
Damien Bouchabou, Sao Mai Nguyen, Christophe Lohr,
Ioannis Kanellos, Benoit LeDuc

Reconnaissance d'activités de la vie quotidienne au moyen de capteurs domotiques et d'apprentissage profond : lorsque syntaxe, sémantique et contexte se rencontrent

Volume 4, nº 1 (2023), p. 129-156.

https://doi.org/10.5802/roia.53



*La Revue Ouverte d'Intelligence Artificielle est membre du Centre Mersenne pour l'édition scientifique ouverte*
*www.centre-mersenne.org*
*e-ISSN : 2967-9672*

**Revue Ouverte d'Intelligence Artificielle**
Volume 4, n° 1, 2023, 129-156

# Reconnaissance d'activités de la vie quotidienne au moyen de capteurs domotiques et d'apprentissage profond : lorsque syntaxe, sémantique et contexte se rencontrent

**Damien Bouchabou[a], Sao Mai Nguyen[a], Christophe Lohr[a], Ioannis Kanellos[a], Benoit LeDuc[b]**

[a] IMT Atlantique, Dept. Informatique, 655 avenue du technopole, 29280 Plouzané, France
*E-mail :* damien.bouchabou@gmail.com, dbouchabou@deltadore.com, nguyensmai@gmail.com, christophe.lohr@imt-atlantique.fr, ioannis.kanellos@imt-atlantique.fr

[b] Delta Dore, Dept. Informatique, 655 avenue du technopole, 35270 Bonnemain, France
*E-mail :* bleduc@deltadore.com.

Résumé. — Dans la thématique grandissante de la reconnaissance d'activités de la vie quotidienne au sein de maisons intelligentes, les réseaux de neurones basées sur les Long Short Term Memory (LSTM) ont démontré leur efficacité. En étudiant l'ordre des activations des capteurs et leurs dépendances temporelles, on traduit les actions humaines comme une suite d'événements dans le temps plus ou moins corrélés. Cependant, l'activité humaine n'est pas une suite d'actions dénuées de sens ni de contexte. Nous proposons d'utiliser et de comparer deux méthodes provenant du traitement du langage naturel pour, justement, prendre en compte la sémantique et le contexte des capteurs afin d'améliorer les algorithmes dans les tâches de classification de séquences d'activités : Word2Vec, un embedding de sémantique statique, et ELMo, un embedding contextuel. Les résultats, sur des datasets réels de maisons intelligentes, indiquent que cette approche fournit des informations utiles, comme une carte de l'organisation des capteurs, et réduit par ailleurs la confusion entre les classes d'activités quotidiennes. Elle permet d'obtenir de meilleures performances sur des datasets contenant des activités concurrentes avec plusieurs résidents ou des animaux domestiques. Nos tests montrent également que les embeddings peuvent être pré-entraînés sur des datasets différents du jeu de données cible, permettant ainsi un apprentissage par transfert. Nous démontrons ainsi que la prise en compte du contexte et de la sémantique des capteurs augmente les performances de classification des algorithmes et permet l'apprentissage par transfert.

Mots-clés. — Reconnaissance d'activités quotidiennes, maisons intelligentes, modèles sémantiques, modèles syntaxiques, réseaux de neurones, apprentissage profond.

## 1. Introduction

### 1.1. Contexte

Devenue un des grands défis mondiaux de santé publique de nos jours et, en France du moins, enjeu majeur d'une cinquième tranche de la Sécurité Sociale, la question de la dépendance de personnes cherche des solutions techniques en puisant avec espoir dans le paradigme de l'Intelligence Artificiel (IA). La perte de la dépendance n'adresse pas seulement le vieillissement de la population [13], mais résume toute perte d'autonomie, transitoire ou permanente, qui établit un lien avec l'érosion des capacités d'être au monde (physiques, fonctionnelles, cognitives, sociales, affectives...) chez une personne [43]. Le plus souvent, elle se traduit par une difficulté de réaliser des tâches simples, comme cuisiner, prendre des médicaments, aller aux toilettes, etc., tâches qui peuvent vite rendre une vie quotidienne pénible, voire coûteuse, tant sur le plan d'une économie personnelle ou familiale que, plus avant, sur celui des institutions médicales nationales. Plusieurs indicateurs, plus ou moins fiables, que l'on trouve sur Internet, chiffrent à plusieurs centaines d'heures par an les occupations domestiques nécessaires et récurrentes. Fournir des services automatisés afin de permettre aux personnes dépendantes de vivre de manière aussi autonome, confortable et saine que possible dans leur propre maison, tout en limitant les risques liés à leurs activités in domo, a déjà ouvert un champ économique inédit [9, 41] et stimule désormais tant les sciences que les technologies, notamment numériques.

### 1.2. La maison intelligente

La maison de demain se conçoit dès aujourd'hui comme une « maison intelligente ». Elle ne doit pas seulement être un lieu de résidence, mais une authentique plateforme de services et d'expériences utilisateur, fournis par un écosystème de technologies interconnectées. En effet, grâce aux progrès de l'Internet des Objets (IoT), la maison intelligente tente dès aujourd'hui d'explorer une palette grandissante de services à domicile, tels que la gestion de la sécurité, la gestion de l'énergie, l'assistance dans les tâches quotidiennes ou le suivi des soins de santé.

Une maison intelligente est une maison capable de « percevoir » des faits se produisant à son intérieur, grâce à des capteurs de nature variable, installés à des positions stratégiques et d'agir sur un environnement physique via des actionneurs. Elle est équipée de capteurs domotiques et d'appareils contrôlables. Un tel écosystème technologique est interconnecté au moyen de protocoles de communication spécifiques [22].

### 1.3. La reconnaissance d'activités humaines dans une maison intelligente

Pour pouvoir fournir des services adéquats, une maison intelligente doit « comprendre » et « interpréter » les routines de vie de ses résidents. Il devient ainsi nécessaire de développer des techniques de reconnaissance de l'activité humaine (RAH) ad hoc, afin de suivre et d'analyser le comportement d'une voire de plusieurs personnes pour en déduire leur activité.

Les différentes approches de RAH se départagent, grosso modo, en deux catégories principales [11] : les approches basées sur la vidéo et la reconnaissance d'images, et les approches basées sur des capteurs ambiants. Compte tenu des problèmes d'intrusion au sein de la vie privée des résidents et les réticences, tant des individus que du législateur, afférentes à une installation de caméras dans un espace réputé privé, les systèmes de RAH à base de capteurs ambiants semblent encore dominer dans le domaine de la recherche sur les maisons intelligentes [6]. En effet, les capteurs ambiants sont généralement considérés moins intrusifs et, donc, sont mieux acceptés [2, 43]. La RAH dans les maisons intelligentes consiste, ainsi, à traduire des traces de capteurs ambiants en Activités de la Vie Quotidienne (AVQs). Simple dans sa formulation, la reconnaissance des AVQs reste néanmoins une tâche proposant de nombreux défis, tant matériels qu'algorithmiques, tant de conception que de généralisation.

En effet, il faut tout d'abord tenir compte de la structure, de la topologie et des équipements des maisons. Ensuite, il faut modéliser les habitudes de vie des résidents qui peuvent varier d'une maison à une autre et d'un résident à l'autre. Plus avant, le nombre de résidents vivant en même temps surajoute une dernière, et pas la moindre, difficulté. Plus il y a de résidents, plus les séquences d'activation des capteurs qui correspondent à des activités différentes se trouvent entrelacées. Par exemple, si un résident cuisine tandis qu'un autre prend sa douche au même moment, les changements d'états des capteurs, qui correspondent à chacune des activités par l'un et par l'autre, ont une tendance à se mélanger. Il convient ultimement de prendre en compte le contexte lié à la pièce, aux objets, aux dispositifs de la maison utilisés, voire au type d'interactions durant les AVQs. Par exemple, ouvrir une porte pour entrer dans une pièce ou ouvrir la porte d'entrée pour quitter la maison utilise le même type de capteur, sur un même type d'objet, avec la même interaction, mais transcrit une activité différente.

Un autre défi s'impose, ainsi, de par le choix même des capteurs. Il est clair que les capteurs ambiants, contrairement aux caméras, ne fournissent que peu d'informations. Un capteur de mouvement, comme son nom l'indique, ne donnera que l'information binaire de l'existence d'un mouvement ou non dans l'espace qu'il surveille. L'activation de chaque capteur, indépendamment, ne donne donc que peu d'informations sur l'activité en cours en tant que telle. Par exemple, l'activation du capteur de mouvement dans la cuisine peut indiquer des activités telles que « cuisiner », « faire la vaisselle », « faire le ménage », etc. Ainsi, l'information offerte par ce capteur ne peut être utilisée qu'en conjonction avec l'activation d'autres capteurs. Autrement dit, par le contexte. C'est, effectivement, le contexte dans lequel un capteur s'active qui donne l'information. Par ailleurs, ces capteurs sont déclenchés par des événements liés à l'action du résident. Ce qui génère des séries temporelles non régulièrement échantillonnées et temporellement éparses, ce qui rend difficile l'interprétation temporelle et demande, in fine, un traitement différent des séries temporelles classiques. L'activation de deux capteurs consécutifs à une seconde ou à une heure d'intervalle peut transmettre une information fort différente.

Enfin, un dernier problème, et pas le moindre, est lié à la labellisation des activités des activations des capteurs. Afin de fournir des environnements de test pour les algorithmes de RAH, des datasets publiques et labellisés sont disponibles dans la littérature [12]. Cependant, au sein de ces datasets, une grande partie des données d'activation des capteurs ne sont pas annotées. Certaines de ces activations peuvent, ainsi, n'appartenir à aucune activité en particulier ou être liées à des moments de transition entre activités. L'une des raisons d'un tel état des choses est que les datasets sont construits avec un ensemble prédéfini de classes d'activités. Il existe, donc, des activités réalisées par le/les résidents qui restent inconnues. Une autre raison est la difficulté de traduire aisément en activité les données fournies par les capteurs. Le choix du début, de la fin ou tout simplement du label d'une activité est un défi à part entière et relève de l'appréciation de la personne en charge de l'annotation.

Au vu de ces problèmes, problèmes d'imprécision, de robustesse, de normalisation, de fiabilité, etc, des données somme toute, les algorithmes de RAH doivent continuellement relever des défis en termes de reconnaissance de formes et d'analyses de séquences temporelles [6].

