



Reconnaissance Intelligente de l'activité Humaine en Temps Réel à l'aide de Signaux Wi-Fi

Hadi EL ZEIN^{1,2}, Farah MOURAD-CHEHADE¹, et Hassan AMOUD²

¹ Laboratoire Informatique et Société Numérique (LIST3N), Université de Technologie de Troyes, Troyes, France

² Plateforme Azm pour la Recherche en Biotechnologie et ses Applications, Université Libanaise, Tripoli, Liban

Correspondance: hadi.el_zein@utt.fr, farah.chehade@utt.fr, hassan.amoud@gmail.com

Résumé

La reconnaissance de l'activité humaine ou **Human Activity Recognition (HAR)** est une tâche importante dans de nombreuses applications de santé à domicile, telles que la détection des chutes en temps réel ou la surveillance de l'état de la santé des personnes âgées. Cet article propose une nouvelle solution HAR basée sur l'information d'état du canal des signaux Wi-Fi ou Channel State Information (CSI). L'approche proposée utilise les techniques d'apprentissage profond avec la technologie de communication Wi-Fi pour proposer une méthodologie intelligente de reconnaissance de l'activité humaine en temps réel que l'on nomme **Intelligent Real-time Human Activity Recognition (IR-HAR)**. Un modèle de réseau de neurones convolutionnel profond à deux couches ou Convolutional Neural Network (CNN) en anglais est alors développé pour réaliser la classification des données CSI brutes générées pendant les activités humaines. Le modèle CNN est couplé avec une augmentation de données en séries temporelles pour surmonter le problème des ensembles de données de petite taille. Nos résultats montrent que notre méthodologie permet de reconnaître l'activité d'un seul utilisateur avec une précision supérieure à 90%, surpassant ainsi les approches les plus récentes.

Keywords— Reconnaissance de l'activité humaine, applications de santé à domicile, information d'état du canal, Deep learning, Convolutional neural network, augmentation de données

1 Introduction

La reconnaissance des activités humaines est un vaste domaine de recherche qui étudie le problème de l'identification et de la prédiction des comportements d'un ou de plusieurs utilisateurs. Ces dernières années, ce domaine de recherche a attiré l'attention des chercheurs. Cette attention particulière s'explique par sa forte contribution appli-

cative dans de nombreux domaines d'études tels que la santé, la robotique et l'Internet des objets [1]. Dans notre travail, nous nous intéressons aux applications liées au domaine de santé et spécifiquement celui des soins à domicile.

Dans ce travail, nous nous basons sur les technologies sans fil sans dispositif porté. Ce domaine de recherche se concentre sur la surveillance des activités humaines à l'aide de la technologie de

communication sans fil Wi-Fi. Nous utilisons cette technologie pour l'omniprésence des routeurs Wi-Fi ainsi que leur connectivité à longue portée. Par rapport aux technologies des caméras et des capteurs, ces dispositifs sont bon marché, peu gourmands en énergie, invariants à la lumière, faciles à mettre en œuvre et posent moins de problèmes de confidentialité [2, 1, 3].

En tant qu'ondes électromagnétiques, les signaux Wi-Fi sont sujets à l'absorption, à la réflexion, à la diffusion et aux interférences. Par conséquent, lorsque ces signaux rencontrent des obstacles et des objets, ils sont réfléchis et diffusés, ce qui crée un canal Wi-Fi avec des signaux superposés à trajets multiples. Dans l'environnement, ces obstacles peuvent être des objets statiques tels que le sol ou des objets mobiles tels que les corps humains. En analysant les effets de ces corps en mouvement sur les signaux environnants, nous pouvons reconnaître les activités exécutées. La CSI est utilisée pour observer et analyser ces effets [3, 4]. La collecte des données CSI prend place du côté du récepteur. Pour donner au système du récepteur la capacité d'analyser les données CSI brutes, il est nécessaire d'utiliser des algorithmes intelligents, tels que ceux basés sur les Machines Learning (ML) ou le Deep Learning (DL) [5, 6]. Dans la littérature, les auteurs de [7] proposent un modèle 2D-CNN lourd qui distingue sept activités réalisées dans des environnements intérieurs. Quant aux auteurs de [8], ils ont mis au point un modèle CNN Bi-directional Long Short-Term Memory (CNN-BiLSTM) qui effectue une classification multi-activités.

