0.99 100

2 1.00 1.00 1.00 100

3 1.00 1.00 1.00 100

accuracy 0.99 400

macro avg 0.99 0.99 0.99 400

weighted avg 0.99 0.99 0.99 400

détails fichiers anomaly

precision recall f1-score support

0 0.46 0.16 0.24 356

1 0.38 0.82 0.52 267

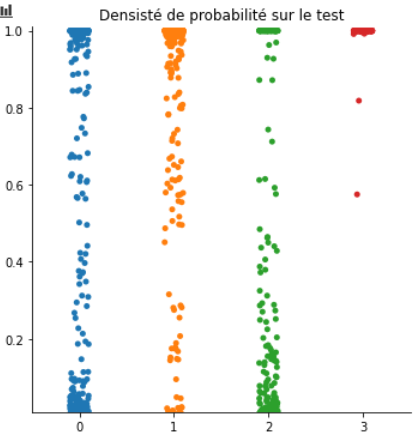
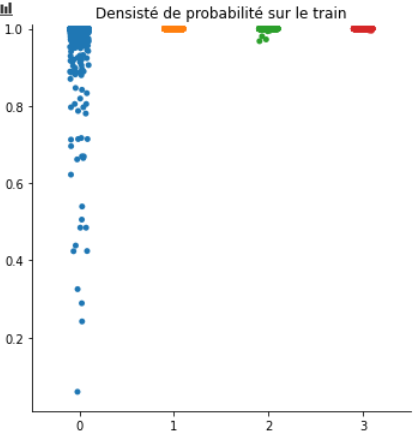
2 1.00 0.10 0.18 178

