

---

# Sur la classification d'Ã©toiles en fonction de leur spectre d'absorption par apprentissage automatique

---

**Patrice B  chard**  
D  partement d'informatique  
et de recherche op  rationnelle  
Universit   de Montr  al  
Montr  al, QC H3T 1J4  
patrice.bechard@umontreal.ca

**Jean-Pascal Gu  vin**  
D  partement de math  matiques  
et de statistique  
Universit   de Montr  al  
Montr  al, QC H3T 1J4  
jean-pascal.guevin@umontreal.ca

IFT6390 - Fondements de l'apprentissage machine - 22 d  cembre 2017

## Abstract

La classification de spectres stellaires provenant du *Sloan Digital Sky Survey* a   t   effectu  e par trois algorithmes d'apprentissage, soit les machines    vecteurs de support, les r  seaux de neurones de type perceptron multicouche et les r  seaux de neurones convolutifs. Un taux de classification 93.40% a   t   obtenu avec les SVM, 94.41% avec les MLP et 94.57% avec les CNN. Une t  che similaire a   t   conduite avec des donn  es d'  lectrocardiogramme avec des r  sultats moins concluants. Les taux de classification obtenus sont de 61.36% pour les SVM, 60.69% pour les MLP et 77.39% pour les CNN.

## 1 Introduction

L'exploration de notre univers observable a amen   les astrophysiciens    observer et    cat  goriser des centaines de milliers d'Ã©toiles en fonction notamment de leur taille, de leur masse, de leur temp  rature et de leur composition. Ce processus de classification se fait, entre autre,    partir du spectre d'absorption des Ã©toiles qui est une mesure de l'intensit   du spectre   lectromagn  tique   mis par celles-ci en fonction de la longueur d'onde. Une automatisation efficace de ce processus pourrait par cons  quent   tre un grand avantage. Nous proposons donc trois algorithmes de classification d'Ã©toiles en fonction de leur type spectral    l'aide de m  thodes d'apprentissage automatique. Pour la validation des algorithmes, des donn  es d'  lectrocardiogramme,   tant aussi des donn  es corr  l  es en 1 dimension, seront utilis  es. Les algorithmes d'apprentissage utilis  s pour effectuer la classification sont les r  seaux de neurones de type MLP, les r  seaux de neurones convolutifs (CNN) ainsi que les machines    vecteur de support (SVM)    noyau souple. Les bases de donn  es ainsi que les algorithmes d'apprentissages utilis  s sont pr  sent  s en d  tails    la section ?? et les r  sultats obtenus sont pr  sent  s    la section ?. Les codes et les figures pr  sent  es pour l'ensemble du projet sont disponibles en ligne sur GitHub : [https://github.com/patricebechard/Machine\\_learning\\_IFT6390](https://github.com/patricebechard/Machine_learning_IFT6390).

## 2 M  thodes

Les donn  es utilis  es pour les spectres d'Ã©toiles proviennent de la base de donn  es *Sloan Digital Sky Survey* (SDSS) *Science Archive Server* (SAS) donnant gratuitement acc  s aux observations faites par diff  rents t  lescopes. Il est   videmment n  cessaire de traiter les spectres obtenus par le biais du SDSS, ceux-ci   tant g  n  ralement tr  s bruit  s. Un processus lissage et de normalisation utilisant notamment des moyennes mobiles permet d'en extraire

l'information pertinente en éliminant le plus possible le bruit et en ne conservant que ce qui semble correspondre à des tendances plus globales. De plus, chaque spectre a été tronqué de sorte que seul la section correspondant aux log-longueurs d'onde entre 3.65 et 3.80 – correspondant aux longueurs d'onde entre  $\approx 398.1$  nm jusqu'à  $\approx 707.9$  nm, ce qui représente le spectre de lumière visible – a été conservé. Nous avons aplati les spectres en ajustant une courbe de degré 3 aux données et divisé par celle-ci. Finalement, une interpolation linéaire de points a permis de diminuer le nombre de traits caractéristiques à 1000, ce que les algorithmes peuvent manipuler sans problème. Un échantillon de 60000 étoiles a été parti également pour 6 types spectraux différents utilisés (A, F, G, K, M, WD) a été traité. Puisque ces spectres sont les seules entrées des algorithmes essayés, le prétraitement des données a un impact majeur sur les résultats. La figure ?? présente un exemple d'un spectre d'étoile avant et après avoir subi le processus de prétraitement. Ce processus est implémenté par le module XXXXXXXX qui permet également d'extraire les données directement de SDSS.