La solution IR-HAR proposée consiste en un CNN profond pour une tâche de classification multi-activités. Cette tâche consiste à distinguer sept activités réalisées par un seul utilisateur dans des environnements intérieurs. Nous considérons le CNN comme un candidat prometteur pour permettre la reconnaissance des données CSI brutes représentant la variation des canaux Wi-Fi induite

par les activités humaines. En outre, cet article utilise l'augmentation de données de séries temporelles pour surmonter le problème des ensembles de données de petite taille et améliorer les performances du modèle. L'approche proposée est testée sur un ensemble de données réelles accessibles au public provenant de [7], conduisant à une précision de classification de plus de 90%.

Le reste du document est organisé comme suit : La section 2 décrit le principe de la reconnaissance de l'activité humaine basée sur les données CSI. La section 3 présente la méthodologie. La section 4 illustre les performances de l'approche, tandis que la section 5 conclut le document.

2 Principe de l'HAR basée sur les données CSI

La CSI décrit la propagation du signal Wi-Fi de l'émetteur au récepteur. Elle représente l'effet combiné de la réflexion, de la diffusion, de la variation et de la diminution de la puissance en fonction de la distance. Il peut y avoir plusieurs antennes au niveau de l'émetteur et du récepteur, ce qui donne lieu à plusieurs liaisons Wi-Fi. Pour un signal Wi-Fi émis, la CSI est mesurée au niveau du récepteur pour chaque liaison, et l'on considère qu'il y a une variation lorsque l'amplitude et la phase du signal varient de manière significative dans le temps ou dans l'espace [4].

Pour permettre à plusieurs nœuds de partager simultanément un seul canal, la méthode de multiplexage par répartition orthogonale de la fréquence (OFDM) a été mise au point. Cette méthode divise les canaux en de nombreux sous canaux porteurs qui se chevauchent. En conséquence, la CSI est mesurée pour chaque sous canal porteur, pour chaque liaison entre l'émetteur et le récepteur. Il est représenté par un nombre complexe donné par l'équation suivante [9] :

$$csi_{s,l}(t) = a_{s,l}(t) + i b_{s,l}(t) \quad (1)$$

où $s \in S$ est l'indice de la sous-porteuse, $l \in L$ est l'indice de la liaison, $a_{s,l}(t)$ est la partie réelle de la CSI mesurée à l'instant t pour une sous-porteuse s pour une liaison l , et $b_{s,l}(t)$ sa partie imaginaire. L'activité dans le champ Wi-Fi a un impact plus important sur l'amplitude de la CSI que sur la phase. Dans ce qui suit, les amplitudes des signaux CSI, données par $x_{s,l}(t) = \sqrt{a_{s,l}(t)^2 + b_{s,l}(t)^2}$, sont utilisées comme données d'entrée pour le HAR.

3 Méthodologie proposée

3.1 Présentation générale de la méthode

La méthode IR-HAR se compose de trois phases principales : la détection des données d'activité, le cadrage des données et la reconnaissance des activités, telle qu'illustrée dans la Figure 1. Dans la première phase, l'émetteur diffuse des signaux Wi-Fi et le récepteur recueille en temps réel les données d'amplitude CSI $x_{s,l}(t)$, $s \in S$, $l \in L$. Dans la deuxième phase, la méthodologie regroupe ces points de données dans des trames d'une durée fixe $\delta t = 1s$, appelées trames d'activité. En outre, chaque trame d'activité collectée suit certaines étapes pour se conformer aux spécifications d'entrée de la phase suivante. Par exemple, les trames sont sous-échantillonnées ou sur-échantillonnées, de manière à avoir les dimensions de l'entrée de la phase suivante, si nécessaire. Enfin, dans la troisième phase, un classifieur entraîné classe les trames d'activité dans l'une des sept activités journalières considérées, à savoir se pencher, tomber, s'allonger, courir, s'asseoir, se lever et marcher. Les ensembles de données utilisés dans notre étude sont présentés ci-dessous, suivis du prétraitement des données.

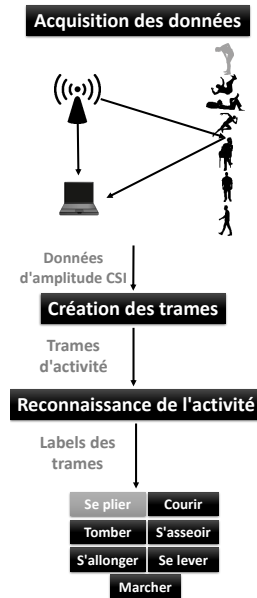


FIGURE 1 – Présentation de la méthode IR-HAR.