## 2. Travaux relatifs

Les défis de la RAH dans les maisons intelligentes peuvent être résumés par quatre problèmes, tous majeurs :

- l'adaptabilité à l'environnement que constitue la maison, mais aussi à la multitude de manières de vivre dans cet environnement ;
- la reconnaissance de motifs dans des séquences qui peuvent être bruitées ou très ressemblantes ;
- l'interprétation de séquences ou d'évènements temporels, potentiellement ordonnés et corrélés ;
- l'extraction et l'interprétation des informations fournies par les capteurs domotiques.

Les algorithmes issus de techniques d'apprentissage automatique semblent actuellement comme les directions de recherche les plus prometteuses pour relever ces défis.

### 2.1. Adaptabilité

Plusieurs modèles et techniques basés sur des méthodes d'apprentissage automatique dit « Shallow Learning » ont été explorés dans cet objectif [36]. Cet ensemble de modèles et de techniques peut être divisé en deux types : (1) les algorithmes exploitant une représentation spatio-temporelle, avec les Naive Bayes, Dynamic Bayesian Networks, Hidden Markov Models, et (2) les algorithmes basés sur la classification de caractéristiques, par les Decision Trees, les Support Vector Machines ou les Conditional Random Fields.

Ces approches sont robustes, faciles à mettre en œuvre et nécessitent une faible puissance de calcul. Cependant, elles utilisent généralement des méthodes d'extraction

de caractéristiques manuelles. Une telle tâche devient vite fastidieuse et doit se refaire à chaque fois qu'il y a un changement. Cela nécessite une expertise et ne permet pas à ces méthodes de s'adapter lorsque la topologie devient différente ou que les habitudes des résidents changent. Fatalement, ces méthodes fonctionnent uniquement dans l'environnement pour lequel elles ont été conçues et ne sont pas extensibles en raison de leur manque de généralité.

L'extraction automatique de caractéristiques est l'un des défis relevés par les approches de Deep Learning (DL). Ces méthodes apprennent directement à partir des données brutes de manière hiérarchique et découvrent, seules, des caractéristiques de haut niveau. Ces mêmes algorithmes peuvent non seulement extraire des caractéristiques de manière automatique, mais aussi réaliser la tâche de classification. Récemment, divers algorithmes de DL ont été appliqués à la RAH [6].

### 2.2. Reconnaissance de forme

La RAH peut être ramené à un problème de reconnaissance et de classification de formes. En effet, à partir de caractéristiques et de critères, une méthode identifie des motifs afin de leur attribuer une catégorie.

Les réseaux neuronaux convolutifs (CNN) ont fait leurs preuves en termes de reconnaissance et de classification de formes, notamment dans les domaines de la vidéo, de l'image et du son. Ils présentent trois avantages pour la RAH : (1) ils peuvent capturer les dépendances locales, c'est-à-dire l'importance des observations voisines corrélées à l'événement actuel ; (2) ils sont invariants à l'échelle en termes de différence de pas et de fréquence d'événements ; et enfin (3), ils peuvent apprendre une représentation hiérarchique des données.

C'est pourquoi, afin d'extraire des motifs dans les activités, [16] et [27] ont utilisé des CNN 2D en transformant les séquences d'activité de capteurs en images. Cette approche a obtenu de bons résultats de classification sur les séquences d'activité pré-segmentées. Toutefois, l'ordre temporel des activations des capteurs n'est pas pris en compte. D'autres travaux [42], utilisant une représentation sous forme d'images dans des fenêtres glissantes, ont été menés pour s'attaquer à ce problème. Cependant, cette dernière méthode n'est pas assez robuste pour traiter des ensembles de données déséquilibrées, (c'est-à-dire, lorsque certaines classes ont plus d'occurrences que d'autres), ou encore des séquences d'événements non étiquetés. De plus, cette représentation sous forme d'image est adaptée aux capteurs binaires, mais pas aux capteurs à valeur numérique, comme, par exemple, un capteur de température.

### 2.3. Séries temporelles

Les CNN 1D semblent être une solution compétitive pour les problèmes de classification de séries temporelles [15, 45]. Ils permettent d'extraire des motifs et, en même temps, d'interpréter l'aspect temporel.

Les travaux de [39] montrent que les CNNs 1D peuvent tout aussi bien être appliquées à la RAH dans les maisons intelligentes.

Les modèles CNN ont l'avantage d'être rapides à entraîner. Ils atteignent une grande précision, capturent des motifs, mais souffrent du manque d'interprétation de dépendances à long terme, et traitent mal les séquences de tailles variables. Les Long Short Term Memory (LSTM) sont une autre approche de DL axée davantage sur l'aspect temporel et les dépendances à plus ou moins long terme dans une séquence. Ils ont permis d'obtenir de bonnes performances dans le domaine de la RAH pour les maisons intelligentes, comme le rapporte [20, 40].

Une comparaison entre une approche CNN 1D et LSTM a été réalisée dans les travaux de [40]. Ils démontrent que les LSTM sont plus performants dans la tâche de classification, car ils permettent l'apprentissage d'informations temporelles à partir des données des capteurs.

Par ailleurs, différentes structures basées sur les LSTM ont été étudiées et comparées à d'autres méthodes de l'état de l'art [20]. Les approches LSTM et, en particulier, les LSTM bidirectionnels, obtiennent les meilleurs résultats, en raison de leur capacité à exploiter leurs mémoires internes pour capturer les dépendances à long terme dans des séquences de longueur variable. Ils modélisent mieux l'ordre et la densité des événements ; ils représentent donc mieux la macro-structure d'une activité.

Les LSTM semblent, par conséquent, être une solution viable pour améliorer de manière significative la tâche de RAH dans la maison intelligente, malgré un temps d'apprentissage plus long que les approches basées sur CNN.

### 2.4. Codage et capture de l'information

Nous avons déjà attiré l'attention sur le fait que les capteurs domotiques ou embarqués dans les objets du quotidien ne fournissent que peu d'informations. C'est le type de capteur, le type d'activation, l'ordre et le contexte composé des activations des capteurs avant et après un capteur donné, qui apportent une information plus riche et susceptible d'être mieux exploitée. Cette dernière remarque indique déjà qu'il serait nécessaire d'étudier quelque notion de sémantique, de syntaxe et de contexte lié à chaque activation.

Les approches de DL citées ci-dessus ont pour point commun la capacité d'extraire automatiquement des caractéristiques à partir des données brutes. Les LSTM prennent, en plus, en compte les dépendances à long terme et l'ordre d'apparition des activations de capteurs.

Néanmoins, il est nécessaire d'adapter le codage des activations de capteurs pour extraire plus d'informations. Pour encoder les activations de capteurs [44] traite le flux de capteurs comme plusieurs séries temporelles (une par capteur). Ils proposent d'utiliser une fenêtre temporelle et quatre manières de coder ces fenêtres. Un premier codage transforme la fenêtre en une matrice binaire où les colonnes sont les différents capteurs et les lignes leurs changements dans la fenêtre. Ils proposent une

deuxième représentation de ces fenêtres suivant un vecteur à une dimension de taille égale au nombre de capteurs déployés dans le dataset. Ce vecteur peut ensuite prendre trois formes. Premièrement, une représentation binaire où, dans les vecteurs, seuls les capteurs présents dans la fenêtre sont à l'état 1. Deuxièmement, le vecteur peut prendre une forme de suite d'entiers, où chaque entier indique, pour chaque capteur du dataset, combien de fois le capteur en question apparaît dans la fenêtre. Et troisièmement une forme de vecteur de probabilité, où chaque probabilité transcrit un ratio d'apparition du capteur en question dans la fenêtre vis-à-vis de l'ensemble des capteurs. Il semblerait, d'après les auteurs, que le codage numérique soit le plus performant. Cependant, ce type d'encodage ne permet pas de représenter des capteurs non binaires.

Un autre encodage, à partir de Fuzzy Time Windows (FTW) [25], est utilisé dans [17, 18]. Cet encodage utilise une succession de FTW de différentes tailles définies par une suite de Fibonacci par capteurs. Pour chacun des capteurs, les suites de FTW codent, sous forme de valeur numérique, les activations, à plus ou moins long terme. Chaque FTW évalue l'activation du capteur en question sur une période de temps définie par sa taille et produit un score. Ces scores sont regroupés en vecteur par capteur, puis assemblés pour former une matrice de caractéristiques où chaque ligne est un capteur et chaque colonne son score d'activation au cours du temps. C'est cette matrice qui sera utilisée par le classifieur. La force de cette représentation est de parvenir à prendre en compte dans une fenêtre des activations lointaines d'un même capteur. Malheureusement, comme la méthode précédente, elle ne permet pas de représenter des capteurs non binaires. De plus, ces deux approches, ainsi que la cohorte des méthodes de DL vues précédemment, ignorent la sémantique, autrement dit le type de capteur et le contexte dans lequel le capteur est activé.

L'analyse sémantique et la modélisation du contexte ont été, très tôt et naturellement, au centre des recherches en Traitement du Langage Naturel (TLN). Les dernières avancées dans ce domaine ont proposé différentes méthodes de pré-entraînement non supervisées de projection de mots dans un espace vectoriel, appelé embedding, pour créer des représentations de mots et des modèles de langage. Ces embeddings capturent des informations concernant la construction des mots, des phrases et des textes. Ils sont capables de capturer aussi le contexte d'un mot dans un document, des similarités sémantiques ou syntaxiques, des relations avec d'autres mots, etc.

Plusieurs structures et méthodes d'entraînement ont permis des avancées dans le domaine du TLN. Elles peuvent être classées en deux catégories. Les approches d'embedding statiques, telles que Word2vec [26], GloVe [29], FastText [4], etc. Et les approches contextualisées, telles que ELMo [30], BERT [14], GPT [32], etc. Ces modèles, créés de manière non supervisée, se sont avérés efficaces dans le cadre de transfert learning sur les tâches de traduction, de génération de texte et de classification.

Dans le cadre de la RAH, des travaux antérieurs ont déjà étudié l'utilisation de méthodes d'embedding. L'entraînement d'un modèle Word2Vec a été utilisé pour regrouper et créer une relation sémantique des habitudes d'une population [8]. L'utilisation d'un embedding public pré-entraîné sur des textes a été utilisé pour associer

un label à des activités inconnues dans le cadre de capteurs portés [23, 38]. La même approche a été étudiée pour annoter des activités inconnues et réaliser un apprentissage de type « zero-shot »[19] dans une maison intelligente [1].