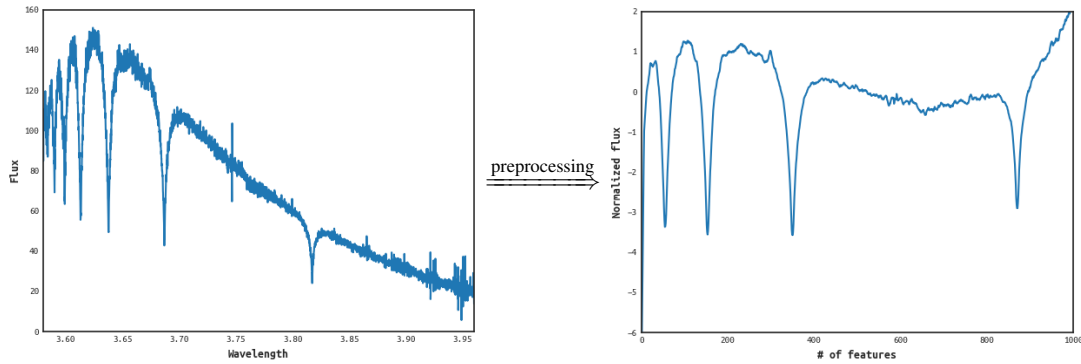


FIGURE 1 – Effet du prétraitement des spectres d'étoiles. À gauche, un exemple de données brutes fournies par le *Sloan Digital Sky Survey* est présenté. À droite, le même exemple a été normalisé et lissé.

Une première validation des algorithmes est effectuée par le biais d'une tâche connexe, soit la classification d'électrocardiogrammes selon l'état de santé du patient duquel il provient. Les électrocardiogrammes (ECG) étant fort semblables dans leurs formes à des spectres d'étoiles (tous deux étant des données corrélées dans l'espace en 1 dimension), ceci permet un premier ajustement des algorithmes en plus de nous initier au fonctionnement de ceux-ci dans le contexte d'une analyse spectroscopique. Les électrocardiogrammes utilisés proviennent du *PhysioNet/Computing in Cardiology Challenge 2017* et ont l'avantage d'être plus simple à analyser, puisqu'ils sont plus lisses et moins bruités. Une séquence de 10 secondes a été conservée pour chaque ECG et les séquences ont été classées en 2 catégories, soit un patient en santé (5050 exemples), soit un patient avec une arythmie ou un autre problème cardiaque (3478 exemples). Le nombre de traits caractéristiques pour cet ensemble de données est de 500 pour chaque exemple. Le prétraitement des données est implémenté par le module XXXXXXX.

Trois familles d'algorithmes seront étudiées dans le cadre de ce projet. Tout d'abord, nous utiliserons des réseaux de neurones de type perceptron multicouche (MLP). L'usage de bibliothèques telles Keras ou TensorFlow rend très simple l'implémentation de ce type d'algorithme. Le réseau créé prendra en entrée un spectre traité (centré, réduit et lissé) et retournera en sortie la classification du spectre selon un encodage *onehot*. La méthodologie utilisée pour ajuster les hyperparamètres (nombre de couches cachées, nombre de neurones dans chaque couche, régularisation, taille des lots) consiste à faire un *grid search* sur plusieurs architectures de réseaux ainsi que plusieurs types de régularisation et plusieurs tailles de lots pour trouver une ou plusieurs combinaisons prometteuses en mesurant l'erreur de classification sur l'ensemble de validation. Un processus de *fine tuning* des paramètres suivait ensuite pour améliorer le plus possible l'efficacité de l'algorithme. Finalement, un entraînement sur l'ensemble d'entraînement ainsi que l'ensemble de validation avait lieu, accompagné d'un test sur l'ensemble de test nous a permis d'obtenir les résultats finaux. Une présentation en détails des résultats obtenus est faite à la section ?. L'utilisation d'un processeur graphique (GPU) nous a permis d'effectuer une panoplie de différentes expériences sans que celles-ci

soient trop coûteuses en temps. Des travaux similaires ont été menés par [?] ainsi que [?] pour les spectres stellaires, et par [?] pour les électrocardiogrammes.