3.2 Base de données CSI & Prétraitement

Nous utilisons la base de données d'activité de [7], qui est accessible au public. Dans cet ensemble de données, les auteurs ont utilisé un émetteur et un récepteur dotés d'un extracteur CSI Nexmon [10], avec une seule liaison et 52 sous canaux porteurs. Dans un environnement expérimental intérieur, le récepteur a recueilli les données d'amplitude CSI pour les sept activités journalières énumérées ci-dessus, avec un taux d'échantillonnage de 200 échantillons par seconde. Trois utilisateurs d'âges différents ont effectué chaque activité 20 fois. Ils ont exporté les données de séries temporelles collectées pour chaque essai d'activité dans un fichier CSV. Le nombre total de fichiers est donc de 420 ce qui est équivalent à 60 par activité. Par conséquent, chaque fichier d'activité comporte 52 colonnes, pour les 52 sous-canaux porteurs, et 600 à 1100 lignes, en fonction de la durée de l'activité. Les données sont ensuite divisées en deux sous-ensembles : Apprentissage-

Test, avec un pourcentage de division de 85%-15% (Apprentissage : 85%, Test : 15%), ce qui conduit à 357 fichiers pour la formation et 63 pour le test. Ensuite, nous avons découpé les fichiers en des trames d'activité pour les sous-ensembles d'apprentissage et de test. Dans notre travail, nous avons fixé la taille des trames à 200, ce qui correspond au taux d'échantillonnage. Ainsi, chaque trame représente une instance de durée 1 s de l'activité enregistrée. Il est à noter que les trames se chevauchent avec un chevauchement de 160 points. Par conséquent, le nombre total de trames de taille 200×52 est de 5,681 pour le sous-ensemble d'apprentissage et de 994 pour le sous-ensemble de test. En appliquant ces mesures, seules les trames d'activité du sous-ensemble d'apprentissage sont utilisées pendant la préparation du modèle.

En observant le sous-ensemble de test obtenu, nous remarquons un léger déséquilibre entre les classes. Afin de surmonter ce problème, nous rééquilibrions l'ensemble de données de test en éliminant les trames supplémentaires pour chaque classe afin d'atteindre le plus petit nombre de trames (128 trames) de la classe minoritaire (assis, classe 4) [11]. Nous obtenons ainsi un sous-ensemble de test pré-traité équilibré, apte à fournir une évaluation objective. Cette manière de tester consiste en une évaluation en aveugle du monde réel puisqu'elle reflète une impression réelle de la performance du modèle sur des données non vues. Parallèlement, nous procédons au prétraitement du sous-ensemble d'apprentissage avant d'entraîner le modèle. Tout d'abord, ce sous-ensemble doit être divisé en ensembles d'apprentissage et de validation. Cependant, avec un petit ensemble de données comme celui de [7], recourir à la division à nouveau des données n'est pas pertinent [6]. Pour résoudre ce problème, nous proposons d'appliquer une augmentation de données (DA). Cette étape est détaillée dans la section 3.3. Après l'augmentation de données, la division apprentissage-validation est effectuée. Pour ce faire, nous me-

nons deux expériences avec différentes procédures de fractionnement. Dans la première expérience, nous utilisons un pourcentage de division 85%-15% (apprentissage : 85%, validation : 15%). Cette procédure divise aléatoirement les cadres en deux sous-ensembles. Dans la deuxième expérience, nous utilisons la procédure de validation croisée k-fold (CV) avec $k=5$ folds.

3.3 Augmentation de données (DA)

Les ensembles des données réelles souffrent principalement du problème de petite taille. Dans des tâches complexes, comme notre tâche de classification multi-activités, où les techniques de Deep Learning (DL) sont nécessaires, un grand ensemble de données d'apprentissage est requis. Adapter un modèle sur un sous-ensemble d'apprentissage avec un nombre insuffisant de trames entraînera un sur-apprentissage du modèle, ce qui entraînera une faible performance (space) [5]. Pour résoudre ce problème, nous proposons d'utiliser des techniques d'augmentation de séries temporelles pour accroître la quantité de trame [12]. Séparément, sur les trames de chaque classe (activité), nous appliquons quatre techniques d'augmentation de données basées sur des transformations aléatoires pour générer des trames supplémentaires. Ces techniques sont le Jittering, le Scaling, le Slicing et le Window Warping [12]. Le Jittering ajoute aux trames un bruit gaussien avec une moyenne (μ) = 0 et un écart-type (σ) = 0.03. Le Scaling multiplie les trames par un paramètre de mise en échelle aléatoire issu d'une distribution gaussienne avec une moyenne (μ) = 1 et un écart-type de (σ) = 0.02. Le Slicing, et plus précisément le Window Slicing, recadre les trames à l'aide d'une fenêtre de taille fixée à $0.9 \times frame_length$. Ensuite, nous interpolons les trames générées pour revenir à la longueur de trame d'origine $frame_length = 200$ échantillons. En ce qui concerne le Window War-

