Comparative Analysis of Artificial Hydrocarbon Networks versus Convolutional Neural Networks in Human Activity Recognition

Hiram Ponce, Lourdes Martínez-Villaseñor
Universidad Panamericana. Facultad de Ingeniería.
Augusto Rodin 498, Ciudad de México, 03920, México
hponce@up.edu.mx, lmartine@up.edu.mx

***Abstract*—Human activity recognition (HAR) has gained interest in the research communities in order to know the behavior and context of users for medical, sports performance evaluation, ambient assisted living and security applications. Recent works suggest that convolutional neural networks (CNN) are very competitive machine learning techniques for HAR. Nevertheless, CNN require many computational resources, high number of parameter tuning, and many data samples for training. In this paper, we present a comparative analysis of a novel technique, artificial hydrocarbon networks (AHN), with CNN on human activity recognition classification task. We choose to compare AHN with CNN given that it is a very well-suited machine learning technique for HAR. We show that AHN architecture is simpler to set up than CNN, it needs less hyper-parameter configuration and has a slightly better accuracy performance.**

***Index Terms*—artificial hydrocarbon networks, convolutional neural networks, machine learning, hyper-parameters, model performance**

I. Introduction

Research on human activity recognition (HAR) has received growing attention given the success of its use in medical, sports, security, ambient assisted living, and diverse mobile applications. These systems gather information in order to know the behavior and context of the user mainly using two approaches for HAR: vision based [1] and sensor based [2].

There is a trend towards implementing online human activity recognition on smartphones given that they have become more powerful in terms of CPU, memory and battery [3]. Online activity recognition means been able to identify different activities with a mobile device, but it can also mean online machine learning for adaption. In both cases, there is a need of light weight models to perform HAR that can be easily deployed [4]. Deep learning models high level features in data and deep learning structure is more feasible to perform unsupervised and incremental data [4]. Therefore, deep learning has become an important trend recently in human activity recognition [5].

Convolutional Neural Network (CNN) [6] is one of the most researched learning techniques in image classification, speech recognition, and recently in human activity recognition based on mobile and wearable sensors [5]. Convolutional neural networks are specially attractive for HAR due to its general architecture: convolutional layer, pooling layer, and fully connected layer [7]. CNNs have the ability to extract features stacking several convolutional operators to create more abstract feature hierarchies [8]. Thus, a main goal of applying a CNN in accelerometer signal would be to learn multiple discriminative features useful for HAR classification process [9]. Moreover, the advantage of using CNN for human activity recognition (HAR) is that it captures local dependencies of an activity signals and preserves feature scale invariant [10] [4]. CNN are able to capture local data patterns and variations, which is particularly useful for HAR using accelerometer time series where the form and size of periods are defined by the activity type [7].

On the other hand, a novel supervised machine learning method called artificial hydrocarbon networks (AHN) was presented and applied for human activity recognition in [11] [12]. In [11], Ponce et al. experiment results over the raw public Physical Activity Monitoring Data Set proved that the artificial hydrocarbon networks-based classifier is suitable for human activity recognition when compared to the other fourteen well-known supervised classifiers. We considered flexibility of the approach regarding the ability to support new users (user-independent) in paper [12] using AHN. The AHN-classifier performance was compared with eighteen well-known supervised techniques.Its worth noticing that features for time and frequency domain were extracted and a reduced set of features was used for classification. Hence, AHN can be used with raw signals of sensors or with extracted features.

Shoaib et al. [3] discussed that there is a need to implement multiple classifiers for comparison purposes in HAR, which includes a CPU, memory and other resources analysis. Such an analysis is important to determine the feasibility of a classifier implementation. The trade-off between accuracy and resource consumption must be taken into account. [3].

In this paper, we present a comparative analysis of AHN versus CNN in human activity recognition using the publicly available dataset [13] for human activity recognition, which has sensor signal recordings from accelerometer, gyroscopes and magnetometers.

For the evaluation of the models we considered (i) the evaluation of the performance in terms of the hyper-parameters (ii) the optimization of the architecture of the models (iii) the assessment of the performance of the models due to the input

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

size. Our goal in this work is to present AHNs as a well suited machine learning technique for HAR as its performance is similar to CNN. AHN architecture is simpler to set up than CNN, it needs less hyper-parameter configuration and shows a slightly better accuracy performance using an optimized architecture. We choose to compare AHN with CNN deep learning method given that CNN is a very competitive machine learning technique for HAR.