Néanmoins, l'entraînement d'embedding non supervisé, tel qu'on l'utilise dans le domaine du TLN, mais pour des activations de capteurs, n'a pas été exploité. La capture du contexte et de la sémantique de l'activation des capteurs n'est pas exploitée non plus de nos jours. Or, grâce à ces méthodes de génération d'embedding, il serait possible de capturer des informations sémantiques, syntaxiques et contextuelles des activations des capteurs. Ce type d'information enrichirait les connaissances sur l'activation en question et permettrait de la distinguer en fonction des cas d'apparition. Ce qui pourrait améliorer les performances des algorithmes de classification des AVQs. De plus, la création d'un embedding pré-entraîné pourrait permettre, par transfert learning, de réaliser la tâche de RAH dans de nouvelles maisons. Ce sont ces idées qui ont motivé nos travaux.

### 2.5. Contributions

Notre contribution se résume en quatre propositions :

- mettre sur pied une méthode pour encoder et représenter les activations de capteurs dans une maison intelligente ;
- enrichir et capturer automatiquement au travers de méthodes d'embeddings d'activation de capteurs les informations fournies par ces derniers avec des informations temporelles, syntaxiques, voire sémantiques ;
- transférer des connaissances pour la RAH d'une maison à une autre grâce à l'embedding de capteurs.

## 3. Méthode

### 3.1. Vue d'ensemble

Dans nos travaux, nous avons choisi de considérer les traces des différents capteurs comme une seule série temporelle, dans laquelle les activations de capteurs se succèdent, afin de tenter de capturer les relations entre ces activations de capteur. En effet, à chaque activation de capteur est associée une action ou une interaction de l'habitant.

Notre objectif en termes de RAH est de classer des sous séquences d'activation de capteur en AVQs. Pour cela nous proposons d'adopter la structure représentée par la Figure 3.1.

Notre approche se décompose en trois parties. Premièrement, nous encodons les activations de capteurs et, par la même occasion, les séquences d'activations. Deuxièmement, nous utilisons une couche d'embedding pour représenter les activations sous forme de vecteurs de caractéristiques. Enfin, nous ajoutons un classifieur, basé sur une structure de réseau de neurones pour capturer les motifs, ainsi que les meilleures caractéristiques, afin de classer la séquence d'activation en AVQ.

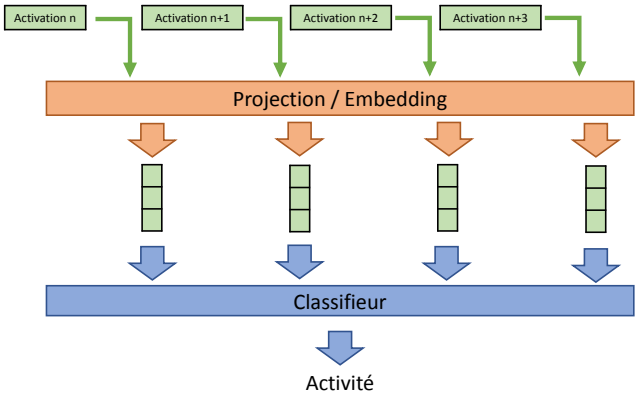



FIGURE 3.1 – Architecture globale proposée

## 3.2. ENCODAGE

Nous utilisons le paradigme du langage pour essayer de représenter et de retirer quelques informations des raisons qui sous-tendent les relations entre activations de capteurs. Par analogie, au même titre qu'une langue peut décrire des activités avec des mots, nous disons que « la maison dispose des mots pour décrire ce qui se passe ». Nous associons une activation de capteur à un mot dont dispose la maison et une séquence d'activations à une phrase décrivant une ou des activités.

Pour transformer chaque activation de capteur en mot, nous concaténons l'identifiant du capteur avec la valeur de son activation. Par exemple, si le capteur « M001 » passe à l'état « ON », cette activation devient le mot « M001ON ». Il en est de même pour les activations de capteurs utilisant des valeurs numériques tels que les capteurs de température. Par exemple, le capteur de température « T004 » qui atteint la température « 24.5 » devient l'activation « T00424.5 ». De cette manière, toutes les activations de capteur possibles sont encodées par un mot ou, plus exactement, une valeur catégorique. Cette représentation permet ainsi au modèle de pouvoir interpréter les relations entre activations de capteur dans une séquence.

Afin que les modèles puissent interpréter correctement les mots, de même que dans le domaine du TLN, nous encodons chaque mot par un entier. La valeur de cet entier est attribuée en fonction de la fréquence d'apparition du mot dans l'ensemble du dataset. Par exemple, si le mot « M001ON » apparaît le plus fréquemment dans tous le dataset, ce mot sera remplacé par l'index le plus petit. Dans ce cas, la valeur 1. L'entier 0 est une valeur particulière réservée au remplissage des séquences, car les modèles de réseau de neurones ne peuvent traiter que des séquences de même taille. Cette attribution d'index, par fréquence décroissante, permet d'attribuer un entier plus grand aux mots / événements les plus rares. Ce dernier autorise ainsi de donner plus ou moins d'importance à certains événements dans la séquence en fonction de leur rareté.

### 3.3. Approche série temporelle

Une première manière que nous proposons pour traiter la RAH dans les maisons intelligentes est de considérer la RAH comme un problème de classification de série temporelle [5]. Dans cette approche, nous considérons la sortie de la couche de d'embedding comme une série temporelle multivariée. En effet, chaque activation de capteur est transformée en vecteur et chaque séquence en une suite de vecteur. En tant que classifieur, nous avons utilisé un Fully Convolutionnal Network (FCN) [45], un classifieur de série temporelle ayant démontré de hautes performances de classification [15].

Dans nos précédents travaux [5] nous avions adopté une fenêtre glissante de taille fixe sur les séquences d'activité pour remédier au problème de taille des séquences, car les séquences d'activités sont de taille variable. L'utilisation d'une fenêtre glissante permet de découper chaque séquence en plus petites séquences limitant ainsi le padding.

Cette méthode a permis d'obtenir une accuracy et un F1-score élevés [5]. Toutefois, l'utilisation de la fenêtre glissante a fini par introduire un biais de ressemblance. Autrement dit, les jeux de test et d'entraînement étaient trop similaires.

Afin de pouvoir entraîner le FCN en utilisant les séquences d'activité complète, il est nécessaire de les tronquer ou de les compléter par un remplissage, en général fait de zéro (zero-padding). Cependant, nous avons observé que le FCN avait tendance à se focaliser sur le remplissage, l'empêchant ainsi de généraliser et donc d'obtenir de bonnes performances de classification.

Pour résoudre ce problème lié au zero-padding nous avons remplacé ce type de padding par d'autres : le symmetric padding [46] et le circle padding [35]. Au lieu d'ajouter en fin de séquence de zéro, le symetric padding répète, selon un axe de symétrie, les activations de capteurs. Le circle padding, lui, répète la séquence de valeur complète à l'infini. Ces types de paddings agissent comme de l'augmentation de données et permettent au FCN de trouver des motifs.

### 3.4. Approche sémantique et contextuelle

Les capteurs domotiques donnant peu d'informations seuls, nous proposons d'enrichir la représentation des activations de capteur par des méthodes embeddings avancées provenant du TLN. Au travers de ces méthodes, nous voulons capter quelques informations sémantiques et contextuelles utiles issues des activations des capteurs ; mais aussi, la relation syntaxique entre ces activations de capteurs.

Pour capturer une relation sémantique entre les activations des capteurs, nous proposons d'entraîner un modèle Word2Vec [26] sur les activations de capteurs. Word2Vec est une technique d'apprentissage non supervisé permettant d'apprendre des représentations continues des mots. De par sa conception, il capture la similarité des mots dans un corpus, mais également un certain « sens » du mot. Néanmoins, le principal problème de cette méthode est qu'elle fournit une représentation unique pour chaque mot, quel que soit le contexte.

Parce que Word2Vec ne peut fournir qu'une seule représentation par mots, il ne parvient pas à traiter efficacement la polysémie lexicale. Ainsi il ne pourrait par exemple pas encoder les différents sens du mot « orange », qui ne fournit pas le même sens dans la phrase « j'ai une voiture orange » ou « je bois du jus d'orange ». De même, ce modèle ne peut pas tirer parti des informations relatives au contexte dans lequel le mot a été utilisé.

C'est pourquoi au lieu d'utiliser une représentation fixe pour chaque mot, un Embedding from Language Model (ELMo) [30] examine la phrase entière avant d'assigner à chaque mot sa représentation. L'idée centrale des embeddings contextuels est de fournir plusieurs représentations pour un mot en fonction du contexte dans lequel ce mot apparaît. Nous proposons d'entraîner un modèle ELMo pour obtenir une représentation contextualisée des activations de capteurs.

### 3.5. Approche Transfer Learning

Entraîner un algorithme de RAH nécessite des données d'activité provenant de la maison cible afin que ce modèle soit adapté à l'environnement. Hélas, il est difficile d'obtenir suffisamment de données, qui plus est labellisées. Dans le domaine du TLN, des embeddings comme ELMo ou Word2Vec sont entraînés sur des datasets contenant une certaine quantité de données. Ces embeddings entraînés sont ensuite réutilisés sur d'autres datasets de taille plus restreinte, par exemple pour des tâches de classification. Les embeddings pré-entraînés permettent ainsi, par transfert learning, d'aider des algorithmes de classification dans leur tâche sur de plus petits jeux de données, grâce à leurs connaissances a priori.

Dans cette approche, nous proposons d'évaluer la capacité de transfer learning d'un embedding d'activations de capteurs pré-entraîné. L'idée que nous proposons est, justement, de pré-entraîner un embedding sur le dataset d'une maison $A$ et d'utiliser cet embedding sur le dataset d'une maison $B$.

## 4. Expériences

### 4.1. Les datasets

Les expériences ont été menées sur trois datastets de CASAS [10] : Aruba, Milan et Cairo, créés par l'université d'État de Washington. Les données collectées proviennent de vrais appartements et maisons avec de vrais habitants. Tous ces lieux de vie ont été équipés de capteurs de température et binaires tels que des capteurs de mouvement ou de portes.

