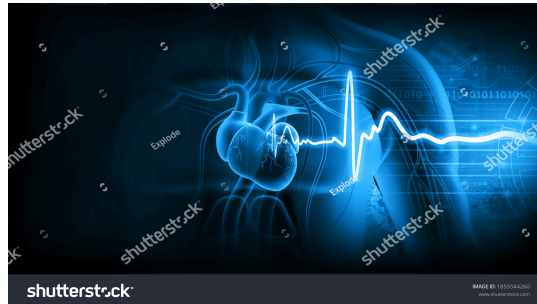


Projet De Deep Learning



ECG Anomaly Detection

Réalisé par: -Meski Melissa

Encadré par: Mme N.Dif

-Lebga Hanane

-Atallah Chaker Abdel Djalil

-Boudali Riyad

Table des matières

Définition du signal ECG	3
Objectif du projet	3
Dataset.....	3
Prétraitement.....	4
Normalisation.....	4
Filtrage.....	5
Equilibrage du dataset.....	6
Extraction des caractéristiques.....	6
construction du modèle.....	7
Les modèles LSTM.....	7
Leur fonctionnement.....	8
Les architectures utilisées.....	8
LSTM Standard	8
LSTM bidirectional.....	10
Stacked LSTM.....	11
GRU.....	13
CNN-LSTM.....	15
Comparaison des architectures des LSTM.....	17
Comparaison des résultats des LSTM.....	17

Auto-encodeur.....	17
Architectures de l'auto-encodeur.....	20
Comparaison des architectures de l'auto-encodeur.....	20
Comparaison des résultats l'auto-encodeur.....	25
Déploiement.....	25
Conclusion.....	27
Références.....	28

Définition de ECG :

Un électrocardiogramme (ECG) est l'un des examens les plus simples et les plus rapides utilisés pour évaluer la santé cardiaque. Des électrodes (petites pièces métalliques posées sur des adhésifs collés directement sur la peau) sont placées à certains endroits sur la poitrine, les bras et éventuellement les jambes. Les électrodes sont connectées à une machine ECG par des fils conducteurs. Cet appareil permet d'enregistrer l'activité électrique du cœur, afin de l'interpréter et de l'imprimer ou de la transmettre sous forme numérique au médecin prescripteur.

Objectif du projet:

Le but est de développer un modèle capable de détecter les anomalies dans les ECG. Les anomalies pourraient indiquer des problèmes cardiaques ou des conditions médicales sous-jacentes.

On va explorer comment les ordinateurs peuvent aider à comprendre les battements de cœur. On va utiliser des données spéciales appelées électrocardiogrammes (ou ECG) pour cela. Ces données montrent comment le cœur fonctionne.

Dataset:

Dans ce projet on a utilisé la base de données ECG PTB Diagnostic (<https://www.kaggle.com/datasets/shayanfazeli/heartbeat/data>) qui est une collection de 14 552 enregistrements ECG provenant de la base de données de diagnostic PTB de Physionet. Ces signaux ECG sont classés en deux catégories : les battements cardiaques normaux et ceux affectés par des anomalies cardiaques. L'ensemble de données est échantillonné à 125 Hz, fournissant des données à haute résolution pour une analyse approfondie.

Plongeons dans quelques détails essentiels sur la base de données ECG PTB Diagnostic :

- Nombre d'échantillons : 14 552

- Nombre de catégories : 2
- Fréquence d'échantillonnage : 125 Hz
- Source des données : Base de données de diagnostic PTB de Physionet

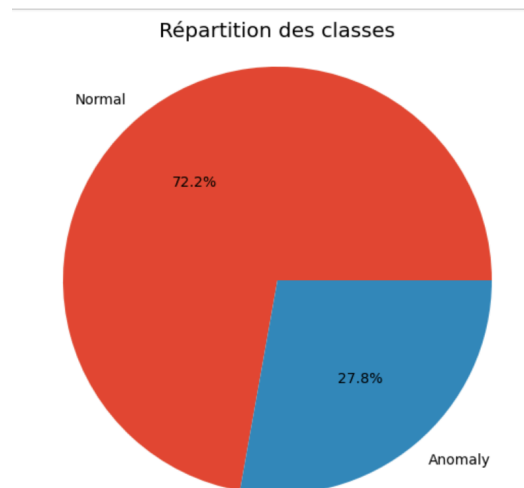


Figure 1 : Diagramme circulaire de l' ensemble de données déséquilibré

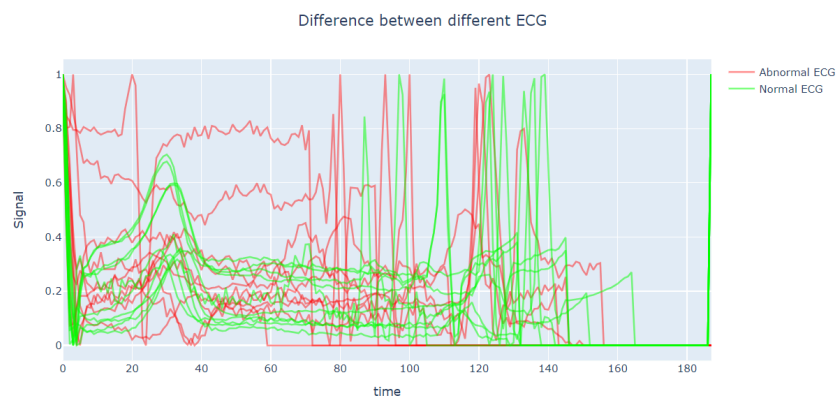


Figure 2 : Diagram montrant la différence entre différent ECG

Prétraitement:

Normalisation :

Les données ont été normalisées entre 1 et -1 à l' aide du la fonction `Minmaxscaler()` ,cela se fait en soustrayant la valeur minimale de chaque

caractéristique puis en divisant par la différence entre la valeur maximale et la valeur minimale.

Filtrage:

Comme nos données comportent des bruits, nous avons ajusté des techniques de filtrage.

Trois techniques de filtrage ont été appliquées :

- **Filtrage médian (median filtering):** Le filtrage médian est une méthode de filtrage non linéaire qui remplace chaque échantillon du signal par la médiane de ses voisins.

Utilité : Il est efficace pour éliminer les impulsions brusques et les pics isolés, ce qui peut être causé par des interférences ou des artefacts dans le signal ECG.

- **Filtrage passe-bas (low-pass filtering):** Un filtre passe-bas Butterworth est un type de filtre qui atténue les fréquences élevées tout en laissant passer les fréquences plus basses.