The rest of this paper is organized as follows. Related work is provided in Section II. We present an overview of CNN and AHN machine learning models in Section III .In Section IV, we describe our comparative analysis between AHN and CNN applied on HAR. Section V presents the results of the comparative analysis. Section VI concludes our work.

II. Related Work

In human activity recognition task, physical activities done in daily life are identified from information extracted from divers modalities of sensors such as cameras, wearable and ambient sensors. Extracting knowledge from raw activity data is critical in functional and behavioral health monitoring, game console designing, personal fitness tracking, and sport analytics [14].

The main goal of using deep learning for human activity recognition is to take advantage of the fact that features can be learned automatically through network instead of being extracted manually [4] [5]. In particular, CNN is computational expensive,require high number of hyper-parameter tuning to obtain optimal features [5], and needs a very large dataset of samples for training and testing [4].

Zeng et al. [10] proposed an approach based on CNN to extract human activity features using different public datasets with accelerometer recordings. Their purpose of using CNN was to capture local dependencies and scale-invariant features of activity signals. They used one pair convolution and max-pooling layer and two normal fully connected neural networks. CNN parameters were estimated by standard forward and backward propagation. The CNN-based algorithm was executed on a server equipped with a Tesla K20c GPU and 48G memory. They evaluated sensitivity with different pooling window size, weight decay, momentum and dropout. It is worth noticing that the performance improved in the datasets with more samples.

Ronao and Cho [15] presented a CNN as feature extractor and classifier for recognizing human activities using accelerometer and gyroscope on a smartphone .They included a greedy-wise tuning to assess the effects of different values on the performance of the CNN using sensor data. They performed experiments, varying the values for number of layers (one to four layers), number of feature maps (10-200), size of convolutional filter (1x3 up to 1x15) , and pooling size (1x2 and 1x3). They incorporated max-pooling, a learning rate of 0.01 They also increased momentum parameter from 0.5 to 0.99. They gathered a dataset with 7352 examples for training and 2947 examples for testing.Their hardware is composed of two Intel Xeon E5 CPU that drive two NVIDIA Quadro K5200 with six cores and twelve threads each. The authors proved that CNNs can exploit temporal local dependency of time-series 1D signals and translation invariance and hierarchical characteristic of the performed activities without feature hand-crafting.In their results, the accuracy of CNN outperformed other state-of-art machine learning techniques(94.79%).Nevertheless, when using additional information of the temporal fast Fourier transform of data, the CNN improved almost 1% accuracy (95.75%). Moreover, they failed to capture temporal variance in complex activity and generalization to differ activity models [5]. Adding time-frequency convolution would increase the computational expense.

Yang et al. [16] proposed a method with CNN to automate feature learning for HAR problem. They built a new deep learning architecture for CNN which feature extraction and classification of human activities are unified in one model. Their results showed that CNN method consistently outperformed support vector machine,K-nearest neighbors, and deep belief network when applied on Opportunity Activity Recognition dataset [17]. These dataset gather data from wearable, object and ambient sensors. It included 18 activity classes. The experiments were conducted on Matlab codes on a PC Intel i5-2500, 3.3 GHz CPU and 8GB RAM. They reported the duration of 1 hour training for the dataset of only 1 subject (136,869 training samples - 32,466 testing samples). The authors suggested parallel computation of the CNN to reduce training and testing time. Hyper-parameter tuning was not reported in this paper, hence their proposed method suggest that CNN is a very competitive for HAR classification task.

In summary, related work reported in literature show that CNN is a very competitive machine learning technique for HAR. Nevertheless, as other deep learning methods, CNN require many computing resources and a very large set of samples for training. Hyper parameter tuning is needed and how to find optimal parameters in CNN is still an open issue [16].

III. Background of the Learning Models

In this section, we present an overview of AHN and CNN learning models with emphasis on the hyper-parameters revised for this comparative analysis.

A. Overview of AHN

AHN is a supervised learning method, proposed by Ponce and Ponce [18], that models data using carbon networks as inspiration. It loosely simulates the chemical rules involved in hydrocarbon molecules to find a way for representing the structure and behavior of data [19]. The main feature of this model is to package data in units so-called molecules. Then, packages are organized and optimized through heuristic mechanisms based on chemical assumptions that are encoded in the training algorithm. The key features of AHN are threefold: modular organization of data, structural stability of data-packages and inheritance of packaging information [20].