Ensuite, puisque les points formant un spectre sont corrélés entre eux, les réseaux de neurones convolutifs (CNN) sont une approche d'apparence prometteuse. Le réseau calculera des convolutions unidimensionnelles sur les spectres. Le nombre de convolutions, de *feature maps* ainsi que le type de *pooling* (max, moyen, etc.) souhaités sont les hyperparamètres à déterminer. Ce type d'algorithme permet de réduire le nombre de dimensions du problème en plus de prendre en compte des caractéristiques des entrées comme la connectivité locale des traits caractéristiques, tout en introduisant une invariance des traductions locales grâce au *pooling*. [?] a mené des travaux similaires pour les spectres d'étoiles. Plusieurs travaux de recherche, notamment par [?] ainsi que [?] se sont penchés sur la classification d'électrocardiogrammes à l'aide de réseaux de neurones convolutifs.

Enfin, les machines à vecteur de support (SVM) à noyau souple seront le dernier type d'algorithme d'apprentissage utilisé pour la classification des spectres stellaires ainsi que des électrocardiogrammes. Ce genre d'algorithme a tendance à bien se débrouiller avec des entrées possédant un grand nombre de traits caractéristiques. Le noyau à utiliser sera déterminé lors de la sélection du modèle. L'implémentation des SVM a été effectuée à l'aide de la librairie Scikit Learn. Des travaux similaires ont été menés par [?] pour la classification de spectres d'étoile et par [?] pour les électrocardiogrammes.

### 3 Présentation des résultats

#### 3.1 Électrocardiogrammes

Contrairement à ce à quoi nous nous attendions, nous avons eu beaucoup plus de problèmes avec les données d'électrocardiogramme qu'avec les données de spectres stellaires. Tout d'abord, pour les SVM, des tests ont été effectués pour déterminer le noyau du SVM à utiliser parmi ceux implémentés par défaut dans Scikit Learn. Nous avons conservé le noyau mou de type polynomial ( $(\gamma \langle x, x' \rangle + r)^d$ ) et un noyau à fonction à base radiale ( $\text{rbf}, \exp(-\gamma \|x - x'\|_2^2)$ ). D'autres noyaux (linéaire, sigmoid) ont également été testés, mais le faible taux de succès obtenu les ont rapidement écartés. L'effet de trois hyperparamètres a été étudié pour cet algorithme : le noyau utilisé, le degré du polynôme dans le cas d'un noyau polynomial et la valeur de la constante de régularisation  $C$  puisqu'il s'agit d'un *soft kernel* SVM. La table ?? montre les taux de classification obtenus pour ces hyperparamètres.

TABLE 1 – Taux de classification obtenu avec l'algorithme SVM pour les trois noyaux étudiés.

NOYAU	ENS. D'ENTRAÎNEMENT			ENS. DE VALIDATION		
	$C = 0.75$	$C = 1.0$	$C = 1.25$	$C = 0.75$	$C = 1.0$	$C = 1.25$
RBF	74.42%	85.68%	91.24%	59.68%	60.24%	59.61%
Poly. deg. 2	85.29%	89.45%	91.73%	61.08%	61.37%	60.94%
Poly. deg. 3	93.39%	98.03%	98.84%	59.68%	58.13%	56.09%

On note d'abord à quel point cet algorithme appliqué à cet ensemble de données est à risque de sur-apprentissage. En effet, pour chacun des noyaux, le taux classifications atteint 90% et plus dès que  $C \geq 1$ . Pourtant, pour des valeurs de  $C$  à peine inférieures à 1, on est clairement en présence de sous-apprentissage, d'autant plus que les matrices de confusions montrent que tous les échantillons sont classés dans la même catégorie. L'écroissance de la fenêtre entre ces deux points s'explique par la faible de taille de l'ensemble d'entraînement. Il apparaît évident que nous aurions davantage de latitude pour l'entraînement des paramètres si nous étions en possession d'un plus grand nombre de données. Des propositions permettant de contourner seront discutées à la section ?? . Il est aussi pertinent de se questionner sur la validité du prétraitement des données pour notre problème.

