Transposition de données multidimensionnelles en images pour pallier le fléau de la dimension

Rebecca Leygonie, Sylvain Lobry, Guillaume Vimont, Laurent Wendling

Transposition de données multidimensionnelles en images pour pallier le fléau de la dimension

Rebecca Leygonie
Sylvain Lobry
Guillaume Vimont
Laurent Wendling

Université Paris Cité, LIPADE, F-75006 Paris, France

Rebeccca.leygonie@etu.u-paris.fr

Résumé

Lorsqu'on traite des problèmes de classification de séries temporelles multivariées à haute dimension, une difficulté bien connue est le fléau de la dimension (curse of dimensionality). Dans cet article, nous proposons une approche de transposition de données multidimensionnelles en images pour aborder la tâche de classification. Nous proposons un petit modèle hybride contenant des couches convolutionnelles comme extracteur de caractéristiques suivies d'un réseau de neurones récurrent qui prend ces données transposées comme entrée. Nous appliquons notre méthode à un grand ensemble de données constitué de dossiers médicaux de personnes résidant en France. Nous montrons que notre approche permet de réduire significativement la taille d'un réseau et d'augmenter ses performances en optant pour une transformation des données d'entrée.

Mots Clef

Fléau de la dimension, séries temporelles multivariées, classification, réseaux de neurones à convolution, réseaux de neurones récurrents.

Abstract

When dealing with high-dimensional multivariate time series classification problems, a well-known difficulty is the curse of dimensionality. In this article, we propose an original approach of transposition of multidimensional data into images to tackle the task of classification. We propose a small hybrid model containing convolutional layers as a feature extractor followed by a recurrent neural network that take this transposed data as an input. We apply our method to a large dataset consisting of individual patient medical records. We show that our approach allows us to significantly reduce the size of a network and increase its performance by opting for a transformation of the input data.

Keywords

Curse of dimensionality, multivariate time series, classification, convolutionnal neural network, recurrent neural network.

1 Introduction

Lorsque l'on considère des données à haute dimension, de nombreuses méthodes de reconnaissance de formes supervisées peuvent échouer à capturer des informations pertinentes. En effet, lorsque le nombre de dimensions augmente, la quantité de données nécessaires pour conserver une densité similaire doit croître de manière exponentielle. Ce phénomène est généralement connu sous le nom de fléau de la dimension (*curse of dimensionality*) [1]. En plus de la difficulté de former des modèles supervisés efficaces, deux problèmes se posent fréquemment lors de l'étude de problèmes à haute dimension : la complexité temporelle et la perte d'information.

Pour résoudre le problème de la complexité temporelle, une méthode souvent utilisée consiste à réduire les dimensions de l'ensemble de données [14]. Cela peut être réalisé par des approches de sélection des caractéristiques pour choisir les colonnes pertinentes à conserver (*Recursive Feature Elimination (RFE)*, ANOVA, *Chi-Square* et la corrélation de Pearson) ou par des approches de factorisation matricielle qui réduisent une matrice de données en ses parties constituantes (*Principal Component Analysis (PCA)* [6] et *Linear Discriminant Analysis (LDA)* [2]).

Souvent, leur application entraîne une perte d'information, ce qui n'est pas acceptable dans certains domaines. L'utilisation de méthodes d'apprentissage profond est de plus en plus courante pour la tâche de classification. Les réseaux de neurones convolutifs (CNN) [8] sont utilisés pour extraire des patterns dans les images et les réseaux récurrents [10] pour prendre en compte la temporalité des séries temporelles. Motivés par ces avancées récentes, notre objectif est de trouver une approche qui réduise le coût de computationnel de la tâche de classification sur des données multidimensionnelles sans perdre d'information.

La génération d'images à partir de données est une technique qui a été appliquée dans plusieurs domaines. Afin de diagnostiquer le cancer du sein, les auteurs de [12] utilisent des métriques sur les données telles que les diagrammes à barres équidistantes et la matrice de distance normalisée. Dans [9], les auteurs transforment des fichiers binaires en images pour la détection de logiciels malveillants. Dans

[13], les séries temporelles des tests d'exercice cardiopulmonaire sont codées en images en utilisant un champ angulaire de Gram et un champ de transition de Markov. Dans notre travail, nous cherchons à développer le même type d'approche et à l'appliquer sur un grand ensemble de données. La transposition des tableaux de données en images permettrait de contourner les approches classiques de réduction de dimension tout en ne fixant aucune limite à la taille du jeu de données.