As described above, the main unit of information is the molecule. It consists of a kernel function parameterized with a set of weights, as written in (1) where $x \in \mathbb{R}^n$ is the feature vector of the input data, H_i is a set of weights namely the hydrogen values, σ is a vector namely the carbon value and $k \leq 4$ is the maximum number of hydrogen values associated to one molecule. Jointly, those weights are known as molecular parameters, and they resemble to the hydrogen and carbon atoms of a hydrocarbon molecule in nature.

$$\varphi(x,k) = \sum_{r=1}^{n} \sigma_r \sum_{i=1}^{k \leq 4} H_{ir} x_r^i \tag{1}$$

Molecules are arranged in groups namely compounds. The latter are structures that represent nonlinearities in molecules. Those compounds are associated with a functional behavior as expressed in (2), where m is the number of molecules in the compound and Σ_j is a partition of the input x such that $\Sigma_j = \{x | \arg\min_j (x - \mu_j) = j\}$, and $\mu_j \in \mathbb{R}^n$ is the center of the jth molecule [20]. In fact, $\Sigma_{j_1} \cap \Sigma_{j_2} = \emptyset$ if $j_1 \neq j_2$. The compound behavior written in (2) is known as linear chain of m molecules since it is similar to organic chains in chemical nature [19].

$$\psi(x) = \begin{cases} \varphi_1(x,3) & x \in \Sigma_1 \\ \varphi_2(x,2) & x \in \Sigma_2 \\ \cdots & \cdots \\ \varphi_{m-1}(x,2) & x \in \Sigma_{m-1} \\ \varphi_m(x,3) & x \in \Sigma_m \end{cases} \tag{2}$$

Compounds can interact among them in definite ratios α_t, namely stoichiometric coefficients or simply weights, forming a mixture $S(x)$. It is represented as shown in (3); where, c is the number of compounds in the mixture and α_t is the weighted factor of the t-th compound [19].

$$S(x) = \sum_{t=1}^{c} \alpha_t \psi_t(x) \tag{3}$$

Literature has reported different training algorithms for AHN. They differ in terms of how to approach the learning process of the molecular parameters and the centers of molecules. For example, the simplest and original method [19] implements the least square estimates (LSE) to learn the molecular parameters and the gradient descent to learn the centers of molecules. In [21] the authors implement the Moore-Penrose pseudo-inverse to find the molecular parameters and particle swarm optimization to learn the centers of molecules. Recently, authors in [20] implemented the stochastic parallel extreme (SPE-AHN) training algorithm that is a fast and reliable method based on the latter training method, but running parallel processing and stochastic learning.

In this work, we adopted SPE-AHN as the training algorithm for AHN. It requires two hyper-parameters to set up the training procedure: the number of molecules in a compound ($m \geq 2$), and the batch size ($0 < \beta \leq 1$) that corresponds to the percentage of input data computed at each iteration of the algorithm. In a nutshell, SPE-AHN works dividing the training phase into two hierarchical steps. On one hand, the algorithm seeks for the centers of molecules μ_j for $j = 1, \ldots, m$ using PSO. These values are encoded in the individual of the PSO. On the other hand, the molecular parameters H_i and σ are learned using extreme learning machines (ELM) through the Moore-Penrose pseudo-inverse. This updating of the molecular parameters is done in one-shot each time an individual of the PSO is evaluated into the objective function. It is noticeable that the whole procedure of tuning the parameters of the AHN-model might be time-consuming. To overcome this issue, SPE-AHN implements parallel processing during the objective function evaluation of the individuals. In addition, each individual only uses a random, possibly small enough, subset of the training data (set up by the batch size β) to estimate the evaluation of the objective function. To this end, SPE-AHN has reported to be 10 thousand times faster than the original training algorithm, without decreasing the predictive power of the AHN-model [20]. Also, SPE-AHN has been proved to be efficient during the training procedure while using high-dimensional and big data [20].

