Gait Recognition using FMCW Radar and Temporal Convolutional Deep Neural Networks

Pia Addabbo
"Giustino Fortunato" University

Mario Luca Bernardi
University of Sannio

Filippo Biondi
University of L'Aquila

Marta Cimitile
Unitelma Sapienza University

Carmine Clemente
University of Strathclyde

Danilo Orlando
"Niccolò Cusano" University

***Abstract*—The capability of human identification in specific scenarios and in a quickly and accurately manner, is a critical aspect in various surveillance applications. In particular, in this context, classical survaillance systems are based on videocameras, requiring high computational/storing resources, which are very sensitive to light and weather conditions. In this paper, an efficient classifier based on deep learning is used for the purpose of identifying individuals features by resorting to the micro-Doppler data extracted from low-power frequency-modulated continuous-wave radar measurements. Results obtained through the application of a deep temporal convolutional neural networks confirms the applicability of deep learning to the problem at hand. Best obtained identification accuracy is 0.949 with an F-measure of 0.88 using a temporal window of four seconds.**

***Index Terms*—Deep Learning, Gait Recognition, Low-power radar, Micro-Doppler**

I. Introduction

Identifying an individual, in a quickly and accurately manner, is a critical aspect in the surveillance context. While conventional systems based on video processing suffer from the main limitations of high computational/storing resources and from the inability to work in all light and weather conditions, radar devices are able to record useful data unaffected by environmental conditions and, even more importantly, to see through the walls. Low cost and low power devices represent a key solution for future application in the surveillance context. Infact, low power frequency-modulated continuous-wave (FMCW) radar algorithms for surveillance applications were designed and addressed in recent studies [1]–[3].

The analysis of radar micro-Doppler, introduced in [4], [5], demonstrated the potential of Doppler information generated by movements of parts of the target for the classification of the latter and micro-motion analysis. Infact, a plethora of studies have been conducted on micro-Doppler analysis and investigation confirm the classification capabilities, able of describing and identifying uniquely features of the targets [6]–[9].

The cutting-edge approaches in target classification are based on the adoption of Deep Learning (DL) algorithms. DL extends classical machine learning by adding more complexity into the model as well as transforming the data using various functions that allow their representation in a hierarchical way, through several levels of abstraction composed of various artificial perceptrons [10]. Indeed, DL is inspired by the way information is processed in biological nervous systems and their neurons. In particular, DL approaches are based on deep neural networks composed of several hidden layers, whose input data are transformed into a slightly more abstract and composite representation step by step. The layers are organized as a hierarchy of concepts, usable for pattern classification, recognition and feature learning. The training of a DL network resembles that one of a typical neural network: i) a forward phase, in which the activation signals of the nodes, usually triggered by non-linear functions in DL, are propagated from the input to the output layer, and ii) a backward phase, where the weights and biases are modified (if necessary) to improve the overall performance of the network. DL is capable to solve complex problems particularly well and fast by employing black-box models that can increase the overall performance (i.e., increase the accuracy or reduce error rate). Because of this, DL is getting more and more widespread, especially in the fields of computer vision, natural language processing, speech recognition, health, audio recognition, social network filtering and moderation, recommender systems and machine translation.

Gait-based human recognition jointly using micro-Doppler features and deep learning is an emerging technology for intelligent surveillance as investigated in [11]–[14]. In this paper, a deep learning framework based on temporal convolutional networks (TCN) is used to identify individuals based on their gait dynamics. TCNs are characterized by casualness in the convolution architecture design and sequence length [15]. This makes them particularly suitable to our context where the causal relationships of the gait signal evolution should be learned. It is worth to highlight that the main contribution of this work is represented by the proposed TCN architecture which is composed by a two-level hierarchical attention layer stack as done in [16] for RNN.

The performance assessment is performed on a large dataset, including several walking session from 5 subjects. The obtained results are promising and showing the effectiveness of the proposed technique.

The remainder of this paper is organized as follows. In Section II, the proposed methodology is presented, whereas in Section III the performance of the proposed method are assessed. Finally, in Section IV the conclusions are drawn.

II. The proposed methodology

The whole classification process proposed in this work is depicted in Figure 1. The main steps are data collection using a FCMW radar, dataset generation defining training and test data after pre-processing and classifier architecture.