Utilité : Il est couramment utilisé pour atténuer le bruit à haute fréquence dans les signaux, ce qui peut provenir de diverses sources telles que l'activité musculaire, les mouvements du patient ou les interférences électriques.

- **Filtrage par ondelettes (wavelet filtering):** la filtration par ondelettes (ou *wavelet filtering*) est une approche distincte de celle des réseaux de neurones convolutifs (CNN), mais elle peut être utilisée en complément pour le traitement des signaux, y compris les signaux ECG.

Le filtrage par ondelettes est une technique qui utilise la décomposition en ondelettes pour séparer le signal en différentes échelles de fréquence.

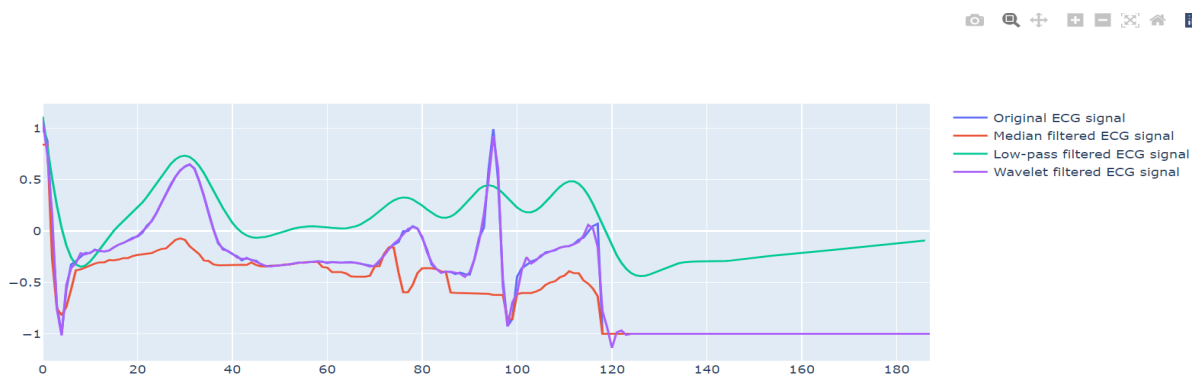
Utilité : Il permet de supprimer le bruit à différentes échelles de fréquence tout en préservant les caractéristiques importantes du signal. Cela peut être particulièrement efficace pour éliminer le bruit non stationnaire et les artefacts présents dans les signaux biologiques tels que l'ECG.

Après avoir appliqué chaque technique, la taille du signal peut être réduite en raison de l'élimination du bruit. Par conséquent, nous complétons les valeurs manquantes du signal avec des zéros afin que tous les signaux aient la même échelle.

Après nous avons calculé le mse de chaque méthodes les résultats sont trouvé comme celui ici :

	Median filtering	Low-pass filtering	Wavelet filtering
MSE	0.0965	0.2967	0.00169

Donc, la meilleure technique c'est Wavelet filtering avec le MSE le plus diminué.



Equilibrage du dataset:

Étant donné le déséquilibre de notre jeu de données, nous appliquons la technique de suréchantillonnage SMOTE (Synthetic Minority Over-sampling Technique) pour augmenter la représentation de la classe minoritaire en générant des échantillons synthétiques.

Extractions des caractéristiques:

Quelles sont les fonctionnalités ?

Les fonctionnalités sont les attributs qui décrivent des données particulières et peuvent être utilisées pour faire des prédictions ou classer des cas inhabituels. Il est important d'extraire les fonctionnalités pertinentes, car fournir au modèle des

informations non pertinentes ou redondantes pourrait avoir un impact négatif sur ses performances et sa précision.

Pourquoi extraire des fonctionnalités ?

Après avoir prétraité les données, nous devons extraire les caractéristiques des signaux ECG pour notre modèle d'apprentissage en profondeur. Entraîner directement le modèle sur toutes les données de séries chronologiques pourrait entraîner un surajustement et être coûteux en termes de calcul. Nous pouvons réduire la quantité de données en extrayant des caractéristiques pertinentes et en améliorant la capacité de généralisation du modèle à de nouvelles données invisibles.

Choisir des fonctionnalités pertinentes

Dans ce contexte, les fonctionnalités suivantes sont utilisées :

Amplitude T : hauteur de l'onde T sur le graphique de l'électrocardiogramme (ECG), qui symbolise le retour au repos du cœur après une contraction.

Amplitude R : La hauteur de l'onde R sur le graphique ECG représente la contraction initiale du muscle cardiaque.

Intervalle RR : le temps entre deux ondes R consécutives sur le graphique ECG, représentant le temps entre les battements cardiaques.

Durée QRS : le temps nécessaire au complexe QRS, représentant le signal électrique traversant les ventricules.

Processus d'extraction de fonctionnalités :

Pour extraire ces caractéristiques, nous calculons d'abord les pics R&T, l'amplitude R, l'intervalle RR, etc., à l'aide de la bibliothèque `scipy.signals`. Ensuite, nous calculons la moyenne, la médiane, la somme et d'autres mesures statistiques de chaque caractéristique pour capturer ses caractéristiques. Toutes ces fonctionnalités sont ensuite stockées dans un tableau.

Construction du modèle :

Étant en possession d'une collection de dessins illustrant les battements de cœur de divers individus, notre objectif est d'exploiter les réseaux de

neurones pour apprendre à interpréter ces tracés et à détecter toute anomalie cardiaque. Nous commencerons par utiliser des réseaux de neurones récurrents (RNN) de type LSTM (Long Short-Term Memory).

Les modèles LSTM:

Les réseaux LSTM sont une variation des réseaux de neurones récurrents (RNN). Ils ont été proposés par Jürgen Schmidhuber et son équipe au début des années 1990. Alors que les RNN classiques ont des difficultés à traiter les dépendances à long terme en raison du problème du "vanishing gradient", les LSTM ont été spécialement conçus pour surmonter ce problème.

Leur fonctionnement:

Les LSTM fonctionnent de manière similaire aux réseaux de neurones récurrents classiques, mais avec une architecture spéciale conçue pour capturer les dépendances à long terme dans les séquences de données.

L'élément clé d'un LSTM est la cellule mémoire qui agit comme une bande transporteuse, transférant l'état de la cellule d'un pas de temps au suivant. Cette cellule mémoire est régulée par trois portes (gates) qui contrôlent le flux d'informations :

1. **Porte d'oubli (Forget Gate)** : Détermine quelles informations de l'état précédent doivent être conservées ou oubliées.
2. **Porte d'entrée (Input Gate)** : Décide quelles nouvelles informations de l'entrée courante et de l'état précédent doivent être stockées dans la cellule mémoire.
3. **Porte de sortie (Output Gate)** : Détermine quelles parties de la cellule mémoire doivent être utilisées pour générer la sortie courante et l'état caché.

