Étude et implémentation d'un algorithme de deep learning pour la détection d'anomalies dans des logs applicatifs

Emna Merdessi, Sarra Hamdi, Mouadh Kouki IOT3B

Introduction

Dans le cadre du Projet de Fin d'Année (PFA), je développe **SurveillAPI**, une solution intelligente de surveillance des API. Après discussion avec mes binômes, nous avons convenu de travailler sur l'une des **perspectives principales** de ce projet : la **détection d'anomalies par le machine learning**, l'**export des résultats en CSV** et une **future surveillance en temps réel**. Le dataset utilisé est un **export réel issu d'une base MongoDB**, contenant des logs applicatifs authentiques.

1. Choix de l'algorithme

Nous retenons l'**Autoencoder**, modèle de **deep learning non supervisé** non vu en classe, adapté aux données non étiquetées et de grande dimension.

2. Principe de l'Autoencoder

L'Autoencoder comprend un **encodeur** qui compresse l'entrée vers un espace latent réduit (bottleneck) et un **décodeur** qui reconstruit l'origine. Une **anomalie** est signalée si l'erreur de reconstruction dépasse un seuil (voir figure **??**).

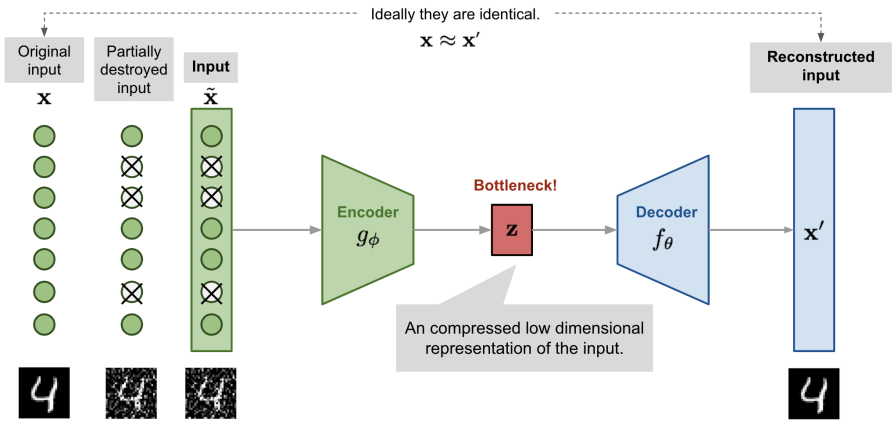

FIGURE 1 – Architecture générale d'un Autoencoder

3. Dataset et nature du problème

Export MongoDB de **577 logs** ; après encodage One-Hot : **605 variables**.
Nature : apprentissage **non supervisé**, détection d'anomalies.

4. Implémentation

4.1. Pré-traitement

Suppression des identifiants uniques, encodage One-Hot, imputation, normalisation Min-Max.

4.2. Variables

Observées : 605 variables pré-traitées ; **expliquée** : reconstruction du log lui-même.

4.3. Partition

80 % apprentissage, 20 % test.

4.4. Architecture

Sequential : Dense(16)→ReLU → Dense(8)→ReLU → Dense(4) *(bottleneck)* → Dense(8)→ReLU → Dense(16)→ReLU → Dense(605).
Paramètres : 20 337.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	9,696
dense_1 (Dense)	(None, 8)	136
dense_2 (Dense)	(None, 4)	36
dense_3 (Dense)	(None, 8)	40
dense_4 (Dense)	(None, 16)	144
dense_5 (Dense)	(None, 605)	10,285

Total params: 20,337 (79.44 KB)
Trainable params: 20,337 (79.44 KB)
Non-trainable params: 0 (0.00 B)

4.5. Entraînement

50 epochs, loss = MAE ; reconstruction utilisée pour calculer l'erreur.

```
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 42/50
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 43/50
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 44/50
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 45/50
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 46/50
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 47/50
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 48/50
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 49/50
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 0.0043 - val_loss: 0.0042
Epoch 50/50
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 0.0043 - val_loss: 0.0042
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step
```

5. Évaluation

Métrique : **MAE**. Seuil : $\mu + 2{,}5\,\sigma = 0{,}0113$.
Courbes d'entraînement et de validation convergent rapidement (figure 2), traduisant une bonne généralisation sans sur-apprentissage.

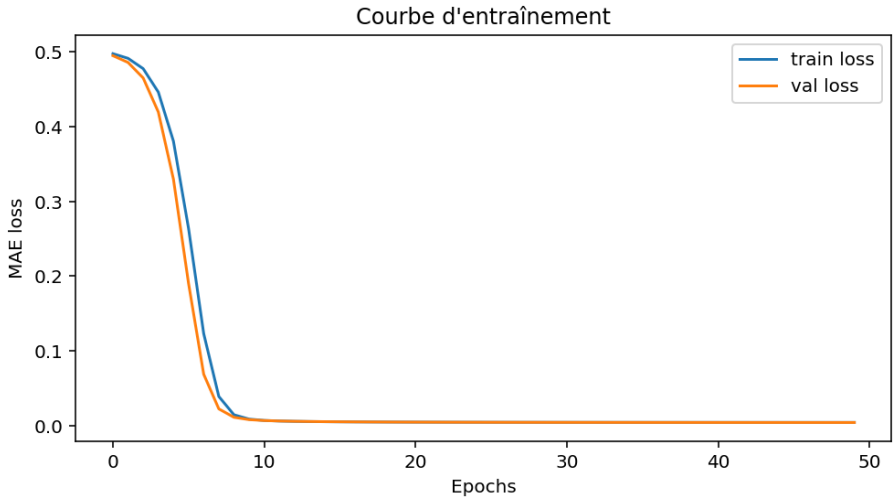

FIGURE 2 – Courbe d'entraînement et de validation (MAE)

Résultat : **1 anomalie détectée** dans le jeu de test.

```
Seuil d'anomalie (μ + 2.5σ) : 0.011313328179179134
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
Nombre d'anomalies détectées : 1
Fichier 'anomalies_detectees.csv' sauvegardé avec 1 anomalies.
```

6. Interprétation

L'anomalie (indice 98, loss = 0,0178) présente une reconstruction très éloignée de l'originale (figure 3), confirmant le comportement anormal.

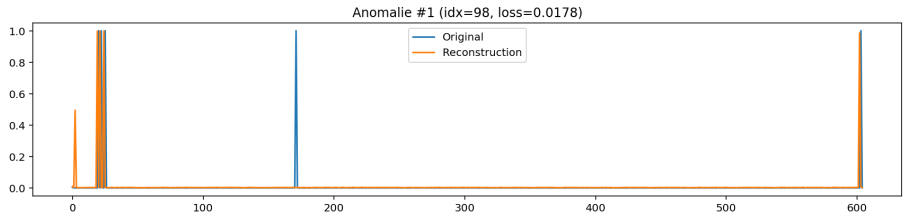

FIGURE 3 – Comparaison originale / reconstruction pour l'anomalie détectée

7. Choix des paramètres

— **Bottleneck = 4** : compacité suffisante, évite le sur-apprentissage.
— **Seuil = 2,5** σ : compromis sensibilité / faux positifs.

Conclusion

L'Autoencoder permet une détection fiable d'anomalies dans des logs non étiquetés. Les exports CSV sont déjà fonctionnels ; une surveillance en temps réel est prévue dans la suite du projet SurveillAPI.