The first step (Figure 1-(a)) consists in collecting data from the FMCW radar and processing it extracting MD features which are produced by the periodic movement of any structural component of the individual. Particularly, the time-varying frequency characteristics of the micro-Doppler modulation is extracted from radar data by using a high-resolution time-frequency transform, which characterizes the temporal and spectral behavior of the analyzed signal [5].

The next step is the dataset generation (Figure 1-(b)): MD signatures are firstly pre-processed through noise reduction as in [14], and, training and test datasets are defined.

The architecture of the adopted classifier is shown in Figure 1-(c). The convolutional operations in the TCN architecture are discussed in [15]. Specifically, the TCN network exploits a 1D FCN (fully convolutional network) and padding to enforce layer length coherence. The architecture applies causal convolutions to ensure that when evaluating the output at current time t only current and past samples are considered. The dilated convolutions specify a dilation factor d_f among each pair of neighboring filters. The factor d_f grows exponentially with the layer number. If the kernel filter size is k_l, the effective history at the lower layer is $(k_l - 1)d$, still growing exponentially by network depth.

For classification, the last sequential activation of the last layer is exploited since it summarizes the information extracted from the complete sequence in input into a single vector. Since this representation may be too reductive for the intricate relationships (as those present in complex multivariate time-series), a hierarchical attention mechanism across network layers is added inspired by [16] evolving classifiers proposed in [17], [18]. As shown in Figure 1-(c), if the TCN has n hidden layers, $\mathbf{L}_i \in \mathbb{R}^{K \times T}$ is the weights matrix containing the convolutional activations at each layer i (with $i = 1, \ldots, n$) defined as:

$$\mathbf{L}_i = [\mathbf{l}_1^i, ..., \mathbf{l}_T^i], \tag{1}$$

where K is the number of filters present in each layer and T is the temporal window length. Hence layer attention weight $\mathbf{m}_i \in \mathbb{R}^{1 \times T}$ can be evaluated as:

$$\mathbf{m}_i = \text{softmax}(\tanh(\mathbf{w}_i^T \mathbf{L}_i)) \tag{2}$$

where $\mathbf{w}_i \in \mathbb{R}^{K \times 1}$ are trainable parameter vectors. The combination of convolutional activations for layer i is calculated as $\mathbf{a_i} = f(\mathbf{L}_i \boldsymbol{\beta}_i^T)$ where $\mathbf{a}_i \in \mathbb{R}^{K \times 1}$ and $f(\cdot)$ is an activation function (in this work, ReLU, Mish and Swish are here used [19]) and $\boldsymbol{\beta}_i$ are the weights of the attention layer i. At the output of the hidden-level attention layers, the convolutional activations $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_i, ..., \mathbf{a}_n]$ (with $\mathbf{A} \in \mathbb{R}^{K \times n}$) are used to calculate the last sequence representation to perform the final classification:

$$\boldsymbol{\alpha} = \text{softmax}(\tanh(\boldsymbol{\omega}^T \mathbf{A})) \tag{3}$$

$$\mathbf{y} = f(\mathbf{A}\boldsymbol{\alpha}^T) \tag{4}$$

where $\boldsymbol{\omega} \in \mathbb{R}^{K \times 1}$ is the vector of weights for the high-level attention layer, $\boldsymbol{\alpha} \in \mathbb{R}^{1 \times K}$, is the output of the high-level attention layer, and $\mathbf{y} \in \mathbb{R}^{K \times 1}$ is the final output of the neural network.

The considered architecture can be instantiated with a variable number of hidden layers where each hidden layer is the same length as the input layer. As shown in Figure 1-(c), the following three types of layers are exploited:

- **Input layer**: it represents the entry point of the considered neural network, and it is composed of a node for each set of features considered at a given time;
- **Hidden layers**: they are made of artificial neurons, the so-called "perceptrons". The output of each neuron is computed as a weighted sum of its inputs and passed through an activation function (i.e., mish, swish, and ReLu) or a soft-plus function.
- **Attention layers**: allows modeling of relationships regardless of their distance in both the input and output sequences.
- **Batch Normalization**: Batch normalization is added to improve the training of deep feed-forward neural networks as discussed in [20].
- **Output layer**: this layer produces the requested output.