Le Système National des Données de Santé (SNDS) [1] regroupe les principales bases de données de santé publique françaises existantes. Ce regroupement de données vise à améliorer les connaissances sur la prise en charge médicale et à élargir le champ des recherches, études et évaluations dans le domaine de la santé. Pour chaque individu vivant en France, le jeu de données contient :

— les données de l'assurance maladie issues de la base Système National d'Information Inter Régimes de l'Assurance Maladie (SNIIRAM) dont le datamart de consommation inter régime (DCIR) qui recense les dépenses médicales des patients de la ville.

— les données de la base Programme de Médicalisation des Systèmes d'Information (PMSI) qui couvre l'hospitalisation dans les établissements publics et privés en France métropolitaine et dans les départements d'outre-mer.

— des données sur les causes de décès issues de la base Centre d'épidémiologie sur les causes médicales de décès (CépiDc) - INSERM.

Après avoir joint les différentes tables, le jeu de données du SNDS contient, pour chaque individu vivant en France, une série temporelle d'informations médicales (visites à l'hôpital ou chez le médecin, remboursement de médicaments, etc.) Le SNDS contient 191 tables et 4 853 variables caractéristiques, il est donc typiquement sujet au *fléau de la dimension* mentionné plus haut.

Notre objectif est de réaliser une classification des pathologies sur les données du SNDS sans extraire les caractéristiques en amont. Pour traiter ce problème général tout en restant exhaustif dans l'exploitation des variables, nous proposons de transposer les données en séries temporelles d'images représentant le parcours médical de chaque individu. Chaque image de la série temporelle est une grille de pixels où la couleur et la position de chaque pixel sont associées à une caractéristique de l'événement médical.

Une fois les données transformées en séries temporelles d'images, nous appliquons un modèle de classification hybride, composé d'un CNN comme extracteur de caractéristiques, suivi d'un LSTM [5]. Cette approche est illustrée dans la figure 1. Nous comparons notre modèle à des approches de classification de séries temporelles multivariées couramment utilisées, telles que le LSTM et le Transformer [15], ainsi qu'à un modèle hybride avec un CNN à couches convolutives unidimensionnelles (1DCNN) [7].

1. https ://www.snds.gouv.fr/SNDS/Accueil

Nos contributions présentées dans cet article sont (1) La transposition de données multidimensionnelles en images pour surmonter le *fléau de la dimension*. (2) La proposition d'un modèle hybride léger permettant la reconnaissance de formes dans les images ainsi que la corrélation entre les pas de temps d'une série temporelle. (3) Validation de notre approche pour la tâche de classification multi-classes sur des données de santé bruitées.

Cet article a été soumis à la conférence *2023 IEEE International Conference on Image Processing*.

2 Méthodologie

Nous considérons un ensemble de données $X = (X_i, y_i)_{i\in[\![1;n]\!]}$ de n séries temporelles multivariées. Plus précisément, chaque X_i est un vecteur $(x_1^i, ..., x_T^i)$, où T est un entier (l'*horizon temporel*), et chaque x_k^i, pour $1 \leq k \leq T$, est un vecteur de $\mathbb{R}^d$, où $d \in \mathbb{N}$:

$$\forall i \in [\![1, n]\!], \forall k \in [\![1, d]\!], x_k^i = [x_{k,1}^i, ..., x_{k,d}^i] \in \mathbb{R}^d \quad (1)$$

Chaque y_i est un vecteur $\in [\![1, C]\!]$, où $C \in \mathbb{N}$ est le nombre de classes. Notre objectif est de concevoir un modèle permettant de prédire $\hat{y}_i \; dans [\![1, C]\!]$.

2.1 Traitement des données

Comme nous l'avons vu dans l'introduction, un grand nombre d'observations (c'est-à-dire un grand T) de données à haute dimension (c'est-à-dire un grand d) peut être soumis au *fléau de la dimension*. Nous proposons de transformer chaque X_i en une série temporelle $\tilde{X}_i$ d'images de taille $m \times m$:

$$\tilde{X}_i = (\tilde{x}_1^i, ...\tilde{x}_T^i) \in \mathbb{R}^{T\times m\times m},$$

c'est à dire que chaque $\tilde{x}_k^i$ pour $1 \leq k \leq T$, est une matrice de taille $m \times m$.