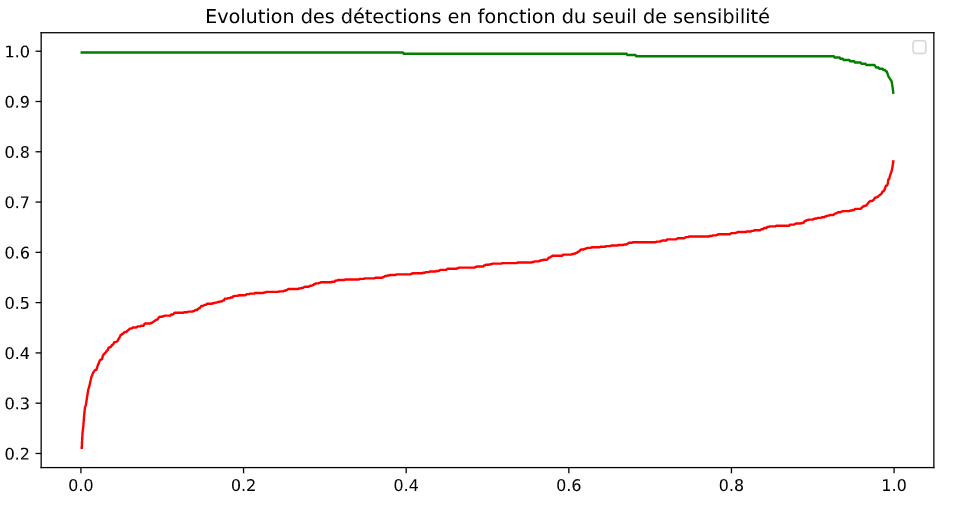
3 0.51 1.00 0.67 89

accuracy 0.43 890

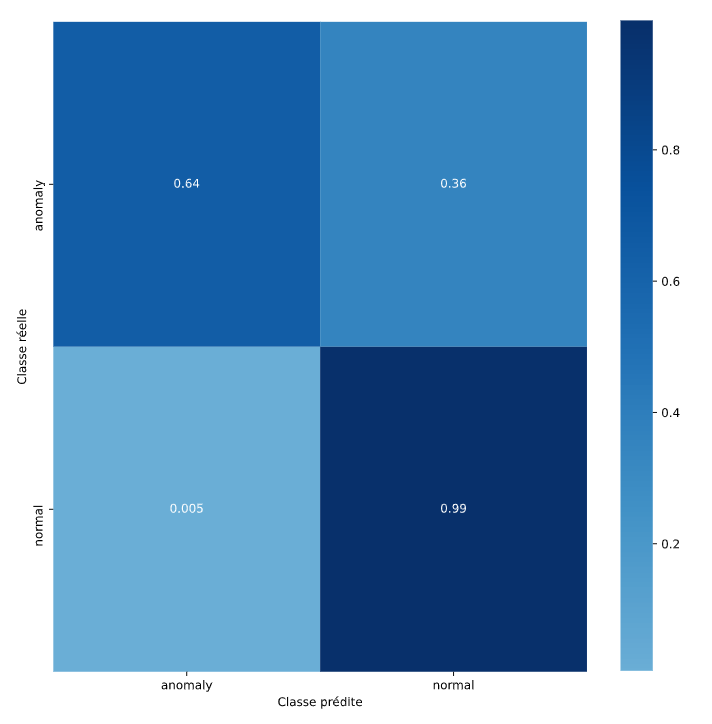
macro avg 0.59 0.52 0.41 890

weighted avg 0.55 0.43 0.36 890





seuil\_ID=np.array([0.4,0.95,0.95,0.95])



PROOF OF CONCEPT

Premier modèle de classification hybride (AE/Classificateur normal/anormal)