Une première tentative de diminuer le risque de sur-apprentissage est d'utiliser la technique d'analyse des composantes principales (PCA) afin de diminuer la dimension des traits caractéristiques,

similairement à ? . Les tables ?? et ?? montrent les taux de classification obtenus lorsque les 300 et les 100 premières composantes principales sont conservées. L'hypothèse était qu'une diminution du nombre de traits caractéristiques diminuerait le nombre de paramètres du SVM également, ce qui devrait résulter en une baisse de la capacité du SVM. Le même phénomène que sans l'usage de PCA se produit cependant : on passe de sous-apprentissage à sur-apprentissage sans que le taux de classification de l'ensemble de validation n'augmente.

Une dernière tentative d'améliorer les résultats est d'appliquer aux données une transformée de Fourier. En effet, puisque les données représentent un signal, il pourrait être avantageux d'étudier les fréquences principales de ce signal plutôt que le signal lui-même. Une démarche similaire a été conduite par ? pour prétraiter les entrées d'un réseau de neurones. La figure ?? en annexe montre les taux de classification obtenus pour le noyau polynomial de degré 2 pour différentes valeurs de  $C$ . La table ?? montre le taux de classification pour les différents noyaux. L'exact même phénomène se reproduit cependant : l'algorithme passe du sous-apprentissage au sur-apprentissage sans que l'erreur de classification de l'exemple de validation ne diminue. La figure ?? illustre cela : on y voit que le taux de bonnes classifications augmente pour l'ensemble d'entraînement, mais pas pour l'ensemble de validation.

On conclue que le meilleur classifieur SVM à noyau *soft* est celui envoyant tous les électrocardiogrammes à la même classe, ce qui est le cas pour le noyau polynomial de degré 2 et pour le noyau de type RBF pour  $C = 1$ . C'est deux classifieurs obtiennent des résultats d'environ 60% sur l'ensemble test, ce qui est seulement dû à la répartition des électrocardiogrammes dans les classes sain et malade. Le résultat d'un tel classifieur aurait été de 50% si les répartition avaient été égales. On conclue que le SVM s'applique mal à cet ensemble de données puisqu'il s'avère incapable de généraliser ses résultats. Notons que l'implémentation du SVM pour les données ECG est effectuée dans XXXXX

Des résultats similaires ont été obtenus avec les réseaux de neurones de type MLP. Pour l'ensemble des réseaux de neurones de ce rapport, l'optimiseur *Adam* a été utilisé, la fonction d'activation ReLU a été utilisée dans le réseau et la fonction softmax a été utilisée à la sortie. Après un *grid search* sur plusieurs architectures différentes, nous avons trouvé une architecture de réseau possédant deux couches cachées de 250 et 100 neurones, respectivement. Nous avons utilisé comme entrée des neurones les données prétraitées normalement (normalisées) ainsi que les données après y avoir effectué une transformée de Fourier. Nous avons utilisé un terme de régularisation  $\lambda$  de 0.0005 pour une régularisation de type  $L1$  et  $L2$ . La courbe d'apprentissage en fonction du nombre d'époques est présentée à la figure ?? en annexe. Comme on le voit, l'implémentation utilisant les données traitées par transformée de Fourier ont peine à dépasser 60% de taux de classification, alors que celles utilisant les données normalisées atteignent à peine 58%, alors qu'il classe tous les exemples dans la même catégorie. Le module XXXXX implémente les réseaux de neurones pour les données ECG.