B. Overview of CNN

Convolutional Neural Network (CNN) is one of the most commonly used techniques in image classification, speech recognition, sentence modeling and lately in wearable sensors based human activity recognition [5]. CNNs are Deep Neural Networks with interconnected structures [6]. CNN has "the ability of multi-layer networks trained with gradient descent to learn complex, high-dimensional, non-linear mappings from large collections of examples" [22]. A CNN model normally consists of convolutional layer, pooling layer and fully-connected layers which perform classification or regression tasks. These multi-layer networks can extract automatic features from raw sensor signals [8] [4]. CNN has the advantages of local dependency and scale invariance when applied time series classification [4]. The convolution operation effectively exploits the local dependency of time-series [15] of wearable sensor signals used for HAR which are probably correlated. People usually perform the same activity differently, so scale invariance of CNN is well suited for HAR. Most approaches combine the convolutional layer with the pooling layer performing max or average pooling, although other strategies like stochastic pooling and spatial pooling units are also used [23]. After convolution, the pooling layer can reduce sensitivity of the output [10], avoid over fitting and speed up the training process [4]. The fully connected neural networks combine local structures in the lower layers with an inference engine for instance SoftMax. Weight sharing is a used strategy to reduce the complexity of the network [24].

IV. Comparative Analysis

This sections describes the comparative analysis between AHN and CNN applied on HAR. First, we introduce the factors to be studied and how to be measured them. Then, we present the case study scenario.

TABLE I
SETUP OF THE HYPER-PARAMETERS IN THE COMPARATIVE ANALYSIS. FOR ALL THE CASES, THE INPUT SIZE IS 50% OF THE TOTAL TRAINING DATA. RANGES ARE MARKED AS LOWER_BOUND:STEP:UPPER_BOUND.

Hyper-parameter	Range	Median
AHN		
molecules	$2:1:20$	11
batch size	$10:10:100$ (%)	50%
CNN		
section depth	$1:1:20$	10
initial learning rate	$1E-6:0.01:0.01$	0.0055
momentum	$0:0.01:0.1$	0.05

A. Factors

For this comparative analysis, we consider the following aspects to evaluate on the performance of the models: (i) the effect of the hyper-parameters and (ii) the influence of the input size.

1) Hyper-parameters: The training procedure in the models highly depends on the architecture chosen and the initial settings of the training algorithm. Both architecture and settings are configured using the hyper-parameters. In this regard, we choose the next hyper-parameters: the number of molecules and the batch size for AHN; and the number of block layers (section depth, as described below), the initial learning rate and the momentum for CNN.

2) Input Size: Another factor that impacts on the performance of the models is the amount of data for training. In this work, we consider the percentage of the training data (input size) used for tuning the parameters of the models.

B. Metrics

We measure the performance of the models due to the hyper-parameters and the input size through three metrics. The first considers the accuracy of the model, as expressed in (4), where TP, TN, FP and FN represent the true positive, true negative, false positive and false negative values, respectively. The second metric evaluates the training time in seconds that the model spends to perform accurately. The third metric computes the number of parameters required in the proposed architecture.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

C. Experimental Setup

We compute the metrics described above by varying the values of the hyper-parameters in a given range. Table I summarizes the ranges for the hyper-parameters considered in the study. We vary the hyper-parameters one at the time while the other hyper-parameters were setup in the median value, also shown in Table I. We perform a 5-fold cross validation at each iteration within the range of the hyper-parameter to be evaluated. Reported values were the mean accuracy and the mean training time. For illustration purposes, the architectures of the models are depicted in Fig. 1.

In addition, we conduct a Bayesian optimization [25] to find the suitable values of the hyper-parameters in the models. This approach obtains an optimal architecture of the models. Then, we measure the mean accuracy and the mean training time in a 5-fold cross validation using these suitable architectures. In the same architectures, we measure the number of parameters required.