Layer (type) Output Shape Param #

=================================================================

conv2d\_11 (Conv2D) (None, 256, 256, 64) 640

max\_pooling2d\_5 (MaxPooling2 (None, 128, 128, 64) 0

conv2d\_12 (Conv2D) (None, 128, 128, 32) 18464

max\_pooling2d\_6 (MaxPooling2 (None, 64, 64, 32) 0

conv2d\_13 (Conv2D) (None, 64, 64, 16) 4624

max\_pooling2d\_7 (MaxPooling2 (None, 32, 32, 16) 0

conv2d\_14 (Conv2D) (None, 32, 32, 8) 1160

max\_pooling2d\_8 (MaxPooling2 (None, 16, 16, 8) 0

conv2d\_15 (Conv2D) (None, 16, 16, 4) 292

max\_pooling2d\_9 (MaxPooling2 (None, 8, 8, 4) 0

flatten (Flatten) (None, 256) 0

dense (Dense) (None, 1024) 263168

dropout (Dropout) (None, 1024) 0

leaky\_re\_lu (LeakyReLU) (None, 1024) 0

dense\_1 (Dense) (None, 1024) 1049600

dropout\_1 (Dropout) (None, 1024) 0

leaky\_re\_lu\_1 (LeakyReLU) (None, 1024) 0

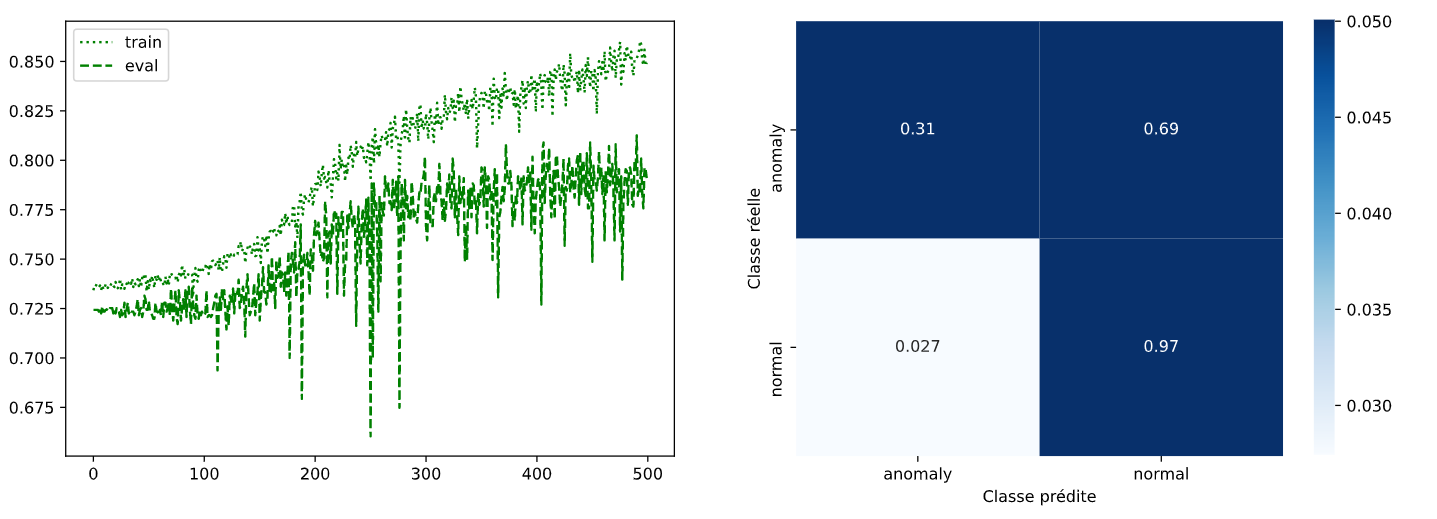
dense\_2 (Dense) (None, 1) 1025

=================================================================

Total params: 1,338,973

Trainable params: 1,313,793

Non-trainable params: 25,180



precision recall f1-score support

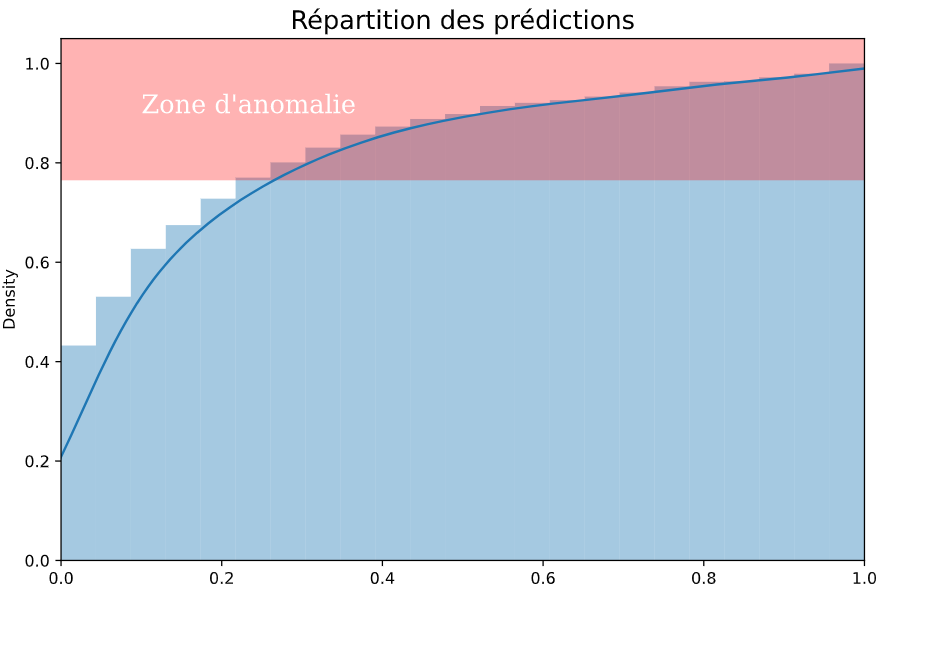
anomaly 0.81 0.31 0.45 306

normal 0.79 0.97 0.87 804