Finalement, en utilisant un CNN pour faire la détection d'arythmies cardiaques, nous avons essayé plusieurs configurations pour maximiser la classification. En mettant un CNN et un réseau MLP complètement connecté bout-à-bout, nous avons vérifié l'erreur de classification en faisant varier le nombre de couches de convolution et de pooling. Nous avons gardé le MLP constant d'expérience en expérience avec une couche cachée de 50 neurones. Nous nous sommes limités à des tailles de fenêtres de convolution de largeur 4 et des filtres de pooling de largeur 4. Nous n'avons pas jugé nécessaire de tester le CNN avec les données prétraitées à l'aide de PCA ou d'une transformation de Fourier. Le taux de classification sur l'ensemble de validation en fonction du nombre de convolutions est résumé au tableau ?? en annexe. Notons que ces résultats correspondent au nombre optimal d'époques d'entraînement pour chaque combinaison d'hyperparamètres. Ceci sera également le cas pour tous les tableaux présentant des résultats des MLP et des CNN discutés dans cet article.

L'évolution de l'apprentissage en fonction du nombre d'époques d'entraînement pour le réseau convolutionnel avec 4 convolutions est présentée au tableau ?? testé sur l'ensemble de test est présentée à la figure ??.

Les résultats finaux de la précision de chaque algorithmes sur les électrocardiogrammes sont présentés au tableau ?? . Les résultats présentés sont ceux pour lesquels les meilleurs résultats ont été obtenus.

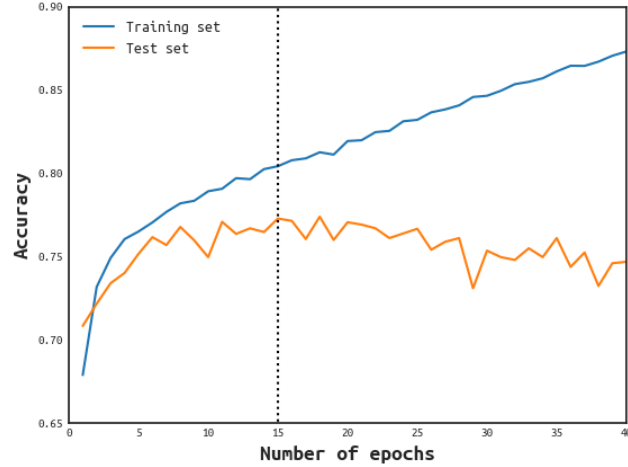


FIGURE 2 – Évolution du taux de classification sur les données d’ECG en fonction du nombre d’époques d’entraînement pour le réseau de neurones convolutif avec 4 convolutions.

TABLE 2 – Résultats pour la classification d’ECG sur l’ensemble de test pour chaque algorithme utilisé.

ALGORITHME	SVM	MLP	CNN
PRÉCISION	61.37%	60.69%	77.39%

Comme attendu, on remarque que le taux de classification des CNN est de plus de 16% plus élevé que celui obtenu avec les autres algorithmes. Cette situation avait été anticipée, puisque les traits caractéristiques des signaux d’ECG sont autocorrélés. Le module XXXXX implémente les CNN pour les données ECG.

### 3.2 Spectres d’étoiles

Les résultats obtenus pour la classification de spectres stellaires ont été beaucoup plus concluants que ceux obtenus pour la classification d’ECG. Tout d’abord, avec les SVM, nous avons sélectionné un noyau RBF, puis avons essayé plusieurs valeurs pour la constante de pénalité  $C$ . Les résultats obtenus sont présentés à la figure ?? en annexe. On voit que la constante  $C$  a un effet important sur les résultats obtenus, et que la valeur optimale de  $C$  s’avère être \*\*\*\*\*. Le module XXXXX implémente le SVM pour les données de SDSS.

Pour les réseaux de neurones, nous avons essayé plusieurs architectures au hasard et avons aussi modifier la taille des lots et la régularisation pour l’apprentissage. Encore une fois, les non-linéarités choisies dans le réseau étaient de type ReLU et celles à la sortie étaient de type *softmax*. La table ?? en annexe présente le taux de classification obtenu pour différentes architectures de réseau de neurones. L’effet de l’architecture sur le taux d’apprentissage était minime, ne faisant varier celui-ci que par moins de 1% à taille de lot constant (50) et sans régularisation. Les résultats de classification dépendent donc davantage du *seed* que de ces hyperparamètres.