Les Architectures utilisées:

Architecture LSTM standard:

Cette architecture de base utilise une couche LSTM avec 64 unités, suivie d'une couche entièrement connectée (Dense) avec une fonction d'activation sigmoid pour produire une sortie binaire. Le modèle a été compilé avec différentes fonctions d'optimisation telles que Adam, SGD (Stochastic Gradient Descent) et RMSprop. La fonction de perte utilisée était l'entropie croisée binaire.

Architecture du modèle :

Model: "sequential_11"

Layer (type)	Output Shape	Param #
lstm_9 (LSTM)	(None, 64)	16896
dense_15 (Dense)	(None, 1)	65
Total params: 16961 (66.25 KB)		
Trainable params: 16961 (66.25 KB)		
Non-trainable params: 0 (0.00 Byte)		

Résultats de LSTM standard:

Pour balanced dataset:

Accuracy: 81.27147766323824 %					Predicted	
AUC: 0.71					Predicted Negative Predicted Positive	
Classification Report:					Actual	<div> <div></div> <div></div> </div>
	precision	recall	f1-score	support		
0	0.82	0.95	0.88	2076		
1	0.79	0.47	0.59	834		
accuracy			0.81	2910		
macro avg	0.80	0.71	0.73	2910	Actual Positive	<div> <div></div> <div></div> </div>
weighted avg	0.81	0.81	0.80	2910		

Pour Imbalanced dataset :

Accuracy: 82.19931271477662 %					Predicted	
AUC: 0.77					Predicted Negative Predicted Positive	
Classification Report:					Actual	<div> <div></div> <div></div> </div>
	precision	recall	f1-score	support		
0	0.86	0.90	0.88	2076		
1	0.71	0.63	0.67	834		
accuracy			0.82	2910		
macro avg	0.79	0.77	0.77	2910	Actual Positive	<div> <div></div> <div></div> </div>
weighted avg	0.82	0.82	0.82	2910		



Architecture Bidirectional LSTM

Cette architecture utilise une couche bidirectionnelle LSTM, qui combine les informations d'un LSTM standard et d'un LSTM inversé pour capturer les dépendances à long terme dans les deux directions. Elle a été testée avec les optimisateurs Adam et SGD.

Architecture du modèle:

Model: "sequential_10"

Layer (type)	Output Shape	Param #
bidirectional_1 (Bidirectional)	(None, 128)	33792
dense_14 (Dense)	(None, 1)	129
Total params: 33921 (132.50 KB)		
Trainable params: 33921 (132.50 KB)		
Non-trainable params: 0 (0.00 Byte)		

Résultats de bidirectional lstm:

Pour balanced dataset :

Accuracy: 71.34020618556701 %

AUC: 0.5

Classification Report:					
	precision	recall	f1-score	support	
0	0.71	1.00	0.83	2076	
1	0.00	0.00	0.00	834	
accuracy			0.71	2910	
macro avg	0.36	0.50	0.42	2910	
weighted avg	0.51	0.71	0.59	2910	

		Predicted	
		Predicted Negative	Predicted Positive
Actual	Actual Negative	2076	0
	Actual Positive	834	0

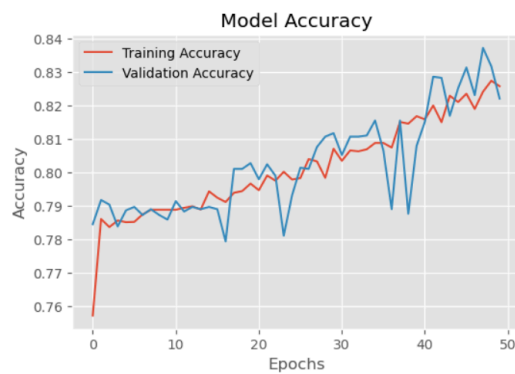
Pour Imbalanced dataset:

Accuracy: 82.19931271477662 %

AUC: 0.77

Classification Report:					
	precision	recall	f1-score	support	
0	0.86	0.90	0.88	2076	
1	0.71	0.63	0.67	834	
accuracy			0.82	2910	
macro avg	0.79	0.77	0.77	2910	
weighted avg	0.82	0.82	0.82	2910	

		Predicted	
		Predicted Negative	Predicted Positive
Actual	Actual Negative	1864	212
	Actual Positive	306	528



Modèle LSTM Empilé (Stacked LSTM)

Cette architecture empile deux couches LSTM, où la première couche LSTM a un paramètre pour renvoyer toutes les sorties de séquence. La deuxième couche LSTM traite ensuite ces sorties de séquence. Une couche Dense avec une fonction d'activation sigmoid est ajoutée à la fin.

Architecture du modèle :

Model: "sequential_5"

Layer (type)	Output Shape	Param #
lstm_5 (LSTM)	(None, 19, 64)	16896
lstm_6 (LSTM)	(None, 64)	33024
dense_5 (Dense)	(None, 1)	65

Total params: 49985 (195.25 KB)
Trainable params: 49985 (195.25 KB)
Non-trainable params: 0 (0.00 Byte)

Résultats de stacked LSTM:

Pour Balanced dataset

Accuracy: 83.16151202749141 %

AUC: 0.76

Classification Report:					
	precision	recall	f1-score	support	
0	0.85	0.93	0.89	2076	
1	0.78	0.58	0.66	834	
accuracy			0.83	2910	
macro avg	0.81	0.76	0.77	2910	
weighted avg	0.83	0.83	0.82	2910	

		Predicted	
		Predicted Negative	Predicted Positive
Actual	Actual Negative	1940	136
	Actual Positive	354	480

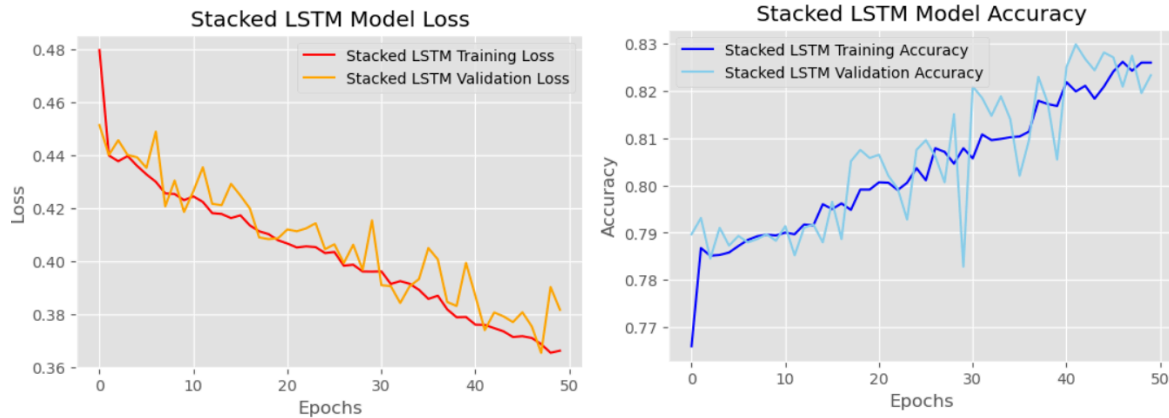
Pour Imbalanced dataset :