Pour transformer le vecteur x_k^i dans $\mathbb{R}^d$ en une matrice $\tilde{x}_k^i$ dans $\mathbb{R}^{m\times m}$ nous appliquons les étapes suivantes :

(i) poser $m = \lceil\sqrt{d}\rceil$,

(ii) remplir x_k^i avec $m^2 - d$ zéros (*padding*) pour obtenir un vecteur dans $\mathbb{R}^{m^2}$,

(iii) projeter le vecteur x_k^i en une matrice de taille $m \times m$.

2.2 Modèle de classification

Nous proposons d'utiliser un modèle de classification qui consiste en un CNN pour extraire les caractéristiques, suivi d'un LSTM pour obtenir la prédiction finale basée sur toutes les étapes temporelles :

1. Le CNN transforme une image de $\mathbb{R}^{m\times m}$ en un vecteur de $\mathbb{R}^N$, où N est un entier arbitraire. En appliquant le CNN à chaque image de la série temporelle $\tilde{X}_i \in \mathbb{R}^{T\times m\times m}$, on obtient une série $L \in \mathbb{R}^{T\times N}$.

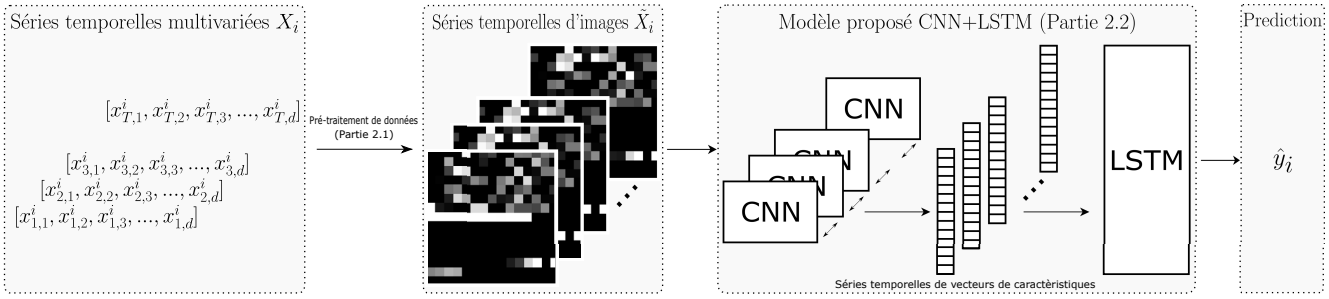

FIGURE 1 – Vue d'ensemble de l'approche proposée. Tout d'abord, la série temporelle multivariée X_i est transformée en une série temporelle d'images $\tilde{X}_i$. Chaque image de cette série temporelle est ensuite transmise à un CNN unique pour extraire des vecteurs de caractéristiques. La série de vecteurs de caractéristiques est ensuite utilisée comme entrée d'un modèle LSTM pour obtenir une prédiction.

2. Le LSTM, appliqué à L, renvoie la prédiction $\hat{y}_i \in [\![1, C]\!]$.

Nous comparons ce modèle aux modèles couramment utilisés pour les séries temporelles multivariées : LSTM [5], $1D$CNN-LSTM et Transformers [15]. Les architectures des modèles utilisés sont détaillées dans la sous-section 3.3 et la comparaison entre les modèles est décrite dans la section 4.

3 Expérimentations

3.1 Construction du jeu de données

Afin de garantir l'anonymat des patients, nous travaillons sur une version pseudonymisée du SNDS. Nous nous intéressons aux données 2014 et 2015 du SNDS et nous utilisons 12 tables du DCIR, 6 tables du PMSI MCO (Médecine, Chirurgie et Obstétrique), et les 2 tables Bénéficiaires, nous permettant de chaîner les données. Nous agrégeons toutes les informations dans une seule base de données en joignant les tables à l'aide de l'identifiant du patient. Dans la base de données finale, nous appelons chaque ligne une *observation* ou un *événement médical*. Chaque observation contient un identifiant de patient et est donc associée à un individu. Pour obtenir notre jeu de données X, nous regroupons les événements médicaux par identifiant de patient. Dans X, chaque élément X_i représente un patient et sa série temporelle de T événements médicaux qui sont des vecteurs de $\mathbb{R}^d$ comme dans l'équation 1.