ping, il sélectionne une fenêtre aléatoire de taille fixée à $0.1 \times frame_length$ dans les trames et déforme cette fenêtre par une valeur de multiplication. De plus, sur les trames des 7 classes (activités), nous appliquons une technique d'augmentation de données basée sur le mélange de motifs pour générer des trames synthétiques supplémentaires. Cette technique est le Random Guided Warping (RGW) [12]. En appliquant ces techniques sur l'ensemble de données, le nombre total de trames du sous-ensemble d'apprentissage passe de 5,681 à 52,360. En appliquant les techniques d'augmentation de données, nous étendons la frontière de décision de notre modèle et atténuons le problème de sur-apprentissage. Par conséquent, nous améliorons la capacité de généralisation de ce modèle sur des données non vues, ce qui améliore à son tour ses performances globales.

3.4 Classificateur d'activités CNN

Le classifieur proposé est un CNN qui prend la trame d'activité CSI en entrée et produit l'une des sept activités en sortie. Pour créer ce classifieur, nous avons utilisé l'API Keras qui offre une large gamme de couches intégrées. Le réseau de notre classificateur est composé de trois blocs de convolution (Conv), d'un bloc entièrement connecté ou encore appelé en anglais Fully Connected (FC) et d'un dernier bloc (output). Les deux premiers blocs Conv se compose des couches suivantes : Conv2D, LeakyReLU, MaxPooling2D et BatchNormalization de Keras. LeakyReLU est la fonction d'activation appliquée avec une valeur de pente par défaut fixée à 0.3. La couche Conv2D a 32 filtres de taille 3×3 . Quant à la couche MaxPooling2D, la taille de la fenêtre de pooling est fixée à 3×3 et le padding zéro est réglé sur "same", ce qui entraîne un padding uniforme à gauche/droite ou haut/bas de la sortie convolutionnelle. Pour chaque étape de pooling, la fenêtre se déplace verticalement et horizontalement de 2. En utilisant cette

couche, l'entrée est sous-échantillonnée lorsqu'elle traverse chaque bloc. Enfin, la couche BatchNormalization normalise les activations. Ces dernières sont présentées au bloc suivant. L'utilisation d'une telle couche permet au modèle de converger plus rapidement, accélérant ainsi l'apprentissage. Dans le 3^{ème} bloc Conv, nous changeons quelques paramètres. La taille du filtre et la taille de la fenêtre de pooling sont fixées à 2×2 . Le bloc FC est composé de couches Flatten, Dense et Dropout de Keras. La couche Flatten redimensionne la sortie du 3^{ème} bloc (Conv) en une seule colonne (tableau 1D). Ce vecteur Flatten représente l'entrée de la couche Dense entièrement connectée avec 64 neurones et une fonction d'activation ReLU. Pour éviter le sur-apprentissage, nous avons utilisé une couche Dropout avec un taux (probabilité de dropout) fixé à 30% [5, 6]. Enfin, le dernier bloc de sortie se compose d'une couche Dense avec 7 neurones et une fonction d'activation softmax. Le nombre de neurones est équivalent au nombre d'activités que nous souhaitons prédire. Ici, softmax crée une distribution de probabilité pour chacune de ces 7 neurones (activités possibles). Pour effectuer une prédiction, nous prenons le score avec la valeur la plus élevée et le relient à l'activité correspondante.

4 Résultats expérimentaux

Cette section illustre les performances de l'approche proposée en utilisant le jeu de données réel décrit dans la section 3.2. Tout d'abord, notre classifieur entraîné est évalué sans puis avec l'augmentation de données. Ensuite, nous comparons notre approche aux autres approches de la littérature. Dans la suite, quatre métriques d'évaluation sont prises en compte : la précision (accuracy), qui est le pourcentage des instances de test correctement classées ; le score F1, qui est la moyenne harmonique pondérée de la précision et du rappel (recall) ; la perte (loss), qui représente la somme des erreurs ; et le temps de réponse, c'est-à-dire la durée