accuracy 0.79 1110

macro avg 0.80 0.64 0.66 1110

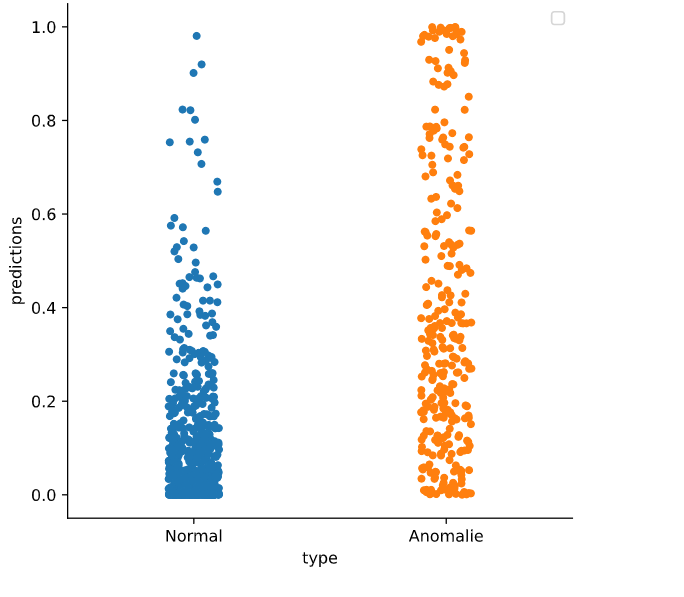
weighted avg 0.80 0.79 0.76 1110

On peut constater un Overtraining à partir de l’EPOCH 300.Le modèle manque de sensibilité et ne détecte que 30% des anomalies.

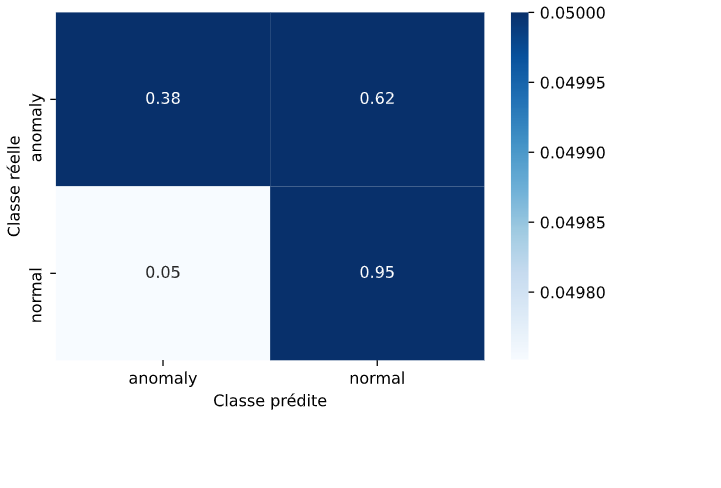
Grace à la courbe de sortie du réseau nous pouvons constater deux choses :

1. Le réseau n’a que peu de certitude sur les anomalies, et une très bonne certitude sur le normal. En effet le label 0 (normal) atteint dès le début 40% des prédictions. Le reste des prédictions est presque uniformément réparti jusqu’à 1 (anomalie). Le réseau a donc une marge de progression pour réaliser une réelle catégorisation, des étapes d’amélioration sont encore nécessaires.
2. Nous pouvons toutefois moduler sa sensibilité assez “finement”, pour espérer détecter plus d’anomalies en risquant également plus de faux positifs.

Réglage de la sensibilité



En regardant la répartition par type nous pouvons voir qu’un autre réglage de la sensibilité pourrait être à 0.4 au lieu de 0.5. Et le faire nous permet en effet de détecter 7% d’anomalies supplémentaires mais avec 2fois plus de fausses détections (5% au lieu de 2.5%)