Pour les expériences présentées, afin de comparer le modèle avec des architectures plus importantes, nous sélectionnons des individus dont l'historique médical comportait au maximum $T = 500$ événements entre 2014 et 2015.

Nous entraînons notre modèle sur la tâche de classification de pathologies. Afin d'assurer une distribution homogène, nous sélectionnons les individus présentant une des 6 pathologies les plus représentées dans notre jeu de données : (c_1) tumeurs, (c_2) maladies endocriniennes, nutritionnelles et métaboliques, (c_3) troubles mentaux et comportementaux, (c_4) maladies du système circulatoire, (c_5) maladies du système ostéo-articulaire, des muscles et du tissu conjonctif, et (c_6) lésions traumatiques, empoisonnements et certaines autres conséquences de causes externes. Pour optimiser l'interprétation des résultats et comprendre les caractéristiques importantes liées à la prédiction d'une pathologie, nous choisissons d'effectuer une classification multi-classes mono-label. Cela signifie que nous entraînons le modèle à reconnaître plusieurs pathologies (multi-classes) mais à n'en associer qu'une seule par individu (mono-label). Ainsi, nous sélectionnons les patients affectés par une seule pathologie.

Finalement, le jeu de données contient $n = 1000$ séries temporelles X_i d'événements médicaux avec une distribution équilibrée entre les $C = 6$ classes. Nous divisons le jeu de données en un ensemble d'apprentissage (65%), un ensemble de validation (15%) et un ensemble de test (20%). Nous utilisons un échantillonnage aléatoire tout en conservant un ensemble de données équilibré. Chaque événement médical est décrit par 347 caractéristiques qui peuvent être des informations sur le patient ou des informations provenant des remboursements effectués par tous les régimes d'assurance maladie pour les soins en ville ou à l'hôpital. Ces caractéristiques peuvent être catégorielles ou continues. Nous expliquons comment nous traitons ces différences dans la section suivante.

3.2 Des données médicales aux données temporelles multivariées

Nous nettoyons les données en éliminant les doublons, en supprimant les colonnes contenant uniquement des valeurs manquantes et en supprimant les colonnes qui sont implicitement fortement corrélées avec la cible à prédire .

Dans le jeu de données, certaines observations ont des valeurs manquantes, comme la date de décès si le patient est vivant. Nous voulons attribuer une valeur spécifique à la valeur manquante car l'absence de cette valeur est une information pour le modèle. Les données numériques de l'ensemble de données sont supérieures ou égales à 0. Nous les incrémentons toutes de 1 afin d'attribuer la valeur 0 à toutes les données manquantes. Ainsi, le modèle sera capable de comprendre que la valeur 0 est associée à une valeur manquante, sur laquelle il ne faut pas apprendre.

Un événement médical est représenté par des caractéristiques à valeur réelle, telles que la taille et le poids du patient, et des caractéristiques à valeur catégorielle, telles que le sexe du patient ou le code représentant le type d'événement médical. Afin d'avoir un type unique dans l'ensemble de données, nous appliquons du *label encoding* sur toutes les caractéristiques catégorielles. Cette approche consiste à transposer les caractéristiques catégorielles en caractéristiques discrètes.

Chaque série temporelle peut avoir un horizon temporel différent T. Cependant, les modèles que nous utilisons ne peuvent pas gérer des séquences de tailles différentes. Ainsi, nous effectuons un remplissage (en ajoutant des observations contenant uniquement des zéros) au début des séries temporelles afin qu'elles aient toutes le même nombre d'observations T. Cette méthode est appelée *padding*

Enfin, après avoir nettoyé les données et obtenu le nombre maximal T d'événements médicaux en comparant la longueur de chaque série temporelle, dans le jeu de données final X, chaque X_i est une série temporelle dans $\mathbb{R}^{T\times d}$ avec $T = 500$ et $d = 347$.

3.3 Modèles et référence

Nous comparons notre modèle de classification considérant une représentation en image des données décrites dans la section 2.1 à trois méthodes classiques basées sur l'apprentissage profond pour le traitement de séries temporelles multivariées. Pour toutes les expériences, nous utilisons une perte d'entropie croisée et les modèles sont optimisés en utilisant l'optimiseur de descente de gradient stochastique avec un taux d'apprentissage de 0,01. Nous utilisons un *batch size* de 128 et entraînons les modèles pendant 500 *epochs*. De plus, nous réduisons le taux d'apprentissage en le divisant par un facteur 10 lorsque la perte de validation a cessé de diminuer après 5 *epochs*.