TABLE 1 – Performance sans DA

	1 ^{ère} Expérience				2 ^{ème} Expérience					Moyenne
	S_0	S_1	S_2	S_3	F_1	F_2	F_3	F_4	F_5	
Précision (%)	89.6	91.8	88.7	88.9	90.4	88.0	89.1	89.7	89.3	89.5
Score F1 (%)	89.4	91.7	88.6	88.8	90.2	87.8	88.9	89.6	89.3	89.3
Loss	0.620	0.612	0.526	0.558	0.580	0.666	0.562	0.643	0.517	0.587
Temps (s)	0.314	0.299	0.227	0.286	0.329	0.240	0.258	0.693	0.488	0.348

TABLE 2 – Performance avec DA

	1 ^{ère} Expérience				2 ^{ème} Expérience					Moyenne
	S_0	S_1	S_2	S_3	F_1	F_2	F_3	F_4	F_5	
Précision (%)	92.7	91.8	93.6	92.8	92.1	93.7	93.7	92.5	93.7	92.9
Score F1 (%)	92.7	91.8	93.6	92.8	92.1	93.7	93.7	92.5	93.7	92.9
Loss	0.385	0.420	0.442	0.364	0.353	0.297	0.412	0.414	0.410	0.388
Temps (s)	0.306	0.224	0.200	0.200	0.275	0.218	0.212	0.206	0.225	0.229

d'exécution de notre classificateur d'activité.

4.1 Évaluation en situation réelle de IR-HAR

Nous avons mené deux expériences avec les différentes procédures de division de l'ensemble d'apprentissage entre les sous-ensembles Apprentissage et Validation, comme décrit à la fin de la section 3.2. Le tableau 1 compare les résultats des tests sur le sous-ensemble de test, où la 1^{ère} expérience correspond à la division holdout répétée 4 fois, et la 2^{ème} expérience avec la division CV avec des résultats pour chaque fold. Dans ce tableau, les précisions et les scores F1 des tests montrent de bons résultats de classification, avec une bonne stabilité. Les choix de division des sous-ensembles d'apprentissage et de validation ont un léger impact sur les performances de classification de nos classificateurs. De plus, leur temps de réponse montre une vitesse stable et rapide. Par

conséquent, bien que nous n'ayons pas eu suffisamment de données d'apprentissage, nos modèles sont restés remarquablement efficaces.

Dans la suite, des expériences sont menées avec l'augmentation des données pour étudier l'impact de l'augmentation de la taille de l'ensemble d'entraînement sur la performance des modèles. Le tableau 2 illustre les résultats des tests sur les données non vues, également avec les deux divisions du sous-ensemble d'apprentissage. Dans ce tableau, les précisions et les scores F1 des tests montrent une meilleure performance de classification. Comparées aux expériences précédentes, ces précisions et ces scores se sont améliorés en moyenne de 3,4% et 3,6%, respectivement. Ainsi, augmenter la taille de l'ensemble d'apprentissage en appliquant les techniques DA a amélioré la performance de classification et la capacité de généralisation de notre classificateurs. De plus, leur temps de réponse montre une vitesse stable et rapide. Par conséquent, notre approche a prouvé son

TABLE 3 – Performance pour IR-HAR comparé aux méthodes de la littérature

	1D-CNN	2D-CNN	LSTM	BLSTM	IR-HAR
Précision (%)	51.6	85.0	74.5	81.7	93.7
Score F1 (%)	50.6	84.8	74.3	80.5	93.7
Loss	1.466	3.285	0.669	0.575	0.412
Time (s)	0.260	0.290	0.318	1.026	0.212

efficacité à améliorer les performances des modèles. La précision de test la plus élevée est égale à 93,7%, obtenue avec l’augmentation des données et la technique de validation croisée appliquée pour entraîner et valider le modèle.

4.2 Comparaison d’IR-HAR avec l’état de l’art

Deux erreurs courantes existent dans la littérature. Ces dernières influencent la fiabilité des résultats d’évaluation. La première erreur est commise par les chercheurs lors du prétraitement de leur jeu de données CSI. Ils exécutent le processus de cadrage sur le jeu de données avant de le diviser en deux sous-ensembles d’apprentissage et de test. Ce comportement entraîne des sous-ensembles d’apprentissage et de test contenant des trames d’activité qui partagent des échantillons de données communs. Certains ne s’arrêtent pas à cette erreur, mais utilisent les mêmes trames d’activité pour valider et tester leurs modèles. Suite à ces comportements incorrects, nous pouvons dire que les résultats d’évaluation finaux ne reflètent pas une impression réelle de la performance du modèle.