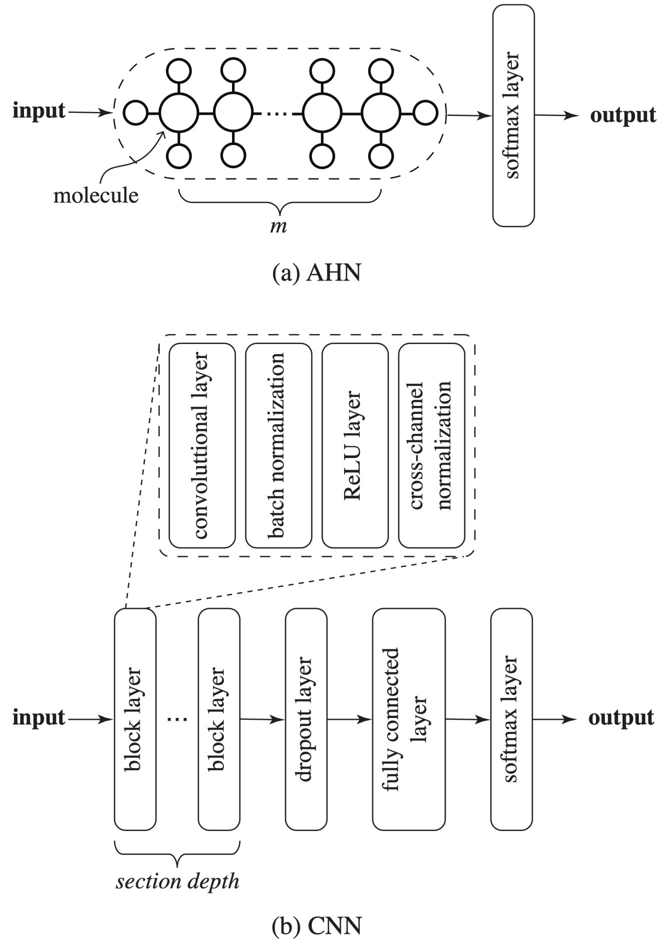

Fig. 1. Architectures of the models implemented in this comparative study: (a) AHN showing m molecules in a compound, and (b) CNN showing the block and the other layers.

Lastly, we use the same suitable architectures to measure the impact of the input size. For this experiment, we conduct a 5-fold cross validation and we report the mean accuracy and the mean training time.

All the experiments were done using a personal computer Dell XPS 13 with Intel Core i7-8550U processor at 4.0GHz and 4 cores, and 8GB in RAM. No GPUs were used.

D. Case Study on Human Activity Recognition

For this comparative analysis, we used the public PAMAP2 dataset [13] for human activity recognition. It consists of sensor signal recordings from nine subjects carrying out locomotion activities. These twelve classes are lying, sitting, standing, walking, running, cycling, nordic walking, ascending stairs, descending stairs, vacuum cleaning, and rope jumping. For this work, we use 51 raw signals coming from the sen-

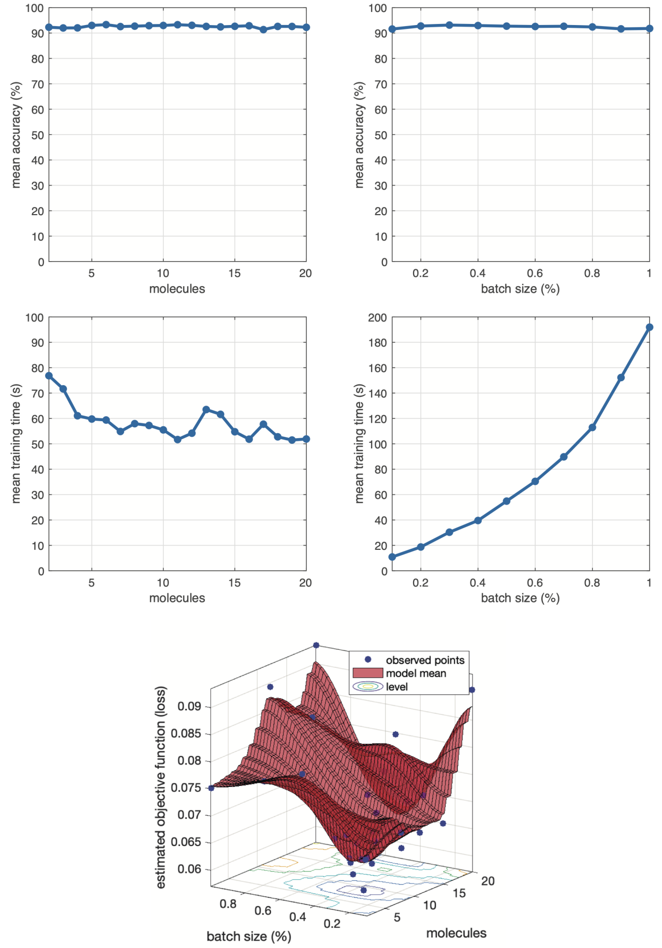

Fig. 2. Performance of AHN in the case study.

sors: accelerometers, gyroscopes, and magnetometers. Other measurements stored in the dataset were not be considered.