Accuracy: 83.05841924398626 %

AUC: 0.76

Classification Report:					
	precision	recall	f1-score	support	
0	0.85	0.93	0.89	2076	
1	0.76	0.59	0.67	834	
accuracy			0.83	2910	
macro avg	0.81	0.76	0.78	2910	
weighted avg	0.83	0.83	0.82	2910	

		Predicted	
		Predicted Negative	Predicted Positive
Actual	Actual Negative	1922	154
	Actual Positive	339	495



Modèle GRU (Gated Recurrent Unit):

Cette architecture utilise une couche GRU (Gated Recurrent Unit) avec 64 unités, suivie d'une couche Dense avec une fonction d'activation sigmoid.

Architecture du modèle :

Model: "sequential_5"

Layer (type)	Output Shape	Param #
lstm_5 (LSTM)	(None, 19, 64)	16896
lstm_6 (LSTM)	(None, 64)	33024
dense_5 (Dense)	(None, 1)	65
Total params: 49985 (195.25 KB)		
Trainable params: 49985 (195.25 KB)		
Non-trainable params: 0 (0.00 Byte)		

Résultats du modèle GRU:

Architecture du modèle :

Model: "sequential_6"

Layer (type)	Output Shape	Param #
gru_1 (GRU)	(None, 64)	12864
dense_6 (Dense)	(None, 1)	65

=====

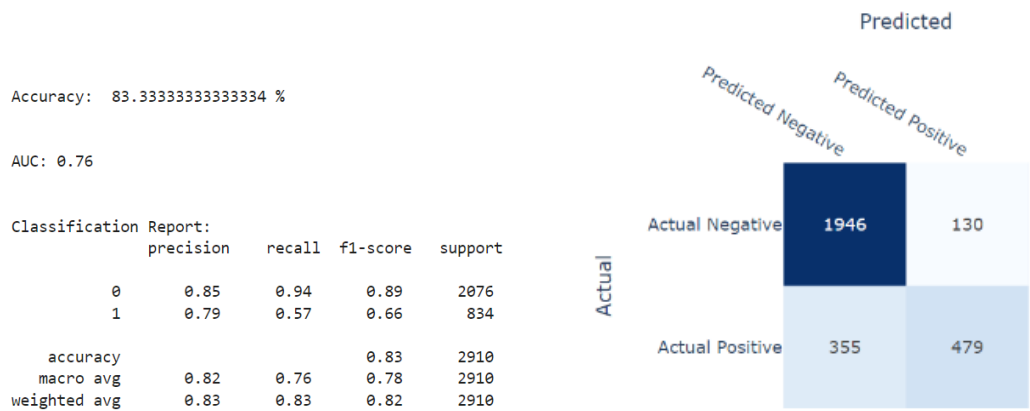
Total params: 12929 (50.50 KB)

Trainable params: 12929 (50.50 KB)

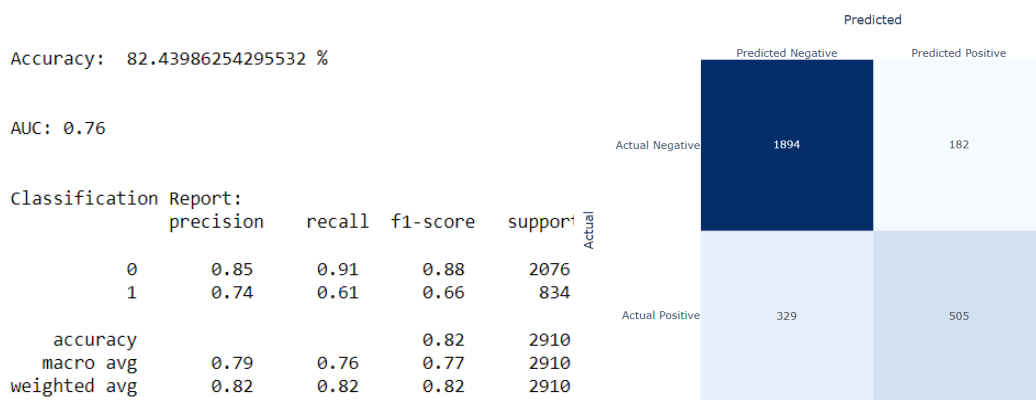
Non-trainable params: 0 (0.00 Byte)

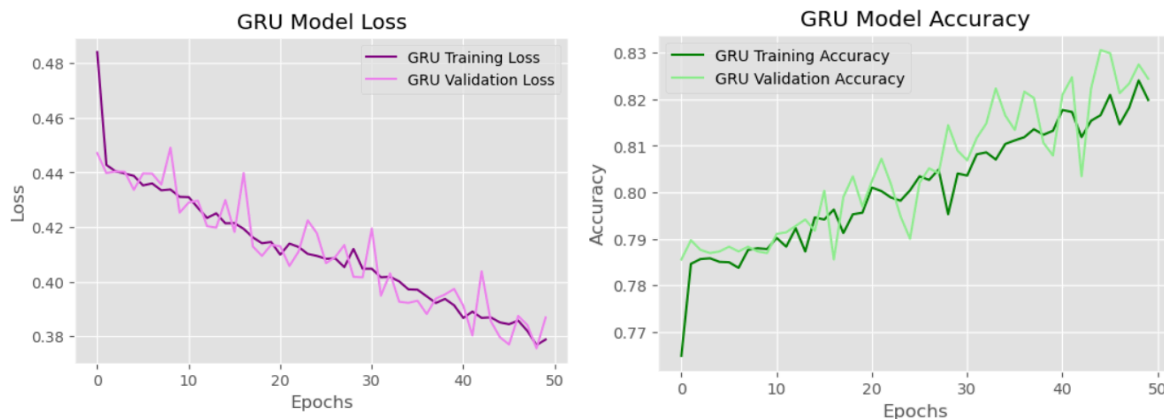
=====

Pour balanced dataset:



Pour Imbalanced dataset:





Modèle CNN-LSTM

Cette architecture combine une couche convolutionnelle 1D (Conv1D) avec une couche de pooling (MaxPooling1D) pour extraire les caractéristiques spatiales du signal ECG. Ensuite, une couche LSTM traite les caractéristiques extraites, suivie d'une couche Dense avec une fonction d'activation sigmoid.