The TCN training is performed by defining a set of labeled sequences (W, l), where each of the W rows is an instance associated with a binary label l, which specifies the target as exemplified in Figure 1-(c). For each of the W instances, the process computes a feature vector submitted to the classifier in the training phase. In order to perform validation during the training step, 10-fold cross-validation is used [21]. The trained classifier is assessed using the real data contained in the test set made of walking sessions that the classifier has never seen.

During the training step, different parameters of the architecture are tested (i.e., number of layers, batch size, optimization algorithm, and activation functions) in order to achieve the best possible performance as further detailed in the next section.

The considered TCNs architecture was trained by using cross-entropy [22] as a loss function, whose optimization is achieved by means of stochastic gradient descent (SGD) technique. Specifically, a momentum of 0.09 is adopted and a fixed decay of $1e^{-6}$. To improve learning performances, SGD has been configured into all experiments with Nesterov accelerated gradient (NAG) correction to avoid excessive changes in the parameter space, as specified in [23].

III. Experiment

A. Experimental settings and dataset

In this paper a publicly available dataset[1], exploiting a FMCW radar with a center frequency of 77GHz to record

[1] See https://www.imec-int.com/IDRad

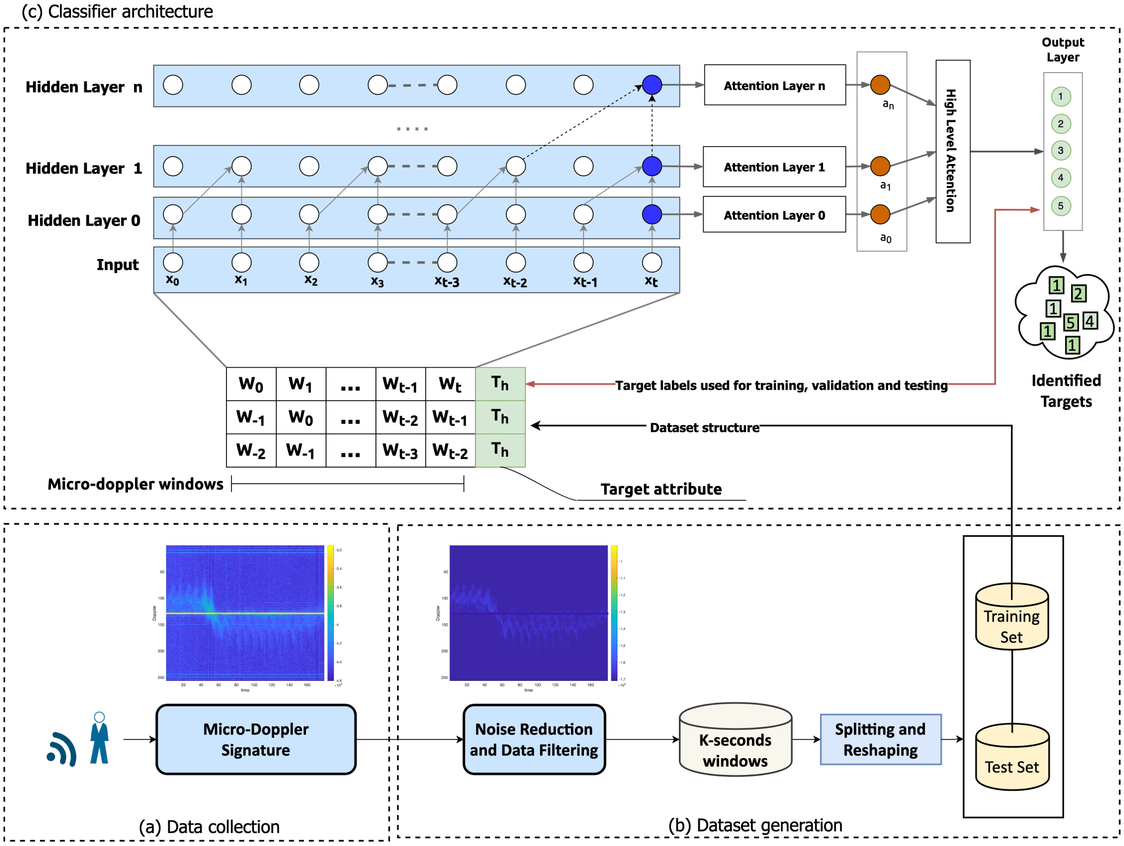

Fig. 1. Overall process and classifier architecture.