We adopted the same procedure to develop a human activity recognition model, as in [26]. To do so, we pre-processed the raw signals using a sliding window approach with a window size of 3 seconds and overlapping of 50%, as in [26].

For the CNN, we also implemented the architecture suggested in [26]. This architecture is built using an input layer that receives the 51 3-second signal segments, a block layer, a dropout layer of 50%. Then, it inputs to a fully-connected layer with softmax function to classify the 12 different activities. Particularly, the block layer comprises a convolutional layer with filter size of 3×3, a batch normalization layer with a rectified linear units (ReLU) layer, and a cross-channel normalization layer of size 5 to independently treat sensor signals over time. Moreover, we increase the number of block layer by setting this up with the hyper-parameter section depth. Depending on this value, the number block layers were added accordingly. This procedure enlarges the architecture of the CNN for evaluating the complexity of the model in those terms.

TABLE II
COMPARISON OF FOOTPRINT IN MODELS.

Model	Memory (KB)	Learnable parameters (units)
AHN	53	$6,110$
CNN	40	$9,004$

V. RESULTS AND DISCUSSION

This section reports the results of the comparative analysis. Three experiments were done: the first considers the evaluation of the performance in terms of the hyper-parameters. The second experiment considers to optimize the architecture of the models using Bayesian optimization and to evaluate the performance of the models. Lastly, the third experiment considers to measure the performance of the models due to the input size.

First, we measured the performance of the AHN and CNN models depending on the variation of the hyper-parameters. Figure 2 shows the mean accuracy and the mean training time of the AHN model due to the variation in the number of molecules and the batch size (first two-row graphs). As noted, the mean accuracy slightly varies over the different values of the hyper-parameters. But, the mean training time decreases when the number of molecules increases, and the mean training time increases while the batch size value also increases. In the same way, Figure 3 shows the mean accuracy and the mean training time of the CNN model with respect to the variation of the section depth, the initial learning rate and the momentum values (first three-row graphs). It can be seen that the section depth slightly modifies the accuracy of the model, but it highly impacts on the mean training time (quasi-linear). The performance of the CNN reports better mean accuracy while the initial learning rate increases; and the mean training time decreases in the same evolution of the initial learning rate. The momentum value does not report a significant impact on the mean accuracy of the model, and larger values of momentum also decreases the mean training time.

In the second experiment, the architectures of the models were optimized using Bayesian optimization. In Figure 2, the 3D-graph reports the estimated objective function (in terms of the loss rate) as the hyper-parameters change. From this analysis, the best hyper-parameter values are: molecules $= 9$ and batch size $= 35.4\%$. On the other hand, Figure 3 reports two 3D-graphs about this optimization procedure. In this case, the optimal hyper-parameter value found are: section depth $= 4$, initial learning rate $= 0.0099$ and momentum $= 0.0659$. Lastly, it was compared the memory footprint and the learnable parameters required in both models, as depicted in Table II. It can be noticed that AHN requires 32.1% less learnable parameters than CNN, but AHN requires 13 KB in memory more than CNN.

For the third experiment, we trained the models again with the optimal values of the hyper-parameters. Figure 4 shows these results in terms of the mean accuracy and the mean training time. It can be observed that the AHN-model slightly