Les trois datastets sélectionnés sont différents de par la structure de l'habitation et du nombre d'habitants (voir les détails dans le tableau 4.1). Aruba est un jeu de données d'une personne vivant seule dans une maison. Milan contient les activités quotidiennes d'une personne vivant avec un animal de compagnie, tandis que Cairo est un jeu de données de deux personnes vivant sous le même toit. Ces datastets contiennent plusieurs mois d'activités étiquetées et sont dit déséquilibrés, c'est-à-dire

que certaines activités sont moins représentées que d'autres. Ce déséquilibre s'explique par le fait qu'il s'agit de données provenant d'habitations réelles avec de vrais habitants, et que les activités sont liées au mode de vie de ces habitants.

TABLE 4.1 – Details des Datasets

| | Aruba | Milan | Cairo |
|---|---|---|---|
| Residents | 1 | 1+animal | 2+animal |
| Number of sensors | 39 | 33 | 27 |
| Number of activities | 12 | 16 | 13 |
| Number of days | 219 | 82 | 56 |

### 4.2. PRÉ-TRAITEMENT DES DATASTETS

Pour pré-traiter les datasets, nous nous sommes inspiré des travaux de [20] qui utilise également les datasets de Milan et Cairo. Dans leurs travaux, les activités sont regroupées sous de nouvelles étiquettes génériques, détails dans le tableau 4.2. Pour comparer notre approche à leur travail, nous avons effectué le même ré-étiquetage.

TABLE 4.2 – Regroupement des activités

| | Milan | Cairo |
|---|---|---|
| Bathing | Master Bathroom<br>Guest Bathroom | |
| Bed to toilet | Bed to toilet | Bed to toilet |
| Cook | Kitchen Activity | Lunch<br>Dinner<br>Breakfast |
| Eat | Dining Room Activity | |
| Enter home | | |
| Leave home | Leave home | Leave Home |
| Personal hygiene | | |
| Relax | Read<br>Watch Tv | |
| Sleep | Sleep | R1 sleep<br>R2 sleep |
| Take medicine | Eve Meds<br>Morning Meds | R2 take medecine |
| Work | Desk Activity<br>Chores | Laundry<br>R1 work in office |
| Other | Meditate<br>Master Bedroom Activity<br>Other | Night wandering<br>R2 wake<br>R1 wake<br>Other |

L'objectif de ce ré-étiquetage, selon les auteurs, est de permettre une comparaison plus juste entre les datastets dans les cas où les mêmes activités sont étiquetées différemment. Le dataset Aruba n'a pas été ré-étiqueté selon les mêmes critères car les activités proposées dans le dataset sont suffisamment proches du nouvel étiquetage.

Il faut noter que ce ré-étiquetage a aussi un effet de rééquilibrer en partie les datasets, mais aussi, paradoxalement, augmente le nombre d'exemples d'une classe particulière appelé « Autres ». Cette classe correspond à des activations de capteurs non identifiées ou à des séquences d'activation non identifiées. Comme cette classe représente souvent plus de 50 % du jeu de données, elle introduit un biais. Cela doit être souligné et gardé à l'esprit lors de l'analyse des résultats. Si l'algorithme de classification est capable de trouver tous les éléments de cette classe « Autre », alors la précision sera d'au moins 50 %.

Contrairement au travail original de [20], nous avons d'abord nettoyé les datastets. Après une analyse détaillée des datasets, nous avons remarqué que certain contenaient des anomalies, notamment dans le dataset de Milan : (1) ils peuvent contenir des données, des journées complètes ou partielles dupliquées ; (2) certaines activations de capteur contiennent des erreurs dues aux problèmes de communications des capteurs ; (3) les traces d'activations des capteurs dans le dataset peuvent ne pas être correctement ordonnées temporairement, c'est-à-dire dans l'ordre chronologique des timestamps.

Pour pallier cela nous avons supprimé le double des journées dupliqué. Nous avons corrigé les activations de capteur contenant des erreur de codage tel que « Oc » transformé en « ON » ou « OFc » en « OFF ». De plus, nous avons réordonné les activations de capteur selon leur timestamps.

Il est également nécessaire d'annoter chaque événement avec une étiquette d'activité, en faisant attention au début et à la fin des activités. Les activités dans les ensembles de données sont étiquetées avec un mot clé « début » ou « fin » pour déterminer quand une activité commence et quand elle se termine. Cependant, des activités peuvent être encapsulées dans d'autres, c'est-à-dire qu'une activité commence avec le mot clé « début » ; puis, quelques événements plus tard, une nouvelle activité commence sans que l'activité précédente ne se soit terminée par le mot clé « fin ». Il est donc important de prêter attention à ces cas particuliers lors de la pré-segmentation du dataset en séquences d'activités.

Nous avons observé en reproduisant le travail de [20] que ce nettoyage avait un impact sur les résultats finaux. Nous avons observé une perte de 5 points de précision sur le dataset de Milan en utilisant le modèle LSTM bidirectionnel de [20]. Cette perte s'explique notamment par une diminution du nombre d'occurrences de la classe « Autre ».

### 4.3. Matériel et logiciel

Les expériences ont été menées sur un serveur, avec un CPU Intel(R) Xeon(R) E5-2640 v3 2.60 GHz, disposant de 32 CPU, 128 Go de RAM et une carte graphique NVIDIA Tesla K80.

Les frameworks Keras et Tensorflow ont été utilisés pour l'implémentation des algorithmes. L'algorithme Word2Vec a été entraîné grâce à la bibliothèque Gensim [34], car cette librairie est très connue et souvent utilisée pour l'entraînement de modèles du domaine du TAL. Nous avons utilisé la méthode « early stop » fournie par le framework Tensorflow pour arrêter les entrainements avant overfitting (sur apprentissage). L'entraînement de Word2Vec a été arrêté après un nombre maximum d'époques, voir Tableau 4.3, dans la mesure où la bibliothèque Gensim ne fournit pas cette méthode.

TABLE 4.3 – Hyperparametre des Embeddings

| | Embedding | Word2Vec | ELMo |
|---|---|---|---|
| Taille de la projection | 64 | 64 | 64 |
| Taille de la fenetre de contexte | None | 20 | 60 |
| Nombre d'époques Max | 400 | 100 | 400 |
| Taille du Batch | None | None | 512 |

### 4.4. MÉTHODE D'ÉVALUATION

Dans cette expérience, la méthode standard de validation croisée stratifiée K-fold [28] a été choisie. Cette méthode consiste, après avoir segmenté le dataset en séquences d'activités, à les répartir en K parties, ici, K est de 3. Chaque partie contient 33 % de l'ensemble de données. Les parties sont dites stratifiées. Elles conservent un pourcentage d'échantillons de chaque classe, c'est-à-dire que chaque classe d'activité est présente dans chaque partie. À partir de ces K parties, K passages sont effectués. Chaque passage utilise K-1 parties comme ensemble d'entraînement et la partie restante comme ensemble de test. Pendant la phase d'apprentissage, 20 % de l'ensemble d'apprentissage est utilisé pour la validation afin de suivre et d'arrêter l'apprentissage des modèles avant le sur-apprentissage (overfitting). Les résultats, rapportés dans notre étude, montrent la moyenne des scores obtenus sur les ensembles de test.

Afin de comparer et évaluer les performances des algorithmes nous avons utilisé en plus de l'Accuracy, la Precision, le Recall et le F1-Score. Le F1-score est une façon de combiner la Precision et le Recall du modèle, elle est définie comme la moyenne harmonique de ces deux métriques. De plus, nous avons utilisé des versions pondérées de ces mêmes métriques (Balanced Accuracy et la Weighted F1-score, Precision er Recall). La Balanced Accuracy est une version adaptée de l'Accuracy pour des datasets déséquilibré. Elle prend en compte le nombre d'apparition de chaque classe. Les versions Weighted, sont une version pondérée par nombre d'apparition de chaque classe, permettant de visualiser si le modèle se concentre sur les classes les plus représentées.

## 5. RÉSULTATS

Dans cette section, nous allons dans un premier temps évaluer l'approche série temporelle du FCN sur les trois datasets contre un classifieur utilisant un LSTM bidirectionnel [20]. Puis, dans un second temps, nous allons observer ce que les embeddings

Word2Vec et ELMo ont appris de manière non supervisée. Nous comparerons ensuite notre approche Word2Vec et ELMo contre une approche basée sur une structure utilisant un embedding non pré-entraîné [20] pour évaluer si ces méthodes sont pertinentes et apporte un gain en termes de classification des AVQs. Le modèle ELMo peut être approximé par une couche d'embedding, suivie de deux couches de LSTM bidirectionnel. Nous ferons une étude comparative entre le model ELMo suivi d'un classifieur utilisant un LSTM bidirectionnel et une structure contenant deux couches de LSTM bidirectionnel. Enfin, nous évaluerons la capacité d'apprentissage par transfert d'un embedding ELMo pré-entraîné. Nous avons utilisé l'un des trois embeddings ELMo pré-entraînés, ici Aruba, pour entraîner un classificateur LSTM bidirectionnel sur le dataset de Cairo. Le code de l'ensemble de ces travaux sont disponible sur notre dépôt Git [33]

### 5.1. Comparaison FCN et LSTM bidirectionnel

Pour évaluer si un classifieur de séries temporelles peut être appliqué à la RHA avec des capteurs domotique, nous avons comparé les résultats obtenus contre une méthode LSTM bidirectionnelle. Les travaux de [20] ont montré l'efficacité du LSTM bidirectionnel face à d'autres méthodes de l'état de l'art.