Le second hyperparamètre à ajuster pour régler la capacité du modèle est la taille des lots. Pour le MLP avec les couches cachées [150, 100, 50], plusieurs tailles de lots ont été testées et les résultats sont rapportés à la table ?? . Encore une fois, le taux de classification ne varie que de très peu pour différentes tailles de lots. Finalement, nous avons essayé plusieurs types de régularisations pour le même réseau de neurones. La figure ?? présente l’effet de la régularisation sur le taux de classification maximale. Les résultats obtenus grâce

À la régularisation n'ont pas non plus eu beaucoup d'influence sur le taux de classification. Les réseaux de neurones sont implémentés pour les données de SDSS par le module XXXXX.

Enfin, un algorithme de type réseau de neurones convolutif a été étudié pour l'ensemble de données SDSS. Les hyperparamètres étudiés sont le nombre de convolution ainsi que le nombre de *feature maps*, le *batch size* et le nombre de couches cachées ainsi que le nombre de neurones les composant du réseau de neurones transmettant le vecteur de sortie du CNN à la sortie du réseau. Mentionnons également qu'un *max pooling* est effectué après chaque convolution.

D'abord, le nombre de convolutions effectuées a été étudié pour des nombres variables de *feature maps*. Pour ces essais, le réseau de neurone à la sortie du CNN a été gardé identique avec une couche cachée de taille 10 et fonctions d'activation ReLU et softmax. Les *batch size* sont également demeurés constants à 50. La table ?? en annexe montre les taux de classifications obtenus pour l'ensemble de validation pour quelques unes de ces combinaisons. Les résultats obtenus sont relativement similaires, mais tentons d'optimiser les autres hyperparamètres pour la configuration [10, 50].

Essayons différents *batch size* afin de pouvoir observer l'effet de cet hyperparamètre. La table ?? en annexe montre les résultats obtenus pour des *batch size* de respectivement 5, 50 et 500. On voit que le choix des *batch size* n'a que peu d'influence sur le taux de classification. Néanmoins, puisque ce sont des *batch size* de 50 qui donnent de (légèrement) meilleurs résultats, prenons cette valeur pour la suite des choses.

Enfin, optimisons la taille de la couche cachée du réseau de neurone en sortie du CNN. Les architectures considérées sont celles se trouvant à la table ?? en annexe. Encore une fois, on voit qu'il n'y a que très peu de différence entre les différentes valeurs de ces hyperparamètres. On note cependant que c'est une seule couche cachée de taille 50 qui a donné les meilleurs résultats.

Concluons cette analyse en notant que les hyperparamètres n'avaient pratiquement aucun effet sur les taux de classification obtenus. Cela signifie probablement que l'ensemble de données est en général très facile à classer. On peut imaginer que les données forment des *clusters* bien définis dans l'espace des traits caractéristiques sauf pour environ 5% des spectres qui se mélangent à des *clusters* d'une autre classe que la leur. Il est donc facile d'obtenir environ 94% de bonnes classifications sur un ensemble test, mais il serait très ardu d'obtenir davantage. On remarque également que le taux de classification pour l'ensemble d'entraînement, bien que plus élevé que celui de l'ensemble de validation, n'a jamais atteint 100%, ce qui est cohérent avec l'hypothèse proposée. Le module XXXXX implémente les CNN pour les données de SDSS.

Les résultats finaux de la précision de chaque algorithme sur les spectres stellaires sont présentés au tableau ??.

TABLE 3 – Résultats pour la classification de spectres d'étoiles pour chaque algorithme utilisé.

ALGORITHME	SVM	MLP	CNN
PRÉCISION	%	%	77.17%

## 4 Conclusion

## 5 Répartition et remerciements

## Annexe

TABLE 4 – Taux de classification obtenu avec l’algorithme SVM pour les trois noyaux étudiés avec utilisation du PCA avec les 300 premières composantes principales conservées dans le prétraitement des données pour l’ensemble de données d’ECG.

NOYAU	ENS. D’ENTRAÎNEMENT			ENS. DE VALIDATION		
	$C = 0.75$	$C = 1.0$	$C = 1.25$	$C = 0.75$	$C = 1.0$	$C = 1.25$
RBF	86.51%	95.62%	98.03%	58.97%	58.55%	58.48%
Poly. deg. 2	93.56%	95.34%	96.13%	53.91%	53.48%	53.55%
Poly. deg. 3	99.70%	99.88%	99.89%	55.03%	55.17%	54.54%

TABLE 5 – Taux de classification obtenu avec l’algorithme SVM pour les trois noyaux étudiés avec utilisation du PCA avec les 100 premières composantes principales conservées dans le prétraitement des données pour l’ensemble de données d’ECG.