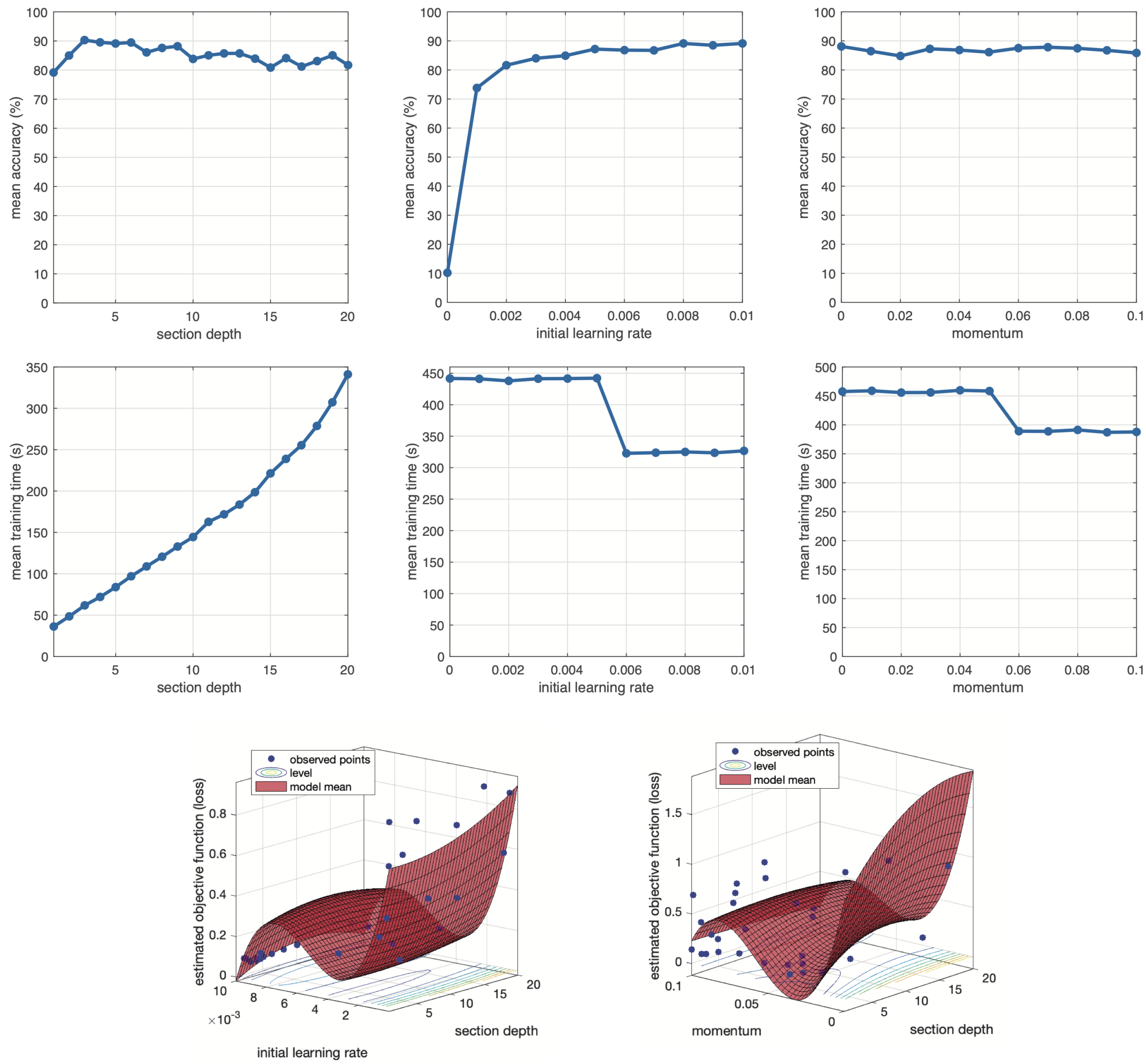

Fig. 3. Performance of CNN in the case study.

overcomes the performance of the CNN-model. The mean accuracy increases while the percentage of input size increases. Moreover, this result gives insights that AHN requires less amount of training data (in addition to the batch size) than CNN. In terms of the mean training time, both optimal models perform similar behavior.

A. Discussion

Through the above experiments, AHN showed better performance in terms of the mean accuracy since it was less variable than the mean accuracy calculated in CNN. On of the advantages of AHN is the ability to learn from smaller size of training data than CNN. These results outperform the implementability of AHN in problem domains with limited data. Another advantage is the robustness of the performance accuracy in terms of the hyper-parameters because it does not require too much effort to set them and obtain high accuracy. Moreover, the number of hyper-parameters in AHN are much less than in CNN. While AHN only has two hyper-parameters,

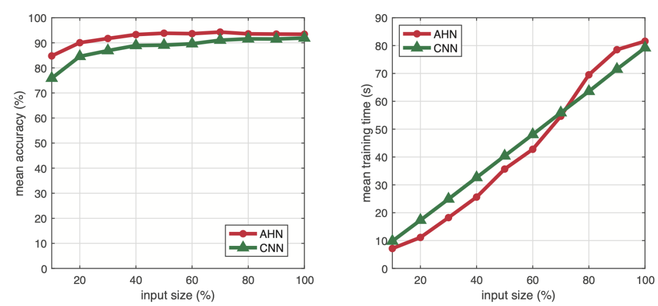

Fig. 4. Performance of the best models of AHN and CNN.