L'un des avantages à l'utilisation de structure basée sur des CNN comme le FCN est le gain en temps d'entraînement. Les CNN sont reconnus pour cet avantage non négligeable lorsque les séquences à traiter sont longues. Cependant, avant d'évaluer les FCN face au LSTM, nous avons comparé la méthode en utilisant le zero-padding ou deux autre type de padding, le symetric-padding et circular-padding sur les trois datasets. Le Tableau 5.1 montre que le symetric padding permet au FCN d'obtenir des scores plus élevés sur deux datasets Aruba et Milan permet d'augmenter le F1-score de 139,2 % et 310 % respectivement. Néanmoins, le dataset Cairo obtient de très mauvais scores, notament en therme d'Accuracy et de Balanced Accuracy. Le FCN obtient son meilleur score sur ce dernier avec le circular padding. Nous pouvons expliquer potentiellement les mauvais scores sur le dataset Cairo par le fait qu'il s'agisse d'un dataset contenant deux résidents. Nous pensons qu'il est difficile pour le FCN d'identifier des motifs clairs parce que les séquences d'activité sont bruitées du

Table 5.1 – Comparaison entre zero padding, symetric padding circular padding

| | FCN zero-padding | | | FCN symetric-padding | | | FCN circular-padding | | |
|---|---|---|---|---|---|---|---|---|---|
| | Aruba | Milan | Cairo | Aruba | Milan | Cairo | Aruba | Milan | Cairo |
| Accuracy | 65.25 | 43.10 | 53.05 | **95.01** | **82.18** | 36.46 | 94.09 | 80.12 | **53.3** |
| Precision | 77.05 | 21.24 | 36.69 | **94.46** | **81.85** | **42.95** | 93.61 | 80.35 | 35.01 |
| Recall | 71.97 | 43.99 | **52.46** | **94.85** | **82.15** | 43.26 | 93.93 | 79.87 | 51.83 |
| F1-score | 67.93 | 28.45 | **39.4** | **94.55** | **81.42** | 37.35 | 93.58 | 79.41 | 37.56 |
| Balance Accuracy | 23.31 | 9.49 | 17.08 | **76.79** | **66.36** | **19.57** | 72.47 | 65.11 | 17.15 |
| Weighted Precision | 44.68 | 4.27 | 17.1 | **78.94** | **69.87** | 15.61 | 77.86 | 65.07 | **21.29** |
| Weighted Recall | 28.68 | 8.45 | 15.2 | **77.54** | **66.43** | **16.16** | 73.73 | 66.48 | 14.19 |
| Weighted F1-score | 28.0 | 5.63 | 11.35 | **77.41** | **65.42** | 10.5 | 73.84 | 64.62 | **13.67** |

fait des deux résidents. De plus, l'augmentation de données provoquées par les padding peut venir ajouter de la complexité dans la recherche de motifs précis.

Le résultat de l'expérience comparative entre le FCN et le LSTM bidirectionnel est visible dans le Tableau 5.2. Les résultats montrent qu'en dépit de l'amélioration proposée, le FCN reste en retrait par rapport au LSTM bidirectionnel. Le LSTM bidirectionnel gagne sur tout les dataset avec près de 3, 10 et 45 points de pourcentage en F1-score pour les dataset Aruba, Milan et Cairo respectivement. Le FCN étant une structure reconnue pour sa capacité à extraire des caractéristiques pertinentes et ses performances en termes de classification de série temporelle, ne permet pas de classifier des séquences d'activités.

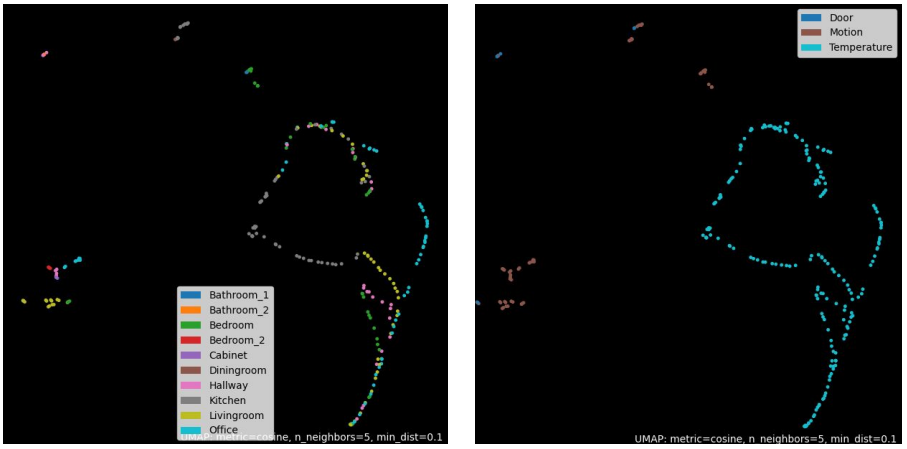
TABLE 5.2 – Comparaison entre Liciotti (embedding + bi LSTM) et FCN (embedding + FCN)

| | Liciotti | | | FCN | | |
|---|---|---|---|---|---|---|
| | Aruba | Milan | Cairo | Aruba | Milan | Cairo |
| Accuracy | **96.52** | **90.54** | **84.99** | 94.09 | 80.12 | 53.3 |
| Precision | **96.11** | **90.08** | **83.17** | 93.61 | 80.35 | 35.01 |
| Recall | **96.50** | **90.45** | **82.98** | 93.93 | 79.87 | 51.83 |
| F1-score | **96.22** | **90.02** | **82.18** | 93.58 | 79.41 | 37.56 |
| Balance Accuracy | **79.96** | **74.31** | **77.52** | 72.47 | 65.11 | 17.15 |
| Weighted Precision | **82.30** | **82.03** | **80.03** | 77.86 | 65.07 | 21.29 |
| Weighted Recall | **80.71** | **75.51** | **73.82** | 73.73 | 66.48 | 14.19 |
| Weighted F1-score | **81.21** | **77.74** | **74.84** | 73.84 | 64.62 | 13.67 |

### 5.2. VISUALISATION DE L'EMBEDDING WORD2VEC

Après avoir entraîné les différents modèles Wor2Vec sur les datasets, nous pouvons visualiser les relations apprises entre les activations de capteur, en projetant dans un espace à deux dimensions le vecteur de chaque activation de capteur produit par ces embeddings. Pour ce faire nous récupérons le vecteur de dimension $d$ de chaque activation de capteur dans l'embedding, puis nous utilisons l'algorithme Uniform Manifold Approximation and Projection (UMAP) [24] pour réduire à deux dimension les vecteurs. Nous obtenons alors la représentation présentée par la Figure 5.1a. Chaque point correspond à une activation de capteur. Pour mieux comprendre et interpréter cette visualisation nous avons coloré chaque point en fonction de la pièce de la maison dans laquelle il se trouve. Il apparaît alors que les points dans la plupart des regroupements sont de la même couleur, ce qui nous indique que le modèle a pris en compte la notion que ces activations de capteurs sont colocalisés. Certains points de couleurs différentes viennent s'ajouter à certains groupes, ce qui indique que ces capteurs situés dans une autre pièce s'activent régulièrement avec ceux du groupe en

question. Ce modèle est donc en mesure d'apprendre quels capteurs s'activent régulièrement ensemble. On remarque aussi que les capteurs de température sont plus ou moins isolés du reste des autres capteurs, Figure 5.1b. Cela indique que le modèle est capable de faire une différence entre les capteurs de température et les autres types de capteur.



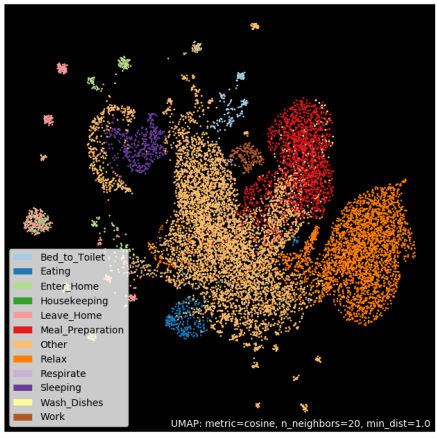

(a) Colorisation par pièces de la maison.

(b) Colorisation par type de capteur.

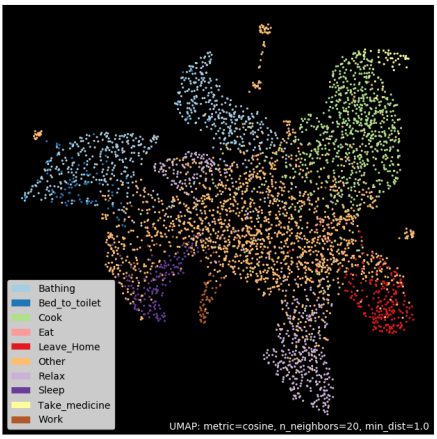
FIGURE 5.1 – Interprétations de l'embedding Word2Vec pour le dataset Aruba

Dans le but de visualiser l'embedding des séquences d'activité, nous avons ajouté au modèle Word2Vec une couche de Global Average Pooling [21] afin de transformer notre séquence de vecteurs en un seul vecteur. Une fois que l'ensemble des séquences d'activation est transformé en un vecteur, nous utilisons UMAP pour réduire chacun des vecteurs d'activité à deux dimensions. Chaque vecteur d'activité est ensuite affiché et étiqueté avec une couleur correspondant à l'étiquette de l'activité (voir Figure 5.2).

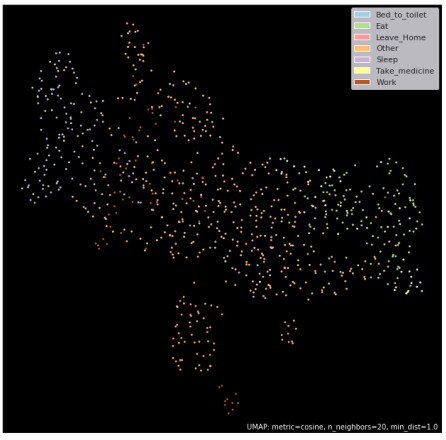
Nous pouvons observer sur la Figure 5.2 que les activités appartenant à la même classe (de même couleur) semblent regroupées. Les activités correspondant à la classe « Autre » (en orange clair) sont principalement concentrées au centre de l'image, tandis que les autres classes d'activités se retrouvent en périphérie. Très peu de clusters distincts apparaissent. Tous les clusters sont connectés par les points correspondant aux séquences labellisées « Autre ». Cette représentation nous permet d'affirmer que Word2Vec est capable d'extraire des caractéristiques capables de classer les séquences d'activités. Cependant, la méthode d'embedding Word2Vec ne semble pas être assez efficace pour isoler dans des clusters individuels toutes les classes d'activité. Certaines classes très ressemblantes comme « Enter Home » et « Leave Home » peuvent être regroupées ensemble.