Essais avec un modèle moins complexe

Layer (type) Output Shape Param #

=================================================================

conv2d\_11 (Conv2D) (None, 256, 256, 64) 640

max\_pooling2d\_5 (MaxPooling2 (None, 128, 128, 64) 0

conv2d\_12 (Conv2D) (None, 128, 128, 32) 18464

max\_pooling2d\_6 (MaxPooling2 (None, 64, 64, 32) 0

conv2d\_13 (Conv2D) (None, 64, 64, 16) 4624

max\_pooling2d\_7 (MaxPooling2 (None, 32, 32, 16) 0

conv2d\_14 (Conv2D) (None, 32, 32, 8) 1160

max\_pooling2d\_8 (MaxPooling2 (None, 16, 16, 8) 0

conv2d\_15 (Conv2D) (None, 16, 16, 4) 292

max\_pooling2d\_9 (MaxPooling2 (None, 8, 8, 4) 0

flatten\_1 (Flatten) (None, 256) 0

dense\_3 (Dense) (None, 256) 65792

dropout\_2 (Dropout) (None, 256) 0

leaky\_re\_lu\_2 (LeakyReLU) (None, 256) 0

dense\_4 (Dense) (None, 256) 65792

dropout\_3 (Dropout) (None, 256) 0

leaky\_re\_lu\_3 (LeakyReLU) (None, 256) 0

dense\_5 (Dense) (None, 1) 257

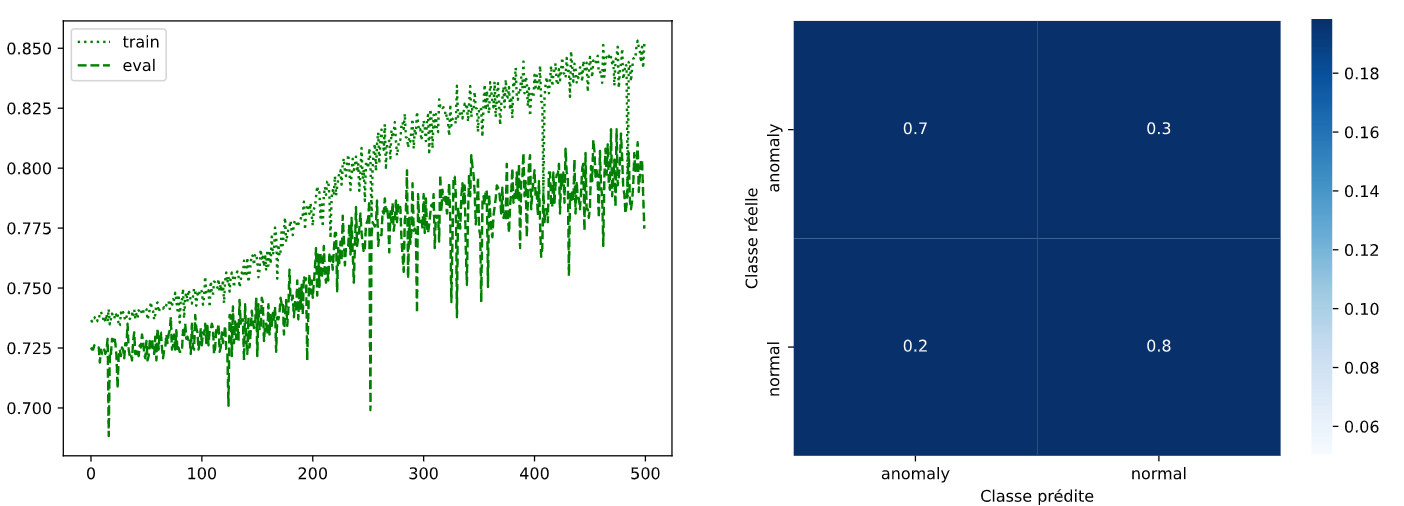
=================================================================