CNN has more than eight different parameters to set, like: number of layers, type of layers, type of activation functions, filter size, number of filters, initial learning rate, momentum, regularization term, among others.

Some limitations found in AHN is the number of learnable parameters that is larger than those required in CNN, and the necessity of extracted features before training in contrast with CNN. Lastly, it is worth noting to say that this comparative analysis was done over a specific case study on human activity recognition, and with a particular public dataset. Thus, a more extendable experimentation and comparison should be done, including other related HAR datasets for validation.

VI. Conclusions

We presented a comparative analysis between AHN and CNN applied on HAR classification problem considering the following aspects to evaluate the performance of the model: (i) the evaluation of the performance in terms of the hyper-parameters (ii) the optimization of the architecture of the models (iii) the assessment of the performance of the models due to the input size. Our results show that AHN architecture is simpler to set up than CNN, it needs less hyper-parameter configuration and shows a slightly better accuracy performance using an optimized architecture. We chose to compare AHN with CNN given that it is a very well suited machine learning technique for HAR.

For future work, we are considering to extend the analysis increasing the number of hyper-parameters in CNN. In addition, it is required to evaluate the performance of both machine learning models in other related HAR datasets and in other problem domains.

References

[1] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4405–4425, 2017.

[2] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[3] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.

[4] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[5] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Systems with Applications*, vol. 105, pp. 233–261, 2018.

[6] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2. IEEE, 2004, pp. II–104.

[7] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[8] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[9] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[10] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *6th International Conference on Mobile Computing, Applications and Services*. IEEE, 2014, pp. 197–205.

[11] H. Ponce, M. D. L. Martínez-Villaseñor, and L. Miralles-Pechuán, "A novel wearable sensor-based human activity recognition approach using artificial hydrocarbon networks," *Sensors*, vol. 16, no. 7, p. 1033, 2016.

[12] H. Ponce, L. Miralles-Pechuán, and M. D. L. Martínez-Villaseñor, "A flexible approach for human activity recognition using artificial hydrocarbon networks," *Sensors*, vol. 16, no. 11, p. 1715, 2016.

[13] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*. IEEE, 2012, pp. 108–109.

[14] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition—a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254, 2018.

[15] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[16] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[17] H. Sagha, S. T. Digumarti, J. d. R. Millán, R. Chavarriaga, A. Calatroni, D. Roggen, and G. Tröster, "Benchmarking classification techniques using the opportunity human activity dataset," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2011, pp. 36–40.

[18] H. Ponce and P. Ponce, "Artificial organic networks," in *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*. IEEE, 2011, pp. 29–34.

[19] H. Ponce, P. Ponce, and A. Molina, *Artificial Organic Networks: Artificial Intelligence Based on Carbon Networks*, ser. Studies in Computational Intelligence. Springer, 2014, vol. 521.

[20] H. Ponce, P. V. de Campos Souza, A. J. Guimarães, and G. González-Mora, "Stochastic parallel extreme artificial hydrocarbon networks: An implementation for fast and robust supervised machine learning in high-dimensional data," *Engineering Applications of Artificial Intelligence*, vol. 89, p. 103427, 2020.

[21] H. Ponce, G. González-Mora, E. Morales-Olvera, and P. Souza, "Development of fast and reliable nature-inspired computing for supervised learning in high-dimensional data," in *Nature Inspired Computing for Data Science*. Springer, 2020, pp. 109–138.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[23] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[24] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Human activity recognition with inertial sensors using a deep learning approach," in *2016 IEEE SENSORS*. IEEE, 2016, pp. 1–3.

[25] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[26] F. Moya Rueda, R. Grzeszick, G. A. Fink, S. Feldhorst, and M. Ten Hompel, "Convolutional neural networks for human activity recognition using body-worn sensors," in *Informatics*, vol. 5, no. 2. Multidisciplinary Digital Publishing Institute, 2018, p. 26.