(a) Word2Vec : Séquences d'activités de Aruba




(b) Word2Vec : Séquences d'activités de Milan

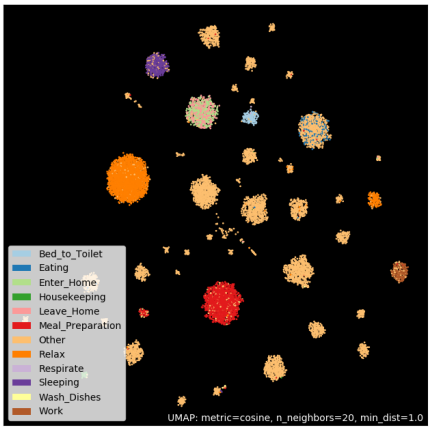



(c) Word2Vec : Séquences d'activités de Cairo

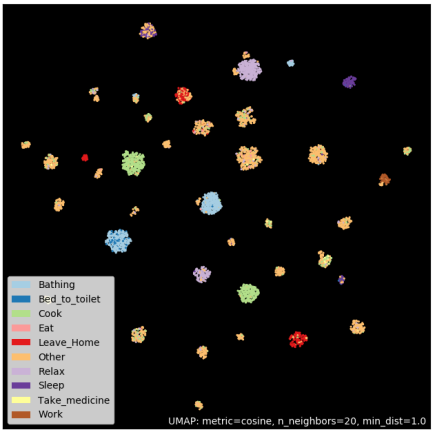
FIGURE 5.2 – Embedding des séquences d'activités avec Word2Vec

## 5.3. VISUALISATION DE L'EMBEDDING ELMO

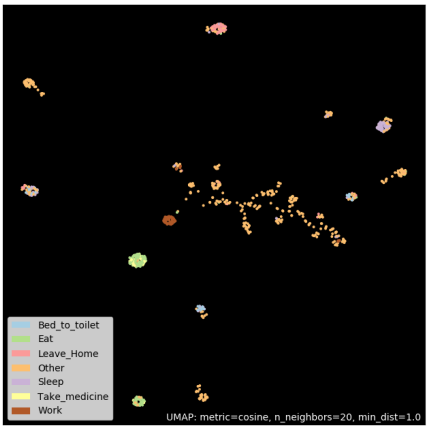
Le principal avantage de ELMo est de pouvoir fournir plus d'une représentation pour chaque mot en fonction du contexte dans lequel il apparaît. Ce ne serait pas pertinent de visualiser l'embedding ELMo de mots isolés dans la mesure où le vecteur mot fourni par cet embedding dépend des mots environnants. Mais la visualisation de l'embedding des séquences d'activité laisse apparaître quelques indices intéressants. Pour visualiser la représentation des séquences d'activation, nous avons procédé de la même manière que pour la méthode Word2vec (voir Figure 5.3).




(a) ELMo : Séquences d'activités de Aruba




(b) ELMo : Séquences d'activités de Milan




(c) ELMo : Séquences d'activités de Cairo

FIGURE 5.3 – Embedding des séquences d'activités avec ELMo

Par rapport à la méthode Word2Vec, les clusters proposés par l'association d'ELMo et de UMAP sont plus isolés les uns des autres. Cela signifie que ELMo est capable d'extraire des caractéristiques plus révélatrices que Word2Vec. Des clusters isolés appartenant à la classe d'activité « Autre« sont apparus. Nous supposons que ces groupes de séquences d'activation, étiquetées « Autres », sont en fait de nouvelles classes d'activité potentielles. ELMo associée à UMAP semble être une solution envisageable pour découvrir de nouvelles activités. Cette visualisation démontre la capacité de ELMo à générer des caractéristiques pertinentes à partir des activations brutes des capteurs.

La combinaison de l'embedding ELMo pré-entraînée et d'un algorithme de réduction de dimension tel que l'UMAP semble capable de réaliser un regroupement non supervisé de séquences d'activités similaires. L'entraînement de ELMo et de UMAP se fait de manière non supervisée, c'est-à-dire en n'utilisant aucun label. Il semble donc possible de regrouper des séquences d'activité pré-segmentées de manière totalement non supervisée. En d'autres termes, s'il est possible de diviser un dataset en séquences d'activité non étiquetées, en utilisant ELMo et UMAP il semble capable de regrouper les séquences appartenant à la même classe ou une classe similaire.

Cependant, ce clustering n'est pas parfait, lorsqu'on regarde de près les clusters. Certains points d'une couleur différente de la couleur majoritaire du cluster, peuvent apparaître. Cette confusion peut s'expliquer de deux manières. Premièrement, certaines séquences appartenant à deux classes différentes peuvent être très similaires en termes d'activation des capteurs, car il n'y a pas assez de capteurs pour les distinguer. Deuxièmement, il s'agit potentiellement d'une erreur d'étiquetage dans le dataset. En effet, il est difficile de créer et d'étiqueter des datasets pour des activités humaines dans les maisons [6].

### 5.4. Comparaison contre des méthodes sans contexte

Pour évaluer la méthode que nous proposons, nous comparons nos résultats aux travaux de [20]. Leurs travaux ont montré que les LSTM et les LSTM bidirectionnels avec une couche d'embedding sont performants pour la RAH. Nous ajoutons également deux modèles pour comparaison, un LSTM et un LSTM bidirectionnel sans embedding, afin d'évaluer l'apport qu'apporte cette couche.

Les résultats de l'expérience dans le tableau 5.3 montrent que l'utilisation d'un embedding améliore les performances de classification globale. Cette couche à la capacité de générer des caractéristiques de similarité entre les activations des capteurs.

Contre toute attente, nous remarquons que le modèle Word2Vec n'a pas amélioré les performances de classification par rapport aux embedding standard. Au contraire, l'utilisation de Word2Vec diminue les performances, sauf sur le dataset Aruba où il est plus performant que le modèle de référence (Liciotti). Cela signifie que Word2Vec n'a pas pris en compte certaines caractéristiques importantes.

ELMo est plus performant que les autres approches sur tous les dataset. Il est en particulier très performant sur le jeu de données multi-utilisateurs Cairo. Il permet d'augmenter le F1-score de 5 points et le F1-score pondéré de 10 points. Nous supposons qu'un datasets multi-utilisateurs nécessite la compréhension du contexte ou d'un certain ordre d'activation afin de différencier les activités des utilisateurs.

De plus, nous pouvons observer que l'embedding ELMo permet au FCN, qui jusque-là n'offrait pas de bonne performance, de revenir dans la compétition voir de dépasser légèrement les scores de l'approche de Liciotti sur le dataset Cairo. Ceci nous permet de confirmer l'apport de cette approche a base de contextualisation grâce à la méthode ELMo.

TABLE 5.3 – Résultats de classification sur les trois datasets Aruba, Milan et Cairo

| | Aruba | | | | | Milan | | | | | Cairo | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | No Embedding | Liciotti | W2V | ELMo | ELMo +FCN | No Embedding | Liciotti | W2V | ELMo | ELMo +FCN | No Embedding | Liciotti | W2V | ELMo | ELMo +FCN |
| Accuracy | 95.01 | 96.52 | 96.59 | **96.76** | 94.59 | 82.24 | **90.54** | 88.33 | 90.14 | 85.33 | 81.68 | 84.99 | 82.27 | **89.12** | 84.39 |
| Precision | 94.69 | 96.11 | 96.23 | **96.43** | 94.09 | 82.28 | 90.08 | 88.28 | **90.2** | 85.31 | 80.22 | 83.17 | 82.04 | **88.41** | 83.91 |
| Recall | 95.01 | 96.50 | 96.59 | **96.69** | 94.53 | 82.24 | **90.45** | 88.33 | 90.31 | 85.36 | 81.68 | 82.98 | 82.27 | **87.59** | 82.27 |
| F1-score | 94.74 | 96.22 | 96.32 | **96.42** | 94.10 | 81.97 | 90.02 | 87.98 | **90.1** | 85.02 | 80.49 | 82.18 | 81.14 | **87.48** | 82.27 |
| Balance Accuracy | 77.73 | 79.96 | **81.06** | 79.98 | 69.57 | 67.77 | 74.31 | 73.61 | **78.25** | 70.86 | 70.09 | 77.52 | 69.38 | **87.00** | 80.29 |
| Weighted Precision | 79.75 | 82.30 | 82.97 | **88.64** | 73.23 | 79.6 | 82.03 | 84.42 | **87.56** | 83.74 | 68.45 | 80.03 | 77.56 | **86.83** | 79.28 |
| Weighted Recall | 77.73 | 80.71 | **81.06** | 79.17 | 69.27 | 67.77 | 75.51 | 73.62 | **78.75** | 70.49 | 70.09 | 73.82 | 69.38 | **84.78** | 80.08 |
| Weighted F1-score | 77.92 | 81.21 | 81.43 | **81.93** | 69.30 | 71.81 | 77.74 | 76.59 | **82.26** | 74.79 | 68.47 | 74.84 | 70.95 | **84.71** | 78.13 |

## 5.5. APPROXIMATION D'ELMO

Le modèle ELMo peut être approximé par une couche LSTM bidirectionnelle avec une couche d'intégration. Dans cette expérience, nous comparons la méthode ELMo à deux couches empilées de LSTM bidirectionnel avec une couche d'intégration. Le tableau 5.4 montre les résultats de cette comparaison.

Les résultats montrent que la structure ELMo obtient un gain non négligeable sur les trois jeux de données sauf sur le jeu de données de Milan. Sur celui-ci, les gains sont moindres, mais non négligeables sur les scores pondérés. ELMo permet tout de même de reconnaître plus de classes et est plus précis que les autres structures. L'empilement de LSTM bidirectionnels ne permet pas d'obtenir de meilleures performances. Ce type de structure peut même dégrader les performances dans certains cas, comme sur le jeu de données du Cairo, ou d'Aruba. Il semble que la méthode d'entraînement d'ELMo permette de capturer des caractéristiques plus utiles.

TABLE 5.4 – Comparaison avec une approximation de ELMo

| | Aruba | | | Milan | | | Cairo | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1L | 2L | ELMo | 1L | 2L | ELMo | 1L | 2L | ELMo |
| Accuracy | 96.52 | 96.46 | **96.76** | **90.54** | 90.03 | 90.14 | 84.99 | 84.99 | **89.12** |
| Precision | 96.11 | 96.04 | **96.43** | 90.08 | **90.22** | 90.20 | 83.17 | 85.04 | **88.41** |
| Recall | 96.50 | 96.41 | **96.69** | **90.45** | 90.28 | 90.31 | 82.98 | 84.4 | **87.59** |
| F1-score | 96.22 | 96.13 | **96.42** | 90.02 | 90.07 | **90.10** | 82.18 | 84.08 | **87.48** |
| Balance Accuracy | 79.96 | 78.74 | **79.98** | 74.31 | 75.51 | **78.25** | 77.52 | 76.52 | **87.00** |
| Weighted Precision | 82.30 | 82.01 | **88.64** | 82.03 | 84.29 | **87.56** | 80.03 | 80.87 | **86.83** |
| Weitghed Recall | **80.71** | 79.05 | 79.17 | 75.51 | 77.31 | **78.75** | 73.82 | 76.6 | **84.78** |
| Weight F1-score | 81.21 | 79.97 | **81.93** | 77.74 | 79.29 | **82.26** | 74.84 | 77.44 | **84.71** |

### 5.6. Transfer Learning

Dans le domaine du TLN, les embeddings sont pré-entraînés sur de grands corpus, puis utilisés sur un corpus plus spécifique pour effectuer des tâches particulières, comme la classification de textes. C'est le principe de l'apprentissage par transfert. L'objectif est d'utiliser les caractéristiques très génériques du grand corpus sur un corpus plus petit et spécifique pour gagner en temps d'apprentissage, mais aussi en généricité.