Notre modèle utilise un CNN comme extracteur de caractéristiques suivi d'un LSTM qui a la particularité de prendre en compte la temporalité des séries temporelles. Nous souhaitions trouver l'architecture la plus légère pour minimiser la complexité temporelle de la tâche. Notre architecture CNN se compose de trois couches convolutives 2D avec 3, 16 et 32 filtres ayant respectivement une taille de 3×3, 5×5 et 3×3. Nous ajoutons une normalisation *batch normalization* entre chaque couche. La fonction d'activation ReLU est utilisée pour chaque couche. Nous ajoutons une couche entièrement connectée avec une sortie de taille $N = 50$ qui correspond au vecteur des caractéristiques extraites. L'architecture CNN est suivie par un LSTM avec une couche cachée de taille 50, une couche entièrement connectée avec une sortie de taille 6 et une couche de *dropout* avec une probabilité de 0.4. Après avoir appliqué les pré-traitements décrits dans la section 2.1, nous normalisons les valeurs des séries temporelles de telle sorte que $\tilde{X}_i \in [\![0, 255]\!]^{T\times m\times m}$.

Afin de justifier notre approche, nous comparons notre modèle à un CNN contenant des couches convolutionnelles $1D$ suivies d'un LSTM. Le modèle est composé de 3 couches convolutionnelles $1D$ avec 16, 16 et 32 filtres ayant une taille de 5, 3 et 3 filtres. La fonction d'activation ReLU est utilisée pour chaque couche. L'architecture $1D$CNN est suivie par un LSTM avec une couche cachée de taille 100 et une couche entièrement connectée avec une sortie de taille 6. Nous comparons également notre modèle à un réseau simple comportant deux couches LSTM avec des couches cachées de taille 100 et 50, suivies d'une couche entièrement connectée de taille 6 en sortie. Concernant le pré-traitement des données, pour tenir compte des grands nombres, $log(x+1)$ est appliqué à toutes les entrées X_i du LSTM et du $1D$CNN-LSTM. Enfin, nous comparons notre modèle à un transformeur considéré comme état de l'art, BERT [4] [2]. Tous les réseaux sont entraînés sur un serveur doté de 64 processeurs, de 256 Go de RAM et d'un GPU Quadro P6000 avec 24 Go de RAM.

4 Résultats

Les résultats présentés dans le tableau 1 montrent que la tâche de classification de séries temporelles d'images avec le modèle CNN-LSTM atteint de meilleures performances que les modèles prenant des données 1D en entrée. En effet, lors de l'utilisation de réseaux récurrents sur des séries temporelles longues, avec de nombreux pas de temps, les modèles souffrent du problème nommé *vanishing gradient problem*, ce qui conduit à une mauvaise performance du réseau. Nous remarquons également que le CNN-LSTM est plus efficace que le LSTM seul. On en conclut que la méthode d'extraction de caractéristiques en amont de la classification influence les performances. Enfin, le CNN-LSTM avec des convolutions 2D obtient une précision supérieure de 26% à celle du $1D$CNN-LSTM. Ainsi, la taille des séquences étant trop longue, la pertinence de transposer les données en images est confirmée.

5 Discussion

Le modèle que nous proposons, l'approche de transposition de données tabulaires en images, combinée à l'application d'un petit modèle hybride CNN-LSTM surpasse les modèles avec lesquels il est comparé. En effet, les modèles considérés comme l'état de l'art pour la tâche de classification de séries temporelles ont des performances proches de l'aléatoire sur notre problème. Nous pensons que les modèles appliqués aux données brutes sont limités par la taille des séquences. En effet, plus les séquences sont longues, plus le modèle a des difficultés à extraire les corrélations entre les caractéristiques. La taille des séquences est également problématique pour l'utilisation des approches classiques de classification. En raison de leur complexité temporelle, nous n'avons pas pu appliquer les modèles *Time series forest (TSF)* [3] et *Bag-of-SFA-Symbols Ensemble (BOSSEnsemble)* [11] sur les données brutes.