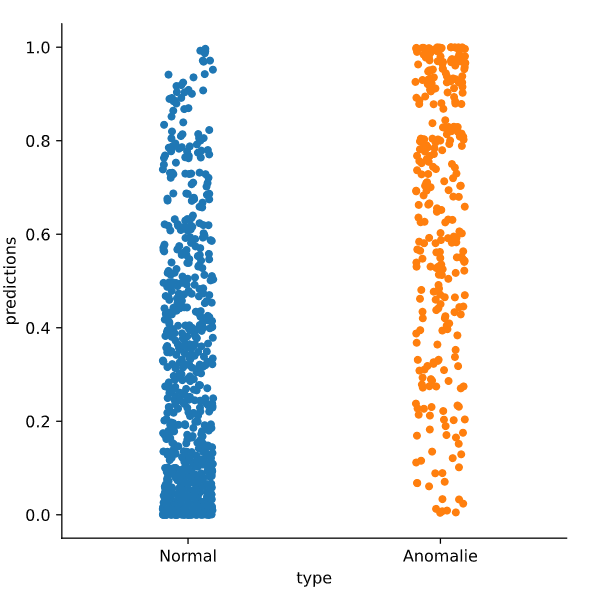
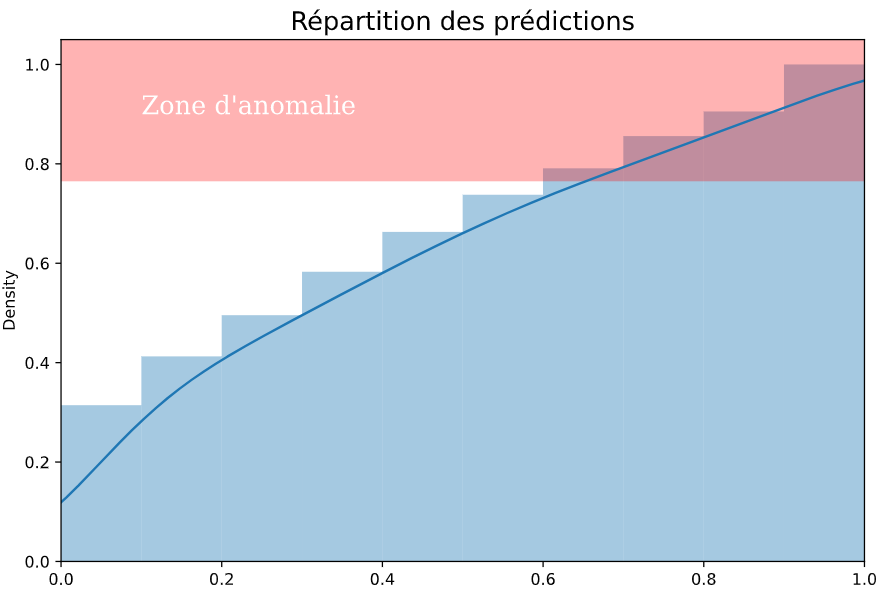
Total params: 157,021

Trainable params: 131,841

Non-trainable params: 25,180

Resultats après 500EPOCHS





La courbe de décision du modèle est plus linéaire, ce qui n’est pas l’idéal, c’est sans surprise que le modèle prédit 70% d’anomalie, tout en se trompant sur 20% des normaux.Même si la courbe d’apprentissage semble atteindre un plafond, nous allons pousser l'entraînement pour essayer de diminuer cette sensibilité.