Dans le cas de la reconnaissance des AVQs, cette pratique permettrait de transférer les connaissances d'une maison intelligente à une autre afin que cette dernière puisse reconnaître les AVQs sans entraînement supplémentaire. Ce modèle transféré pourrait ensuite être affiné pour le contexte de cette nouvelle maison. Nous avons expérimenté cette pratique en utilisant l'intégration ELMo d'Aruba sur le jeu de données du Cairo.

Pour ce faire, nous avons entraîné le modèle ELMo sur le jeu de données Aruba. Ensuite, nous avons utilisé ce modèle entraîné pour extraire et encoder les trames des séquences d'activité du jeu de données du Cairo. Ces caractéristiques sont ensuite données en entrée à un classificateur. Le classificateur est un réseau neuronal composé d'un LSTM bidirectionnel suivi d'une couche softmax. Les poids de l'incorporation ELMo ont été gelés, et seuls le LSTM bidirectionnel et le softmax ont été entraînés pour classifier les activités du second jeu de données. Les résultats de l'expérimentation sont présentés dans le tableau 5.5.

Ces résultats montrent que les caractéristiques génériques apprises sur le jeu de données Aruba ont permis la classification des activités du jeu de données Cairo avec des scores équivalents à l'intégration ELMo entraînée sur Cairo. Nous supposons que l'intégration ELMo d'Aruba a été capable de capturer suffisamment de caractéristiques sur la « syntaxe », l'ordre d'activation des capteurs, ainsi que la nature du capteur activé, pour encoder efficacement les séquences d'activation, et ce, malgré un vocabulaire différent.

En effet, les deux jeux de données ne comportent pas le même nombre de capteurs. Le nom des capteurs est également différent dans certains cas. Le jeu de mots est différent d'une maison à l'autre, mais dans notre cas, les jeux de données d'Aruba et du Cairo font partie des jeux de données CASAS qui utilisent le même type de capteur et la même structure de dénomination. Les capteurs suivent la structure « type de capteur » + indice. Cependant, cette expérience n'aurait pas pu fonctionner pleinement si le vocabulaire était trop différent à cause des cas de hors vocabulaire. Nous observons que, même si un mot n'a pas la même « signification » d'un jeu de données à l'autre (par exemple, « M001ON » correspond au capteur de mouvement de la cuisine dans le jeu de données *A* et au capteur de mouvement de la salle de bain dans le jeu de données *B*), l'encodage fourni par l'intégration ELMo génère des modèles qui permettent au classificateur d'atteindre des performances équivalentes à celles d'un modèle entièrement entraîné sur le jeu de données de destination. Nous supposons que ces bonnes performances proviennent du fait que ELMo prend en compte l'ordre

des mots. Ainsi, même si les mots en entrée ont changé, la syntaxe est capturée par le classifieur, c'est-à-dire l'ordre de chaque mot ainsi que les schémas de leur récurrence dans la séquence. Ces résultats indiquent l'importance des encastrements contextuels.

TABLE 5.5 – Comparaison entre ELMo entrainé sur Cairo et ELMo entrainé sur Aruba, appliqué à Cairo (Classifieur bidirectionnel LSTM)

| | Cairo | |
|---|---|---|
| | ELMo entraîné sur Cairo | ELMo entraîné sur Aruba |
| Accuracy | 89.12 | **89.24** |
| Precision | **88.41** | 87.77 |
| Recall | **87.59** | 86.35 |
| F1-score | **87.48** | 85.88 |
| Balance Accuracy | **87.00** | 84.02 |
| Weighted Precision | 86.83 | **87.55** |
| Weighted Recall | **84.78** | 79.56 |
| Weighted F1-score | **84.71** | 80.80 |

## 6. CONCLUSION

La reconnaissance de l'activité humaine est un domaine de recherche très dynamique et stimulant qui joue un rôle crucial dans diverses applications, notamment pour les maisons intelligentes. Ces environnements IoT nécessitent une technologie robuste d'apprentissage de l'activité pour fournir des services adéquats aux résidents. La topologie des maisons, leurs différentes installations de capteurs, d'actionneurs et les différentes habitudes de vie des résidents ajoutent de la variabilité aux données des capteurs. La modélisation de l'activité est donc un défi. Il ne s'agit pas seulement d'un problème de reconnaissance de motifs, mais aussi d'un problème d'analyse de séquences spatio-temporelles, où la sémantique et le contexte de chaque déclenchement de capteur peuvent changer la signification d'une activation de capteur. De plus, la nature du capteur activé peut donner un certain nombre d'informations sur l'activité en cours.

Dans cette étude, nous avons proposé une nouvelle approche, appliquée pour la première fois au domaine de la reconnaissance des activités de la vie quotidienne dans les maisons intelligentes. Nous avons utilisé des techniques du domaine de la TLN pour capturer le contexte et la sémantique des activations des capteurs dans un espace d'embedding. Cette approche permet la reconnaissance d'un plus grand nombre de classes d'activités malgré le fait que les datasets restent déséquilibrés. En effet, moins de séquences d'activation sont confondues avec la classe « Autre », qui représente néanmoins plus de 50 % des datasets.

La visualisation de l'embedding Word2vec nous a permis de nous rendre compte que cette méthode apprend certaines relations entre les activations des capteurs. Il apparaît que les capteurs de même nature sont proches les uns des autres dans cet espace. De plus, les clusters qui y apparaissent représentent différentes pièces de la maison.

Notre expérimentation montre que la capture du contexte de l'activation d'un capteur via l'embedding ELMo permet d'améliorer la classification de séquences d'activités, en particulier sur des datasets contenant des activités réalisées par plusieurs résidents.

Enfin, nous avons pu évaluer qu'un embedding ELMo entraîné dans une maison pouvait être réutilisé dans un nouvel environnement contenant une autre dénomination de capteurs et permettre un taux élevé de classification d'activités. Il est à noter que ces méthodes sont capables d'extraire des informations génériques, transférables à d'autres datasets. Cette dernière observation suggère que l'apprentissage par transfert entre environnements est possible grâce à ces méthodes, comme c'est le cas aujourd'hui dans le domaine du TLN.

Grâce à notre proposition, qui combine un modèle de langage intégrant la sémantique des activations des capteurs et un algorithme de classification par séries temporelles, nos résultats expérimentaux sur des données réelles de la maison intelligente soulignent l'importance d'une représentation sémantique dynamique contextualisée dans la reconnaissance des ADL. De plus, une telle représentation peut être partagée entre plusieurs ensembles de données pour permettre l'apprentissage par transfert. Ces résultats pourraient être la clé pour résoudre les principaux problèmes des données de la maison intelligente : la rareté et la variabilité qui empêchent toute généralisation possible des modèles de reconnaissance ADL.

Dans un travail futur, nous prévoyons d'appliquer des méthodes d'apprentissage non supervisé basées sur des transformers, tels que BERT [14] ou GPT [32]. En effet, les transformers sont devenus l'état de l'art dans le domaine du TLN grâce à leur capacité à retenir des dépendances distantes et à focaliser l'attention sur les éléments importants dans les séquences. Notre objectif, même via ces structures à base de transformers, reste le même : capturer un contexte plus large, mais aussi utiliser des méthodes de codage des mots plus avancées pour prendre en compte l'activation de capteurs inconnus. Il faut noter que la méthode proposée ici est limitée par une certaine taille de vocabulaire ou d'activation de capteurs possibles. Il est actuellement impossible d'obtenir une représentation des valeurs de capteurs qui n'ont jamais été observés. Des méthodes telles que le codage par paire d'octets (BPE) [37] ou WordPiece [47] pourraient être envisagées pour diviser les mots, représentant une activation de capteur, en sous-mots ou en compositions de mots. Cela devrait permettre aux modèles d'interpréter davantage de sémantique concernant la construction de ces mots, et donc de prendre en compte de nouvelles valeurs d'activation.

Un second axe de travail concerne la génération de dataset, élément indispensable à tous ces algorithmes d'apprentissage. Comme nous l'avons vu, la question de la qualité

de la labellisation, du volume de données, des différentes topologies et rythme de vie, est un enjeu. Or, produire de tels dataset sur la base d'expériences en situations réelle sur le long terme est coûteux et chronophage. Pour pallier cela, nous nous intéressons à la possibilité de générer des données de synthèses, en s'appuyant sur le concept de Jumeau Numérique [3]. L'idée est de développer des simulateurs d'habitations domotisées réelles ou imaginaires, et d'y faire jouer des activités quotidiennes par un usager, lui aussi numérique, un avatar. Pour ce faire, nous avons commencé à adapter le simulateur VirtualHome [31], outil initialement développé pour des algorithmes de vision par ordinateur. Les ajouts en cours concernent le développement d'objets domotiques virtuels pour ces habitats, les stratégies pour scripter les comportements des avatars, et la gestion du temps simulé.

Nous prévoyons de réaliser des AVQs dans un appartement intelligent et d'en enregistrer les traces de capteurs. Puis, nous reproduirons dans notre simulateur l'appartement intelligent pour y faire rejouer les activités par l'avatar afin de comparer les traces de capteur virtuel et réel produites. Enfin nous prévoyons d'entraîner des algorithmes de RAH à partir des données virtuelles et de valider l'apprentissage sur les données réelles. Ces travaux pourront éventuellement permettre de pré-entraîner des algorithmes de RAH dans une version numérique d'une habitation, sans devoir enregistrer plusieurs mois de données et sans faire labelliser ces données par les résidents.

## Bibliographie


[1] E. Abramova, K. Makarov & A. Orlov, « Method for Undefined Complex Human Activity Recognition », in *2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, IEEE, 2021, p. 797-801.