Architecture du modèle :

Model: "sequential_7"

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 17, 64)	256
max_pooling1d_1 (MaxPooling1D)	(None, 8, 64)	0
lstm_7 (LSTM)	(None, 64)	33024
dense_7 (Dense)	(None, 1)	65
Total params: 33345 (130.25 KB)		
Trainable params: 33345 (130.25 KB)		
Non-trainable params: 0 (0.00 Byte)		

Résultats du modèle CNNLSTM:

Pour Balanced dataset

Accuracy: 83.3333333333334 %

AUC: 0.76

Classification Report:	Report:			
	precision	recall	f1-score	support
0	0.85	0.94	0.89	2076
1	0.79	0.57	0.66	834
accuracy			0.83	2910
macro avg	0.82	0.76	0.78	2910
weighted avg	0.83	0.83	0.82	2910

Actual	Predicted	
	Predicted Negative	Predicted Positive
Actual Negative	1946	130
Actual Positive	355	479

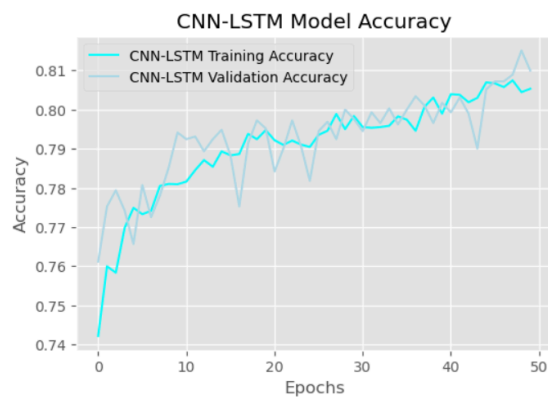
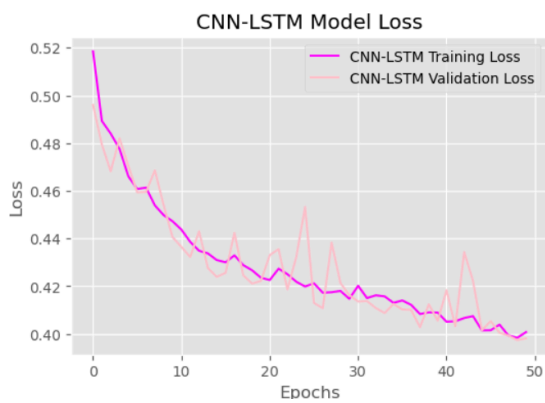
Pour Imbalanced dataset

Accuracy: 82.43986254295532 %

AUC: 0.76

Classification Report:	Report:			
	precision	recall	f1-score	support
0	0.85	0.91	0.88	2076
1	0.74	0.61	0.66	834
accuracy			0.82	2910
macro avg	0.79	0.76	0.77	2910
weighted avg	0.82	0.82	0.82	2910

Actual	Predicted	
	Predicted Negative	Predicted Positive
Actual Negative	1894	182
Actual Positive	329	505



Comparaison des architectures de LSTM:

Architecture	Description
LSTM standard	Couche LSTM (64 unités) + Dense (sigmoid)
Bidirectional LSTM	Couche Bidirectional LSTM
LSTM Empilé(stacked LSTM)	Deux couches LSTM empilées + Dense (sigmoid)
GRU	Couche GRU (64 unités) + Dense (sigmoid)
CNN-LSTM	Conv1D + MaxPooling1D + LSTM + Dense (sigmoid)

Comparaison des résultats des LSTM:

Modèle	accuracy	recall	F1-score
LSTM standard	0.82	0.76	0.77
Bidirectional LSTM	0.82	0.76	0.77
Stacked LSTM	0.83	0.76	0.78
GRU	0.83	0.75	0.77
CNN-LSTM	0.83	0.75	0.77

Auto-encodeur:

L'auto-encodeur est un type de réseau de neurones non supervisé qui vise à reconstruire les données d'entrée en les compressant dans un espace de représentation de plus petite dimension, puis en les décodant vers leur forme d'origine.

Dans notre projet sur l'auto-encodeur, nous avons deux composantes distinctes : une première partie qui prend les dessins des battements de cœur et les simplifie, tandis qu'une autre partie tente de reconstruire ces dessins à partir de la forme simplifiée.

Étant donné que notre objectif principal est de repérer les signaux cardiaques anormaux ou les éventuels problèmes cardiaques dans les électrocardiogrammes (ECG), le réseau de neurones autoencodeur nous assiste dans ce processus de diverses manières :

1. **Compression des données** : La première partie de l'autoencodeur (l'encodeur) réduit la taille des données en les transformant en une forme plus simple. Cela permet de réduire la complexité des données tout en conservant les informations importantes sur les battements de cœur.
2. **Extraction de caractéristiques** : Pendant la transformation en une forme plus simple, le réseau de neurones apprend à reconnaître les schémas importants dans les battements de cœur. Ces schémas peuvent inclure des variations de fréquence cardiaque, des intervalles entre les battements, etc.
3. **Reconstruction des données** : La deuxième partie de l'autoencodeur (le décodeur) essaie de reconstruire les battements de cœur à partir de la forme simple. Si la reconstruction est proche de l'original, cela signifie que le réseau a bien compris les données. Sinon, cela peut indiquer qu'il y a quelque chose d'anormal dans les battements de cœur.

En combinant ces étapes, le réseau de neurones autoencodeur peut détecter les signaux cardiaques anormaux en reconnaissant les schémas inhabituels dans les électrocardiogrammes. Cela peut être extrêmement utile pour les médecins et les professionnels de la santé, car cela leur permet de repérer rapidement les problèmes cardiaques potentiels chez les patients, ce qui peut conduire à un diagnostic et un traitement précoces.

Architecture de l'auto-encodeur:

L'encodeur est défini comme une séquence de trois couches entièrement connectées (Dense) avec des activations ReLU. Les dimensions des couches sont respectivement de 40, 20 et 10 neurones.

Le décodeur est également une séquence de trois couches entièrement connectées (Dense) avec des activations ReLU. Les dimensions des couches sont inversées par rapport à l'encodeur, avec 20, 40 et 188 neurones respectivement. La dernière couche utilise une activation sigmoïde pour produire des valeurs entre 0 et 1, ce qui est courant pour les tâches de reconstruction.