Resultats après 600EPOCHS

precision recall f1-score support

anomaly 0.70 0.54 0.61 306

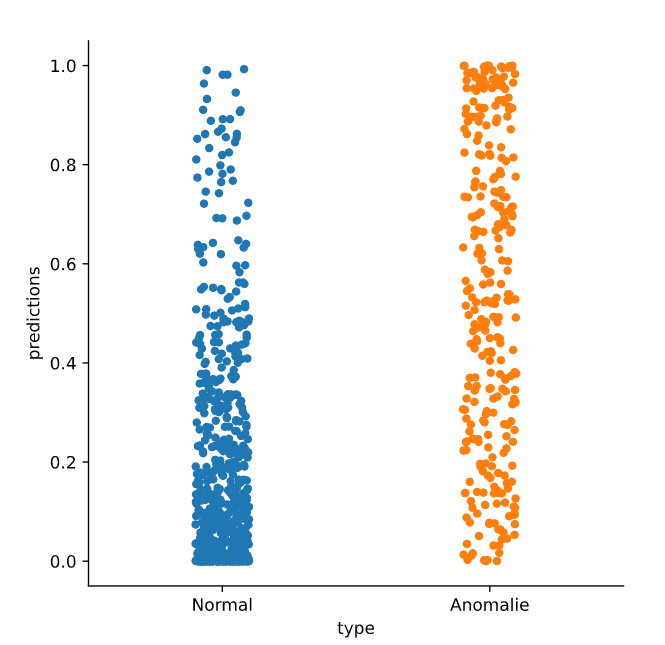
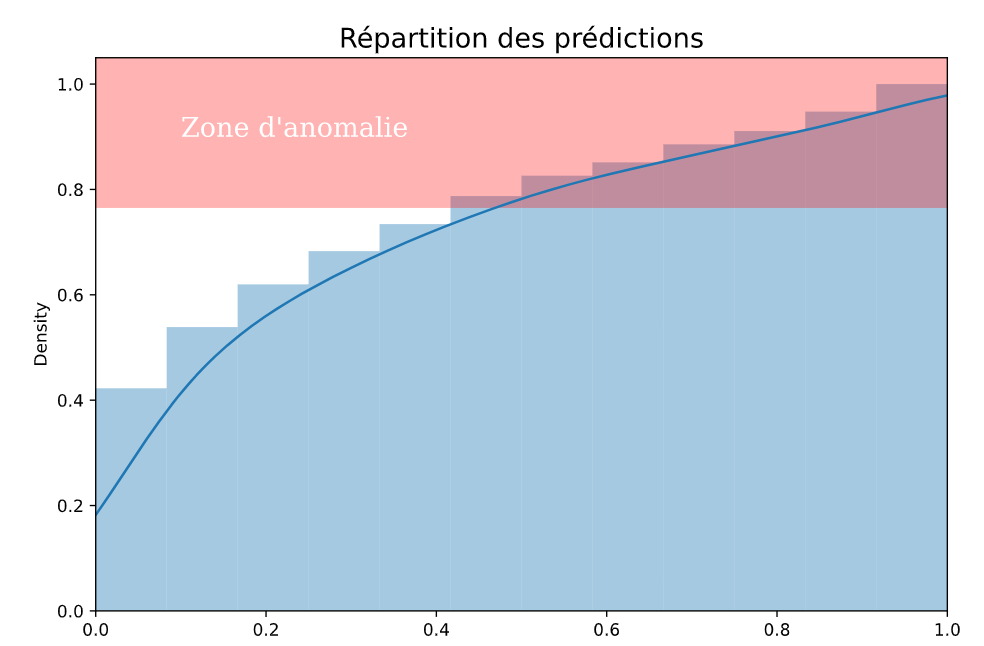
normal 0.84 0.91 0.87 804

accuracy 0.81 1110

macro avg 0.77 0.73 0.74 1110

weighted avg 0.80 0.81 0.80 1110





Nous pouvons constater l’influence de l'entraînement supplémentaire sur la capacité du modèle à correctement identifier les cycles normaux, même si cela s’est fait au détriment de la détection des anomalies. En effet avec 20% d’entrainement en plus, nous avons divisé par deux la sensibilité du modèle. La courbe de décision se rapproche également du S idéal le premier plateau ayant dépassé les 40% de densité de prédiction.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| architecture AE  compilation | “ simple 1” | “simple 2” | “simple 3”  double CONV | “simple 4” |
| “ compile 1” | Yossef 60 epochs / stoper a 11 epoch par callback | Yossef 100 epochs |  |  |
| “ compile 2” | Yossef 60 epochs | Yossef 60 epochs | Corentin  100EPOCHS  IDEM modèle simple |  |
| “ compile 3” | Yossef 50 epochs | Astrid : 140 epochs / MAUVAIS |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| architecture AE  compilation | “complex128” | Modèle 5  Batch normalization | Modèle 6  Batch normalization  +kernel non neg |
| “ compile 1” |  |  |  |
| “ compile 2” |  | Corentin  100EPOCHS  IDEM resultats | Corentin  60EPOCHS  IDEM resultats |
| “ compile 3” | Astrid en cours | yossef 50 epoch |  |

|  |  |  |  |
| --- | --- | --- | --- |
| architecture AE  compilation | INVERSE 1 |  |  |
| Adamgrad | Corentin |  |  |
|  |  |  |  |
|  |  |  |  |