[2] M. Andries, O. Simonin & F. Charpillet, « Localization of humans, objects, and robots interacting on load-sensing floors », *IEEE Sensors Journal* **16** (2015), n° 4, p. 1026-1037.

[3] B. R. Barricelli, E. Casiraghi & D. Fogli, « A survey on digital twin: definitions, characteristics, applications, and design implications », *IEEE access* **7** (2019), p. 167653-167671.

[4] P. Bojanowski, E. Grave, A. Joulin & T. Mikolov, « Enriching word vectors with subword information », *Transactions of the Association for Computational Linguistics* **5** (2017), p. 135-146.

[5] D. Bouchabou, S. M. Nguyen, C. Lohr, I. Kanellos & B. Leduc, « Fully Convolutional Network Bootstrapped by Word Encoding and Embedding for Activity Recognition in Smart Homes », in *IJCAI 2020 Workshop on Deep Learning for Human Activity Recognition* (Yokohama, Japan), 2021.

[6] D. Bouchabou, S. M. Nguyen, C. Lohr, B. LeDuc, I. Kanellos et al., « A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning », *Sensors* **21** (2021), n° 18, article no. 6037.

[7] ———, « Using Language Model to Bootstrap Human Activity Recognition Ambient Sensors Based in Smart Homes », *Electronics* **10** (2021), n° 20, article no. 2498.

[8] H. Cao, F. Xu, J. Sankaranarayanan, Y. Li & H. Samet, « Habit2vec: Trajectory semantic embedding for living pattern recognition in population », *IEEE Transactions on Mobile Computing* **19** (2019), n° 5, p. 1096-1108.

[9] M. Chan, D. Estève, C. Escriba & É. Campo, « A review of smart homes – Present state and future challenges », *Computer methods and programs in biomedicine* **91** (2008), n° 1, p. 55-81.

[10] D. J. Cook, A. S. Crandall, B. L. Thomas & N. C. Krishnan, « CASAS: A smart home in a box », *Computer* **46** (2012), n° 7, p. 62-69.

[11] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee & H. Moon, « Sensor-based and vision-based human activity recognition: A comprehensive survey », *Pattern Recognition* **108** (2020), article no. 107561.

[12] E. De-La-Hoz-Franco, P. Ariza-Colpas, J. M. Quero & M. Espinilla, « Sensor-based datasets for human activity recognition – a systematic review of literature », *IEEE Access* **6** (2018), p. 59192-59210.
[13] Desa. UN, *World population prospects 2019: Highlights*, United Nations Department for Economic and Social Affairs, New York, NY, 2019.
[14] J. Devlin, M.-W. Chang, K. Lee & K. Toutanova, « Bert: Pre-training of deep bidirectional transformers for language understanding », https://arxiv.org/abs/1810.04805, 2018.
[15] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar & P.-A. Muller, « Deep learning for time series classification: a review », *Data Mining and Knowledge Discovery* **33** (2019), nº 4, p. 917-963.
[16] M. Gochoo, T.-H. Tan, S.-H. Liu, F.-R. Jean, F. S. Alnajjar & S.-C. Huang, « Unobtrusive activity recognition of elderly people living alone using anonymous binary sensors and DCNN », *IEEE journal of biomedical and health informatics* **23** (2019), nº 2, p. 693-702.
[17] R. A. Hamad, A. S. Hidalgo, M.-R. Bouguelia, M. E. Estevez & J. M. Quero, « Efficient activity recognition in smart homes using delayed fuzzy temporal windows on binary sensors », *IEEE journal of biomedical and health informatics* **24** (2019), nº 2, p. 387-395.
[18] R. A. Hamad, L. Yang, W. L. Woo & B. Wei, « Joint learning of temporal models to handle imbalanced data for human activity recognition », *Applied Sciences* **10** (2020), nº 15, article no. 5293.
[19] H. Larochelle, D. Erhan & Y. Bengio, « Zero-data learning of new tasks », in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, vol. 1, AAAI'08, nº 2, 2008, p. 646-651.
[20] D. Liciotti, M. Bernardini, L. Romeo & E. Frontoni, « A Sequential Deep Learning Application for Recognising Human Activities in Smart Homes », *Neurocomputing* **396** (2020), p. 501-513.
[21] M. Lin, Q. Chen & S. Yan, « Network in network », https://arxiv.org/abs/1312.4400, 2013.
[22] C. Lohr & J. Kerdreux, « Improvements of the xAAL home automation system », *Future internet* **12** (2020), nº 6, article no. 104.
[23] M. Matsuki, P. Lago & S. Inoue, « Characterizing word embeddings for zero-shot sensor-based human activity recognition », *Sensors* **19** (2019), nº 22, article no. 5043.
[24] L. McInnes, J. Healy & J. Melville, « Umap: Uniform manifold approximation and projection for dimension reduction », https://arxiv.org/abs/1802.03426, 2018.
[25] J. Medina-Quero, S. Zhang, C. Nugent & M. Espinilla, « Ensemble classifier of long short-term memory with fuzzy temporal windows on binary sensors for activity recognition », *Expert Systems with Applications* **114** (2018), p. 441-453.
[26] T. Mikolov, I. Sutskever, K. Chen, G. Corrado & J. Dean, « Distributed representations of words and phrases and their compositionality », https://arxiv.org/abs/1310.4546, 2013.
[27] G. Mohmed, A. Lotfi & A. Pourabdollah, « Employing a deep convolutional neural network for human activity recognition based on binary ambient sensor data », in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2020, p. 1-7.
[28] M. Mullin & R. Sukthankar, « Complete Cross-Validation for Nearest Neighbor Classifiers », in *Proceedings of the Seventeenth International Conference on Machine Learning* (San Francisco, CA, USA), ICML '00, Morgan Kaufmann Publishers Inc., 2000, p. 639–646.
[29] J. Pennington, R. Socher & C. D. Manning, « Glove: Global vectors for word representation », in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, p. 1532-1543.
[30] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee & L. Zettlemoyer, « Deep contextualized word representations », https://arxiv.org/abs/1802.05365, 2018.
[31] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler & A. Torralba, « Virtualhome: Simulating household activities via programs », in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, p. 8494-8502.
[32] A. Radford, K. Narasimhan, T. Salimans & I. Sutskever, « Improving language understanding by generative pre-training », 2018.
[33] « Recognition of activities of daily living using home automation sensors and deep learning context and semantic », https://github.com/dbouchabou/HAR-Context-and-Semantic.git, Accessed: 2022-03-14.
[34] R. Řehůřek & P. Sojka, « Software Framework for Topic Modelling with Large Corpora », in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (Valletta, Malta), ELRA, 2010, http://is.muni.cz/publication/884893/en, p. 45-50 (English).

[35] S. Schoeters, W. Dewulf, J.-P. Kruth, H. Haitjema & B. Boeckmans, « Description and validation of a circular padding method for linear roughness measurements of short data lengths », *MethodsX* **7** (2020), article no. 101122.

[36] M. Sedky, C. Howard, T. Alshammari & N. Alshammari, « Evaluating machine learning techniques for activity classification in smart home environments », *International Journal of Information Systems and Computer Sciences* **12** (2018), nº 2, p. 48-54.

[37] R. Sennrich, B. Haddow & A. Birch, « Neural machine translation of rare words with subword units », `https://arxiv.org/abs/1508.07909`, 2015.

[38] K. Shimoda, A. Taya & Y. Tobe, « Combining Public Machine Learning Models by Using Word Embedding for Human Activity Recognition », in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, IEEE, 2021, p. 2-7.

[39] D. Singh, E. Merdivan, S. Hanke, J. Kropf, M. Geist & A. Holzinger, « Convolutional and recurrent neural networks for activity recognition in smart environment », in *Towards integrative machine learning and knowledge extraction*, Springer, 2017, p. 194-205.

[40] D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist & A. Holzinger, « Human activity recognition using recurrent neural networks », in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, Springer, 2017, p. 267-274.

[41] K. E. Skouby, A. Kivimäki, L. Haukiputo, P. Lynggaard & I. M. Windekilde, « Smart cities and the ageing population », in *The 32nd Meeting of WWRF*, 2014.

[42] T.-H. Tan, M. Gochoo, S.-C. Huang, Y.-H. Liu, S.-H. Liu & Y.-F. Huang, « Multi-resident activity recognition in a smart home using RGB activity image and DCNN », *IEEE Sensors Journal* **18** (2018), nº 23, p. 9718-9727.

[43] D. A. Umphred, R. T. Lazaro et al., *Neurological rehabilitation*, Elsevier Health Sciences, 2012.

[44] A. Wang, S. Zhao, C. Zheng, J. Yang, G. Chen & C.-Y. Chang, « Activities of Daily Living Recognition With Binary Environment Sensors Using Deep Learning: A Comparative Study », *IEEE Sensors Journal* **21** (2020), nº 4, p. 5423-5433.

[45] Z. Wang, W. Yan & T. Oates, « Time series classification from scratch with deep neural networks: A strong baseline », in *2017 International joint conference on neural networks (IJCNN)*, IEEE, 2017, p. 1578-1585.

[46] S. Wu, G. Wang, P. Tang, F. Chen & L. Shi, « Convolution with even-sized kernels and symmetric padding », `https://arxiv.org/abs/1903.08385`, 2019.

[47] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey et al., « Google's neural machine translation system: Bridging the gap between human and machine translation », `https://arxiv.org/abs/1609.08144`, 2016.

ABSTRACT. — Neural networks based on Long Short Term Memory (LSTM) have demonstrated their efficiency in the growing field of recognition of daily life activities in smart homes,. By studying the sensor activations order and their temporal dependencies, human actions are translated as a sequence of more or less correlated events in time. However, human activity is not a sequence of actions without meaning and context. We propose to use and compare two methods coming from natural language processing to take into account the semantics and context of sensors in order to improve algorithms in activity sequence classification: Word2Vec, a static semantic embedding, and ELMo, a contextual embedding. The results, on real smart home datasets, indicate that this approach provides useful information, such as a map of sensor organization, and also reduces confusion between classes of daily activities. It achieves better performance on datasets containing concurrent activities with multiple residents or pets. Our tests also show that embeddings can be pre-trained on datasets that are different from the target dataset, thus allowing transfer learning. We thus demonstrate that taking into account the context and semantics of the sensors increases the classification performance of the algorithms and enables transfer learning.

KEYWORDS. — .

*Manuscrit reçu le 30 octobre 2021, accepté le 18 juillet 2022.*