De plus, nous avons mis en œuvre deux améliorations pour renforcer les performances de l'autoencodeur : nous avons incorporé des techniques de régularisation et de Dropout dans l'encodeur afin de contrer le surapprentissage et d'améliorer la capacité de généralisation du modèle. Les ajustements réalisés sont les suivants :

La première amélioration:

- **Régularisation L2** : Une régularisation L2 (norme L2) a été appliquée aux poids des couches de l'encodeur à l'aide du paramètre λ . Cette régularisation pénalise les poids élevés, ce qui aide à éviter le sur-apprentissage.
- **Dropout** : Des couches de Dropout ont été ajoutées après chaque couche Dense de l'encodeur avec un taux de 0.2. Le Dropout désactive aléatoirement une fraction des nœuds pendant l'entraînement, ce qui force le réseau à apprendre des représentations redondantes et robustes, réduisant ainsi le sur-apprentissage.

La deuxième amélioration: Augmentation de la capacité de l'auto-encodeur:

Nous avons amélioré les performances de l'auto-encodeur en augmentant le nombre de nœuds et de couches à la fois dans l'encodeur et le décodeur, en plus d'équilibrer le jeu de données. Les modifications apportées sont les suivantes :

- **Encodeur** : Le nombre de nœuds dans les couches de l'encodeur a été augmenté à 128, 64 et 32 respectivement. Plus de nœuds permettent à l'encodeur de capturer des représentations plus complexes des données d'entrée.
- **Décodeur** : Le nombre de nœuds dans les couches du décodeur a été augmenté à 64 et 128 respectivement, correspondant à la structure symétrique de l'encodeur. Cela permet au décodeur de reconstruire plus fidèlement les données d'origine à partir de la représentation compressée.
- **Dropout dans le décodeur** : Des couches de Dropout avec un taux de 0.3 ont été ajoutées au décodeur, similaires à celles de l'encodeur, pour régulariser le processus de reconstruction.

Comparaison des architecture de l'auto-encodeur:

Architecture	Description	Régularisation	Learning rate	nombre d'épochs	Batch size
auto-enodeur 1	Encodeur: Dense(40) - Dense(20) - Dense(10) Décodeur: Dense(20) - Dense(40) - Dense(188, sigmoid)	L2 (0.001)	0.001	80	512
Auto-encodeur 2	Encodeur: Dense(64) - Dropout(0.2) - Dense(32) - Dropout(0.2) - Dense(16) Décodeur: Dense(32) - Dropout(0.2) - Dense(64) - Dropout(0.2) - Dense(188, sigmoid)	L2 (0.001)	0.001	80	512
Auto-encodeur 3	Encodeur: Dense(128) - Dropout(0.3) - Dense(64) - Dropout(0.3) - Dense(32) Décodeur: Dense(64) - Dropout(0.3) - Dense(128) - Dropout(0.3) - Dense(188, sigmoid)	L2 (0.001)	0.001	80	512

1/Résultats de la première architecture de l'auto-encoder :

```

Confusion Matrix:
[[1959  113]
 [ 138  700]]
AUC Score: 0.8903927580836889
Accuracy: 0.9137457044673539
Precision: 0.8610086100861009
Recall: 0.8353221957040573
F1 Score: 0.847970926110842

```

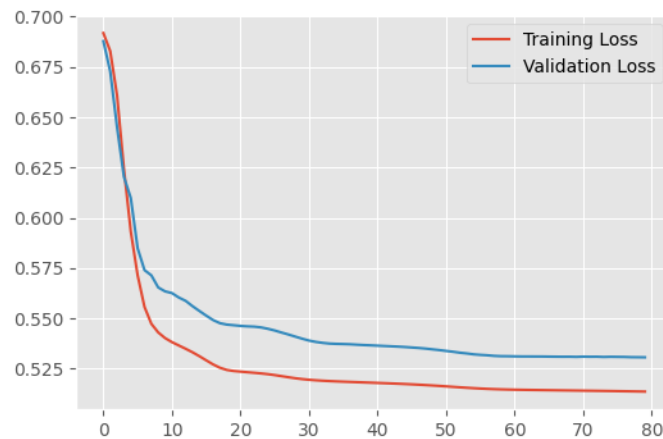


Figure: Auto-encoder results

Après avoir entraîné notre modèle, nous avons choisi un seuil en identifiant les anomalies par examiner l'erreur de reconstruction sur l'ensemble d'apprentissage. La moyenne et la norme l'écart de la perte MAE sur l'ensemble d'entraînement a été utilisé pour calculer le seuil pour la détection des anomalies.

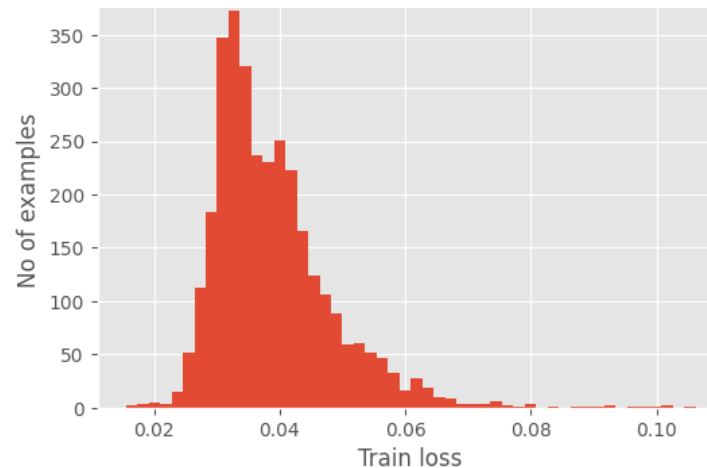


Figure :reconstruction error on the training set

En utilisant le seuil, nous pouvons transformer le problème en une simple classification binaire tâche : Si la perte de reconstruction par exemple est inférieure au seuil, nous la laisserons comme un battement de cœur normal . Alternativement, si la perte est supérieure au seuil, nous la classons comme une anomalie.

-pour évaluer les performances de notre modèle avec des battements cardiaques normaux.