Nous avons également testé différentes architectures du CNN-LSTM. Nous remarquons que l'ajout de couches de

2. https ://huggingface.co/bert-base-cased

modèle	précision	temps d'apprentissage
BERT	21%	93m 19s
$1D$CNN-LSTM	20%	61m 27s
LSTM	19%	19m 16s
Modèle proposé (CNN-LSTM)	**46%**	30m 21s

TABLE 1 – Performances et temps d'apprentissage des modèles de classification.

pooling fait baisser la précision de 46% à 30%. L'information de nos images résidant dans chaque pixel, l'agrégation spatiale génère une perte d'information et donc de performances du modèle.
Le modèle que nous proposons a été entraîné sur des données complexes et très bruitées. Qu'il s'agisse de l'ajout de *padding*, de données manquantes ou encore de caractéristiques non corrélées à la détection de la pathologie, nous pensons atteindre la limite des performances de la tâche de classification de pathologies sur notre échantillon de données. Au vu des résultats obtenus, nous pensons que notre approche peut apporter une réelle valeur ajoutée sur des séries temporelles longues et multivariées. En effet, les images que nous générons à partir du SNDS sont de taille (19×19) ce qui est loin des dimensions spatiales prises en entrée des CNN modernes (par exemple, les réseaux préentraînés sur *ImageNet* prennent souvent 224×224 en entrée). Cela signifie qu'il serait concevable de générer des images à partir de données à plus haute dimension avec notre approche.

6 Conclusion

Dans cette étude, nous avons proposé une nouvelle approche pour la classification des séries temporelles en transformant les données à haute dimension en représentations basées sur les images. Il s'agit d'une contribution importante car elle nous permet de surmonter le *fléau de la dimension*, qui est un problème courant dans l'analyse des données à haute dimension. Les méthodes traditionnelles nécessitent souvent une réduction de dimension, mais cela peut entraîner une perte d'informations importantes. En utilisant des représentations basées sur l'image, nous sommes en mesure de conserver toutes les informations contenues dans les données tout en étant capables d'utiliser de puissantes techniques de traitement d'image.
Une autre contribution clé de notre approche est l'utilisation de petites et légères architectures. En utilisant de petits modèles, nous sommes en mesure d'entraîner nos modèles en un temps raisonnable, ce qui en fait une solution pratique pour les applications du monde réel. Nous avons proposé un modèle hybride qui combine les avantages des CNN et des LSTM. La transposition des données de séries temporelles en séries temporelles d'images nous permet d'exploiter la capacité des CNN à extraire des caractéristiques pertinentes des images, qui peuvent ensuite être transmises au LSTM, capable de capturer les dépendances temporelles dans les données. Nous avons constaté qu'en utilisant cette approche hybride, notre modèle a été en mesure d'obtenir des performances nettement meilleures que les modèles récurrents traditionnels qui sont couramment utilisés pour la classification des séries temporelles.
Les travaux futurs comprennent la focalisation sur l'explicabilité de ces modèles afin de permettre une transparence dans leur utilisation dans des domaines sensibles tels que le domaine médical.

Références

[1] Richard Bellman. Adaptive Control Processes - A Guided Tour, 1961.

[2] Patricia Cohen, Stephen G West, and Leona S Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Psychology press, 2014.

[3] Houtao Deng, George Runger, Eugene Tuv, and Martyanov Vladimir. A time series forest for classification and feature extraction. *Information Sciences*, 239 :142–153, 2013.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics, 2019.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997.

[6] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6) :417, 1933.

[7] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. Real-time patient-specific ECG classification by 1-D Convolutional Neural Networks. *IEEE Transactions on Biomedical Engineering*, 63(3) :664–675, 2016.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.

[9] Lakshmanan Nataraj, Sreejith Karthikeyan, Gregoire Jacob, and Bangalore S Manjunath. Malware images : visualization and automatic classification. In *Proceedings of the 8th international symposium on visualization for cyber security*, pages 1–7, 2011.

[10] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088) :533–536, 1986.

[11] Patrick Schäfer. The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6) :1505–1530, 2015.

[12] Anuraganand Sharma and Dinesh Kumar. Classification with 2-d convolutional neural networks for breast cancer diagnosis. *Scientific Reports*, 12(1) :21857, 2022.

[13] Yash Sharma, Nick Coronato, and Donald E Brown. Encoding cardiopulmonary exercise testing time series as images for classification using convolutional neural network. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1611–1614. IEEE, 2022.

[14] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction : a comparative. *J Mach Learn Res*, 10(66-71) :13, 2009.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.