NOYAU	ENS. D’ENTRAÎNEMENT			ENS. DE VALIDATION		
	$C = 0.75$	$C = 1.0$	$C = 1.25$	$C = 0.75$	$C = 1.0$	$C = 1.25$
RBF	97.98%	99.63%	99.86%	59.11%	58.34%	58.90%
Poly. deg. 2	91.05%	92.42%	93.77%	52.99%	53.41%	53.55%
Poly. deg. 3	99.98%	99.98%	99.98%	52.15%	52.01%	51.58%

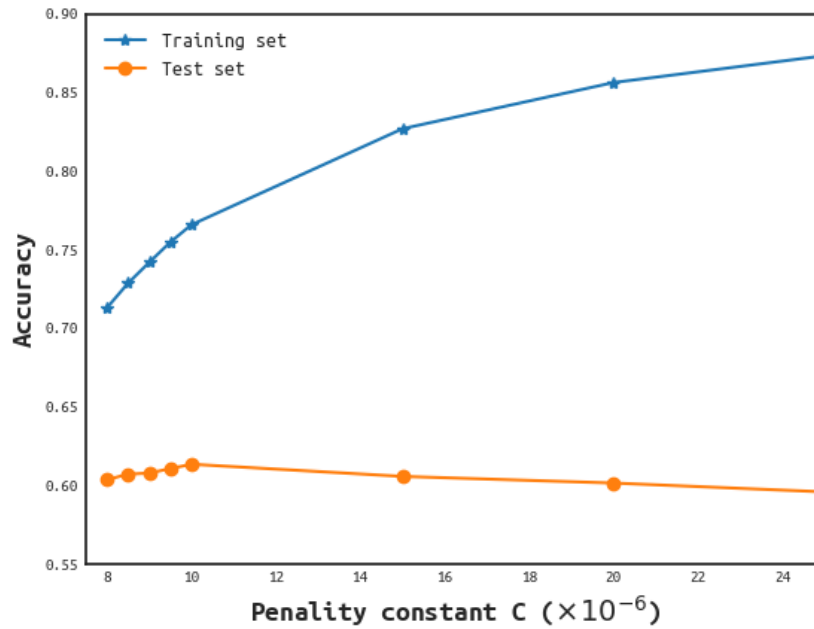


FIGURE 3 – Évolution du taux de classification sur les données d’ECG en fonction de la constante de pénalité  $C$  pour l’algorithme SVM avec des données transformées à l’aide d’une transformée de Fourier.

TABLE 6 – Taux de classification obtenu avec l’algorithme SVM pour les trois noyaux Étudiés avec l’utilisation d’une transformée de Fourier lors du prétraitement des données pour l’ensemble de données d’Électrocardiogrammes.

NOYAU	ENS. D’ENTRAÎNEMENT			ENS. DE VALIDATION		
	$C = 0.5$	$C = 1.0$	$C = 1.5$	$C = 0.5$	$C = 1.0$	$C = 1.5$
RBF	59.54%	100.00%	100.00%	58.90%	58.90%	58.90%
Poly. deg. 2	100.00%	100.00%	100.00%	55.52%	55.52%	55.52%
Poly. deg. 3	100.00%	100.00%	100.00%	59.68%	54.40%	54.40%