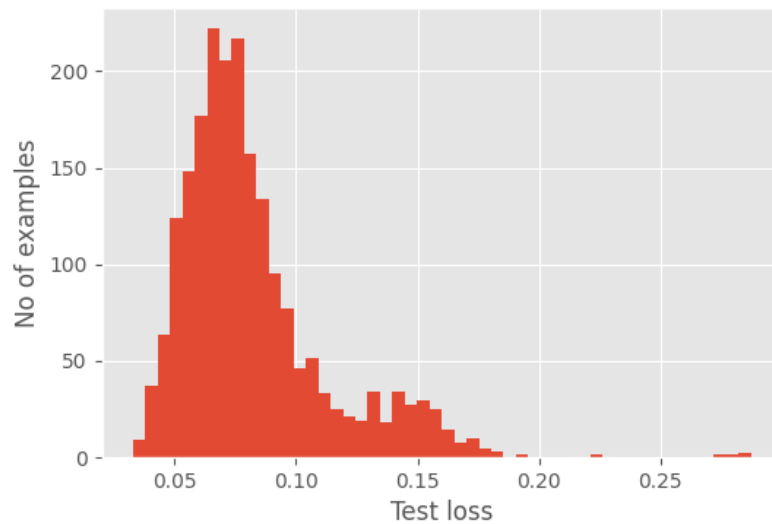


Figure: erreur de reconstruction sur un ensemble de tests anormales

2/Résultats de la deuxième architecture :

```
Confusion Matrix:
[[2072   0]
 [ 122  716]]
AUC Score: 0.9272076372315036
Accuracy: 0.9580756013745705
Precision: 1.0
Recall: 0.8544152744630071
F1 Score: 0.9214929214929215
```



Figure :Auto-encoders results

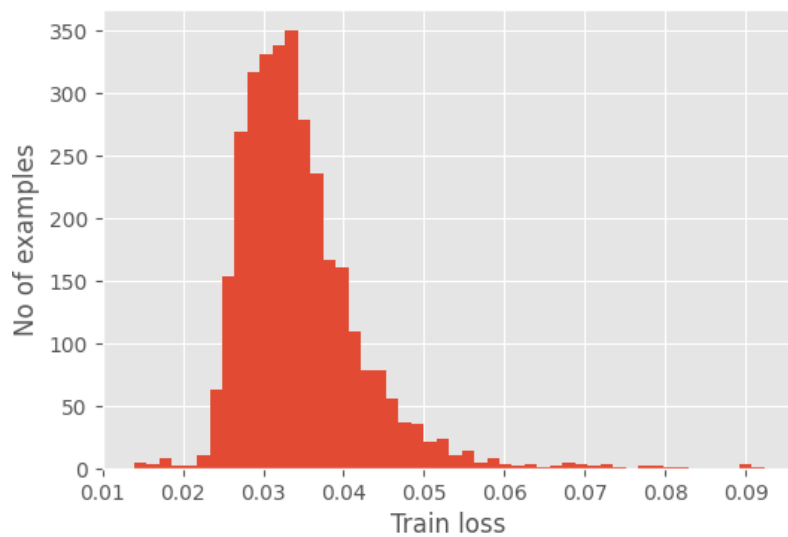


Figure :reconstruction error on the training set

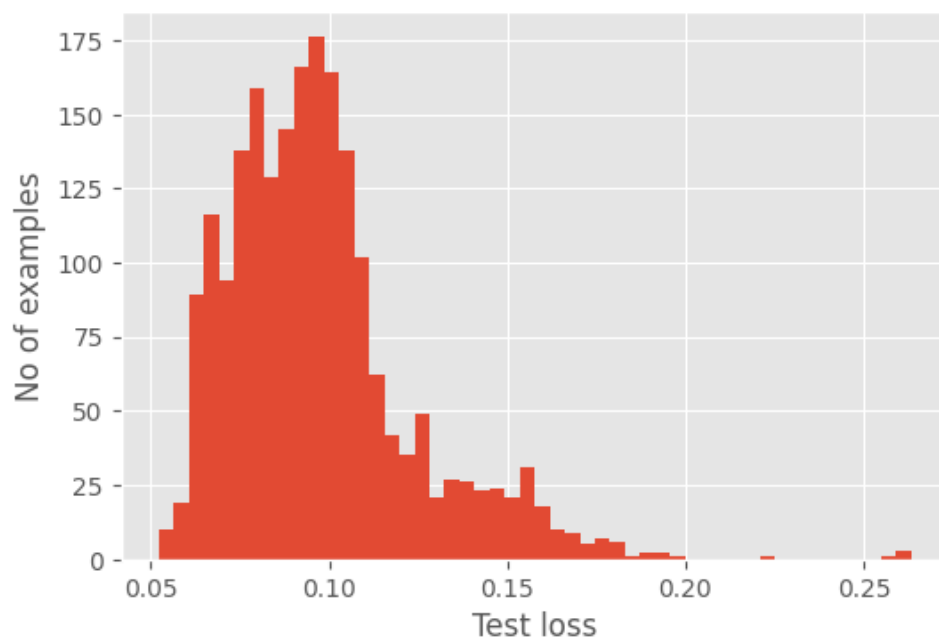


Figure: erreur de reconstruction sur un ensemble de tests anormales

3/Résultats de la troisième architecture:

```
Confusion Matrix:
[[2072   0]
 [ 136  702]]
AUC Score: 0.918854415274463
Accuracy: 0.9532646048109966
Precision: 1.0
Recall: 0.837708830548926
F1 Score: 0.9116883116883117
```