TABLE I
HYPER-PARAMETERS OPTIMIZATION AND SELECTED RANGES.

Hyperparameters	Ranges
Activation function	ReLU, Swish, Mish
Network size	Small, Medium
Learning rate	[0.09, 0.12]
Number of layers	{ 6, 7, 8, 9 }
Batch size	{ 64, 128, 256 }
Optimization algorithm	SGD, Nadam, RMSprop

the data, has been used for assessment. The data set consists in a total of 150 minutes of measured micro-Doppler data recordered over five targets and two different rooms. Three datasets are conceived as training, validation, and test sets and consist in 100, 25, and 25 minutes, respectively. All the subjects are males between 23 and 32 years old with comparable postures with a weight ranging from 60 kg to 99 kg and a height from 178 cm to 185 cm.

Given that the FMCW radar captures the range-Doppler maps with an average speed of 15 FPS, the training set contains 95.650 frames, while the validation and test set contain 22.535 frames each. One frame represents one time step in the MD signature and is depicted by 256 Doppler channels (i.e., the sum over all range channels per Doppler channel of one range-Doppler map). The MD signal is re-organized into windows with a length of 45 frames (representing 3s of data) with an overlap of 1s, for both the validation and test set.

The assessment is conducted by identifying the best parameters reported in Table I found using a Sequential Bayesian Model-based Optimization (SBMO) approach implemented exploiting the Tree Parzen Estimator (TPE) algorithm as defined in [24]. As the table summarizes, the following ranges were considered:

- **Network size**: three levels of network sizes (small and medium) are used, depending on the actual number of layers. A small sized network contains a maximum of 1.5 mln of learning parameters whereas a medium network has a number of parameters in the range [1.5 mln, 7 mln];
- **Activation function**: the widely adopted ReLU activation function is used, but the performances of two activations function that have been recently proposed are also investigated and show good results (i.e., Swish and Mish) [19], [25]. It is well known that ReLU suffers from the ”dead” units problem: during training some ReLU units always output the same value for any input. This happens by learning a large negative bias term for its weights during training and also means that it takes no role in discriminating between inputs. When a ReLU unit ends up in this state, it is very unlikely to be subsequently recovered (because the function gradient at 0 is still 0 meaning that SGD will not alter the weights).

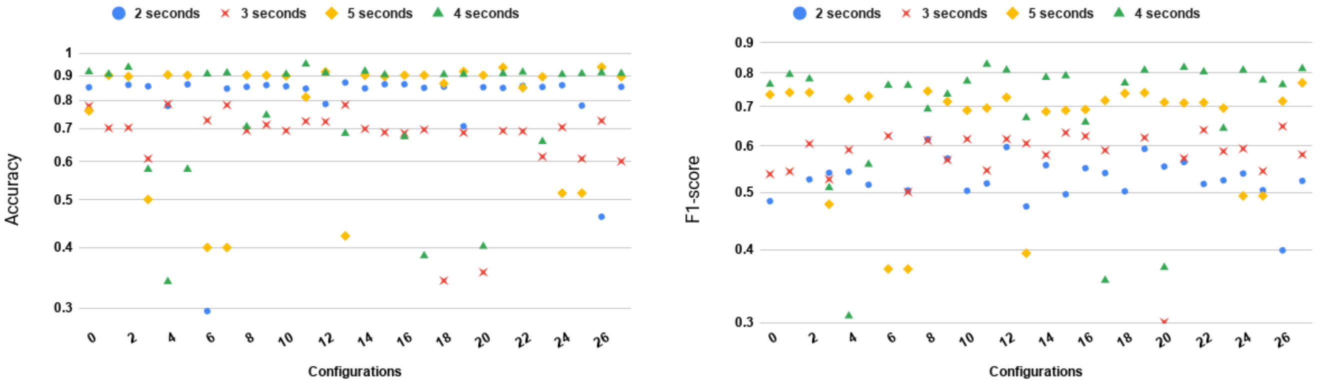

Fig. 2. Hyper-parameters optimization by observation window size (from 2 to 5 sec): each configuration is a choice of the parameters as defined in Table I.