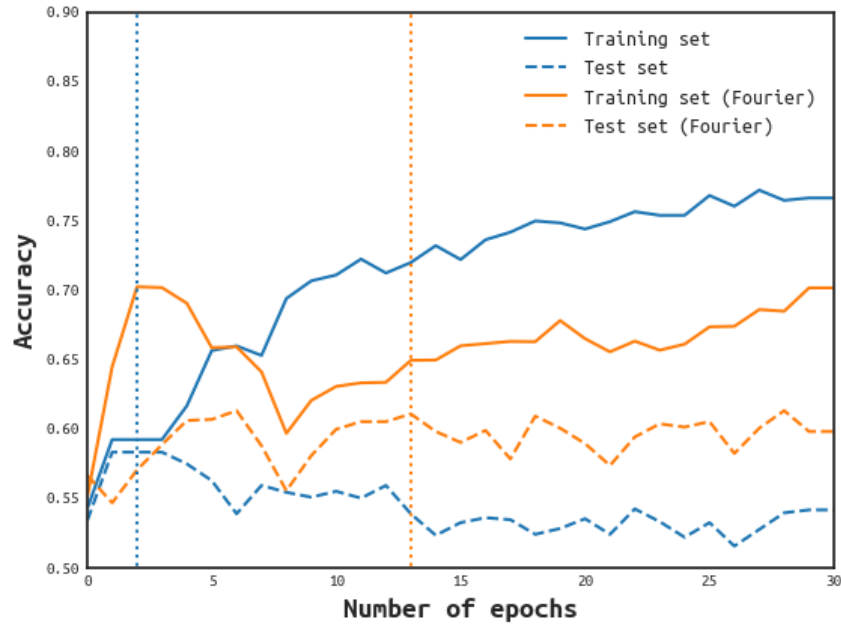


FIGURE 4 – Évolution du taux de classification sur les données d’Électrocardiogramme en fonction du nombre d’époques d’entraînement pour le réseau de neurones de type MLP avec deux couches cachées de 250 et 100 neurones, respectivement.

TABLE 7 – Taux de classification en fonction du nombre de convolutions dans le réseau de neurones convolutionnel pour les données d’Électrocardiogramme.

CONFIG.	ENS. D’ENTRAÎNEMENT	ENS. DE VALIDATION
[100]	0.7962	0.6882
[50, 100]	0.7917	0.7476
[10, 50, 100]	0.7922	0.7672
[10, 50, 100, 200]	0.8198	0.7792



TABLE 8 – Taux de classification obtenu avec l’algorithme DNN pour l’ensemble de données SDSS pour différentes architectures.

CONFIG.	ENS. D’ENTRAÎNEMENT	ENS. DE VALIDATION
[50]	0.9732	0.9393
[100]	0.9774	0.9389
[200]	0.9553	0.9393
[400]	0.9778	0.9396
[200, 100]	0.9783	0.9439
[200, 50]	0.9715	0.9437
[500, 100]	0.9791	0.9436
[150, 100, 50]	0.9792	0.9456
[500, 100, 50]	0.9812	0.9431

TABLE 9 – Taux de classification obtenu avec l’algorithme DNN pour l’ensemble de données SDSS en fonction *batch size* pour l’architecture [150, 100, 50].

BATCH SIZE	ENS. D’ENTRAÎNEMENT	ENS. DE VALIDATION
5	0.9684	0.9431
50	0.9729	0.9408
500	0.9794	0.9441

TABLE 10 – Taux de classification obtenu avec l’algorithme CNN pour l’ensemble de données SDSS en fonction du nombre de filtres de convolution par couche de convolution.

CONFIG.	ENS. D’ENTRAÎNEMENT	ENS. DE VALIDATION
[10]	0.9741	0.9375
[50]	0.9902	0.9423
[100]	0.9862	0.9447
[10, 50]	0.9600	0.9457
[50, 100]	0.9896	0.9440
[10, 50, 100]	0.9741	0.9455

TABLE 11 – Taux de classification obtenu avec l’algorithme CNN pour l’ensemble de données SDSS en fonction de la taille des lots.

BATCH SIZE	ENS. D’ENTRAÎNEMENT	ENS. DE VALIDATION
5	0.9475	0.9413
50	0.9442	0.9440
500	0.9577	0.9431

TABLE 12 – Taux de classification obtenu avec l’algorithme CNN pour l’ensemble de données SDSS en fonction de l’architecture du réseau à la sortie du CNN.

CONFIG.	ENS. D’ENTRAÎNEMENT	ENS. DE VALIDATION
[20]	0.9591	0.9425
[50]	0.9626	0.9436
[100]	0.9802	0.9427
[20, 50]	0.9579	0.9404
[50, 100]	0.9768	0.9429
[20, 50, 100]	0.9435	0.9407