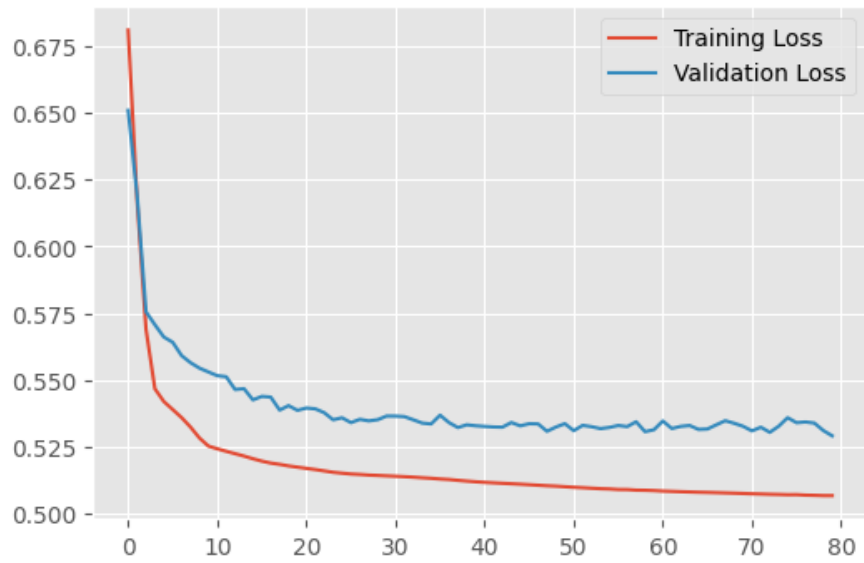



Figure :Auto-encoder results

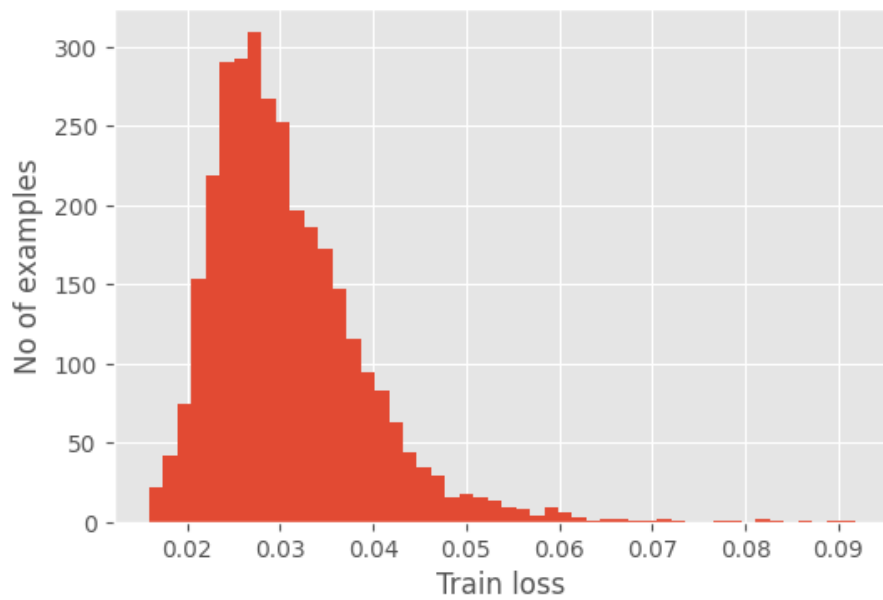


Figure :reconstruction error on the training set

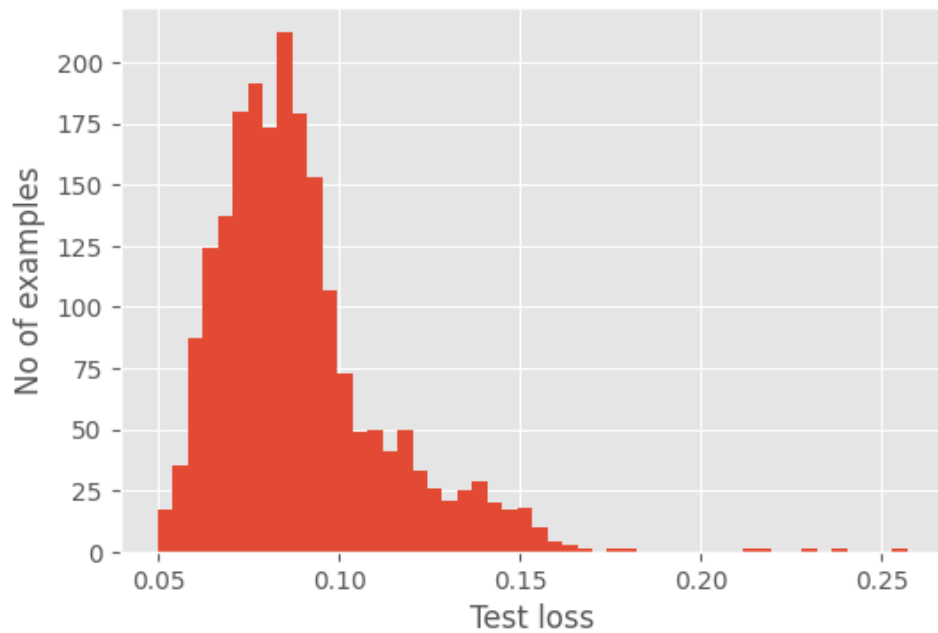


Figure: erreur de reconstruction sur un ensemble de tests anormales

Comparaison des résultats des 3 architectures d'auto-encodeur:

Modèle	accuracy	recall	F1-score
auto-encodeur 1	0.91	0.835	0.84
auto-encodeur 2	0.958	0.85	0.92
auto-encodeur 3	0.953	0.837	0.91

Déploiement:

Cette application, conçue avec Streamlit, offre la possibilité de visualiser les données des patients et de choisir un patient en particulier. Elle permet ensuite de prédire si les battements de cœur de ce patient sont normaux ou anormaux. En cas de prédiction anormale, elle affiche une image rouge avec un point d'exclamation, tandis qu'en cas de prédiction normale, elle affiche une image verte avec un signe de validation.

Détection d'anomalies ECG

Nos données:

	Name	1	2	3	4	5	6	7	8	9	10
0	Sabrina Rivas	1	0.7947	0.3754	0.1169	0	0.1719	0.2839	0.2938	0.3259	0.3451
1	Amanda Cortez	0.909	0.7915	0.4232	0.1867	0	0.0078	0.063	0.077	0.075	0.0773
2	Felicia Hayes	1	0.4789	0.0568	0.0642	0.0813	0.0727	0.0556	0.0488	0.0545	0.0416
3	Virginia Huff	1	0.8672	0.2014	0.0993	0.1413	0.1209	0.1085	0.0964	0.0934	0.1008
4	Joseph Young	0.949	0.5053	0.0042	0.0225	0.0595	0.1073	0.1104	0.1113	0.1166	0.1182

Sélectionnez un patient :

Amanda Cortez

Prédire les anomalies

Amanda Cortez

Prédire les anomalies



Conclusion :

En conclusion, ce rapport a présenté une méthode de détection d'anomalie des signaux ECG utilisant une approche de représentation transférable. Grâce à la formation de réseaux neuronaux profonds sur l'ensemble de données , nous avons démontré l'efficacité de notre méthode pour détecter avec précision les anomalies des signaux cardiaques ECG. Les résultats montrent que notre approche suggérée atteint des précisions significatives . Ces résultats mettent en évidence le potentiel des représentations transférables dans la classification des signaux ECG et offrent des pistes prometteuses pour de nouvelles avancées. recherche et développement dans ce domaine. En tirant parti de la puissance du deep learning techniques, nous pouvons améliorer la précision et l'efficacité de l'analyse ECG, menant ainsi pour de meilleurs soins aux patients, une détection précoce des anomalies cardiaques et une stratégie de traitement.

Référence:

- ELECTROCARDIOGRAM (ECG) SIGNAL PROCESSING LEIF SO" RNMO Lund University Sweden
PABLO LAGUNA Zaragoza University Spain
- A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG) [Hongzu Li*](#) and [Pierre Boulanger](#)
- AUTAN-ECG: An AUToencoder bAsed system for anomaly detectionN in ECG signals
- Anomaly detection in ECG time signals via deep long short-term memory networks