Afin de comparer les performances de notre approche à celles de la littérature, nous avons évalué les différents modèles DL proposés par les auteurs de [7] sur les données de test non vues, prises de leur jeu de données. Le tableau 3 compare les résultats de test de leurs modèles à ceux des

nôtres. Sachant que le classificateur 2D-CNN des auteurs de [7] est composé de deux blocs Conv et d’un bloc FC (*profondeur* = 3), ce classificateur a un total de 6,598,023 paramètres entraînaibles, ce qui le rend coûteux en calcul. Ainsi, pour notre classificateur proposé (entraîné sur le jeu de données du papier [7]), nous avons réduit le nombre de paramètres à 309,511 et sommes allés plus profondément avec seulement un bloc Conv supplémentaire (*profondeur* = 4). Notre classificateur peut être intégré dans des dispositifs ayant des ressources et une puissance de calcul limitées. Se référant au tableau 3, la précision des tests et le score F1 de notre classificateur sont nettement supérieurs à ceux des classificateurs de pointe. Par conséquent, notre modèle a une meilleure performance de classification et une capacité de généralisation sur les données non vues. Il a surpassé les approches de pointe et a atteint l’objectif d’obtenir un classifieur d’activités efficace. De plus, la réduction du coût de calcul et de la complexité n’a pas affecté la performance.

5 Conclusion

Dans cet article, nous avons proposé une méthodologie de reconnaissance intelligente en temps réel des activités humaines. Le principal composant de l’IR-HAR est un modèle CNN léger. Nous avons entraîné différentes instances de ce modèle sur le jeu de données pré-traité de [7]. Comme ce jeu de données rencontre le

problème de petite taille de données, nous avons employé des techniques d'augmentation de données de séries temporelles pour aider certaines instances à surmonter les difficultés. Pour examiner l'influence de ces techniques sur la performance de notre modèle, nous avons préparé un sous-ensemble de test avec des données non vues à l'apprentissage. Ce sous-ensemble a également permis de comparer notre modèle aux approches de la littérature. Les résultats expérimentaux montrent que l'augmentation des données a amélioré la performance de classification de notre modèle et sa capacité de généralisation sur des données non vues. De plus, ce modèle a surpassé les approches de la littérature. De nombreux jeux de données d'activité CSI étiquetés dans la littérature sont sujets au problème de données de petite taille. Dans les travaux futurs, nous avons l'intention de développer des réseaux de neurones génératifs profonds pour générer des trames d'activité synthétiques supplémentaires.

Remerciements : Les auteurs aimeraient remercier les auteurs de [7] pour avoir partagé leur ensemble de données d'activités. Ce travail est soutenu par l'Université de Technologie de Troyes et l'Université Libanaise.

Références

- [1] Z. Wang, B. Guo, Z. Yu, and X. Zhou, "Wi-fi csi-based behavior recognition : From signals and actions to activities," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 109–115, 2018.
- [2] B. Tan, Q. Chen, K. Chetty, K. Woodbridge, W. Li, and R. Piechocki, "Exploiting wifi channel state information for residential healthcare informatics," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 130–137, 2018.
- [3] Y. Zou, W. Liu, K. Wu, and L. M. Ni, "Wi-fi radar : Recognizing human behavior with commodity wi-fi," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 105–111, 2017.
- [4] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information : A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–36, 2019.
- [5] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning : Concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, no. 1, pp. 1–74, 2021.
- [6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [7] P. Fard Moshiri, R. Shahbazian, M. Nabati, and S. A. Ghorashi, "A csi-based human activity recognition using deep learning," *Sensors*, vol. 21, no. 21, p. 7225, 2021.
- [8] Z. Shi, J. A. Zhang, R. Y. Xu, and Q. Cheng, "Environment-robust device-free human activity recognition with channel-state-information enhancement and one-shot learning," *IEEE Transactions on Mobile Computing*, vol. 21, no. 2, pp. 540–554, 2020.
- [9] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity wifi," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, MobiCom '15*, (New York, NY, USA), p. 53–64, ACM, 2015.
- [10] F. Gringoli, M. Schulz, J. Link, and M. Hollick, "Free your csi : A channel state information extraction platform for modern wi-fi chipsets," in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, pp. 21–28, 2019.
- [11] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*, vol. 10. Springer, 2018.
- [12] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *Plos one*, vol. 16, no. 7, p. e0254841, 2021.