## Conclusion

### Difficultés

Nous avons rencontré plusieurs difficultés, que nous pouvons classer dans différentes catégories :

Humaines

La première des difficultés est le travail d’équipe dans un temps restreint sur un domaine qu’aucun d’entre nous ne maîtrise. Il a fallu arriver à se connaître et à se comprendre avant de pouvoir se lancer dans le projet, ce qui en un mois est déjà un premier challenge en soit ! Heureusement, notre bonne humeur à tous et notre envie de travailler ensemble nous ont vite permis de surmonter cette première étape. Une fois ce cap passé nous avons pu commencer à utiliser nos expériences professionnelles très diverses pour amener des visions complémentaires aux problèmes.

Organisationnelles

Au-delà du rythme de la formation à tenir en parallèle du projet, nous avions tous des contraintes liées à la recherche d’emploi ou à un emploi à réaliser en parallèle. Ajoutez à cela les contraintes familiales de chacun et la nécessité de travailler à distance, et nous arrivons à la seconde difficulté. De ce fait, chacun avançait à un rythme différent sur les modules d’une semaine sur l’autre. Les modules étant tantôt très pertinents vis à vis du projet tantôt très éloigné, il était au début difficile d’avancer tous ensemble, nous avons donc mis en place des échanges réguliers via slack et zoom afin d’avoir tous le même niveau d’information sur le projet.

Nous avons également dû apprendre à utiliser des outils de code très différents (Jupyter, Collab, GitHub,VSC) tout en essayant de comprendre comment aller articuler notre projet et donc quels aller être nos futurs besoins. Nous avons réalisé un certain nombre d’aller retours sur des environnements de travail locaux et cloud selon les phases et les difficultés du projet. Cela a mis à rude épreuve l’organisation de chacun et du projet, devant à chaque fois naviguer entre les différents environnements et les données qui y étaient stockées.

Nous avons donc pu apprendre à utiliser plusieurs compilateurs différents, des systèmes locaux et sur le cloud, le tout avec et sans outils de calcul de différence.

Techniques

Au début nos algorithmes étaient simples, et notre connaissance limitée, techniquement nos limites étaient donc le temps de chargement du jeux de données (9Go) qui au delà de la place prise sur le disque, demandait un temps de traitement de 30-50min avant de pouvoir être utilisé. Nous avons réussi avec le temps à contourner le problème de temps de traitement en mettant en place des fichiers npy prétraités, mais pas la problématique de la dimension du jeux de données.

Très vite la taille de ce dernier nous a posé un nouveau problème: l'impossibilité de le stocker dans son intégralité sur le cloud. Nos machines locales étant limitées en puissance de calcul, nous avons donc une double limitation qui nous a obligées à traiter les données machine par machine au lieu de traiter tout la donnée ensemble. De même cette capacité de calcul limitée nous a obligé à bâtir des modèles au plus juste de notre besoin, l’ajout de dimension nous renvoyant immédiatement une erreur de puissance disponible.

Cela nous a permis de découvrir deux axes importants de la DS, des systèmes afin de stocker des données pré-traitées et les charger ainsi que la recherche d’hyper paramètres plutôt que le renforcement de la complexité du réseau.

Expérience

Enfin l’objectif premier du projet étant d'acquérir de l’expérience sur les processus de machine learning et sur les environnements associés il était naturel qu’au début aucun d’entre nous n’avait d’expérience solide dans ce domaine. Toutefois, sans ‘sachant’ dans l’équipe pour nous guider, le choix de la direction n’était pas évident et les aller retours se faisaient au fil des découvertes et des résolutions de problèmes.

Grâce aux erreurs réalisées au fil du projet nous savons un peu plus aujourd’hui comment nous aurions guider le démarrage, le temps et, les ressources que nous aurions accorder à chaque étape.