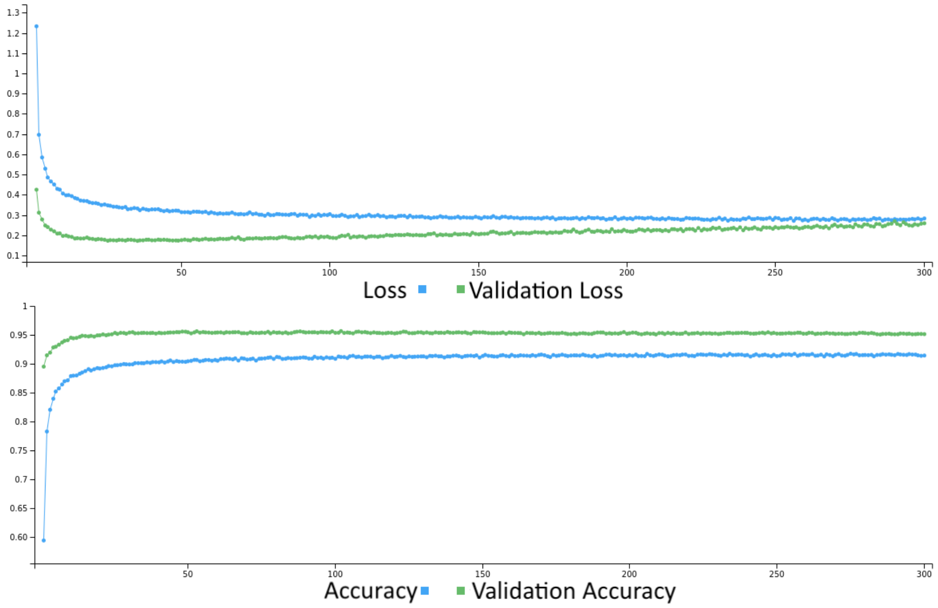

Fig. 3. Accuracy, Loss for training and validation of the best configuration.

There are variants, like ”Leaky” ReLU, with a small positive gradient for negative inputs, that are an attempt to address this issue and give a chance to recover. For our comparison Swish and Mish are choosen since they both do not suffer from the dead neurons issue and deal better with the vanishing gradient problem.

- **Learning rate**: it ranged from 5 to 15, normalized with respect to the optimization algorithm. For instance, using the SGD optimizer, the range was from 0.005 to 0.15 widely considered in literature;
- **Number of layers**: the numbers of considered layers varied from 6 to 9 (considering the amount of data in the dataset);
- **Batch size**: since batch sizes greater than 512 make the training process less stable compromising the final accuracy, three standard and well adopted batch sizes (128, 256, and 512) are compared;
- **Optimization algorithm**: several optimization algorithms are tested to minimize the loss, such as the Stochastic Gradient Descent (SGD) [26], RmsProp [27], Nadam [27]. In particular, SGD has been integrated in all experimentations with Nesterov Accelerated Gradient (NAG) correction to avoid excessive changes in the parameter space, as specified in [23].

PyTorch 1.4 deep learning framework was used to implement the neural network classifier trained on machine with two Intel (R) Core (TM) i9 CPU 4.30 GHZ, 64GB of RAM and four Nvidia Titan Xp.

B. Results and Discussion

Figure 2 shows results in terms of two chosen performance metrics in indentifying subjects: (a) accuracy and (b) F-measure over the hyper-parameters choice (i.e., configurations) as specified in Table I. Most of the network models behave consistently and there are quite small differences among networks with six and seven layers in identifying the five subjects. It is also interesting to observe that there is a small set of network models that are not able to learn from the MD signatures which fall into two categories: (*i*) models with more than nine layers and medium sizes; (*ii*) models trained with learning rates higher than 0.015. For the first case, increasing the MD signature dataset and improving the network architecture could be needed to achieve convergence and to improve the classifier performance. As figure shows, the best result for accuracy is 0.949 with an F-measure of 0.88 using a temporal window of four seconds. The corresponding configuration uses Mish as activation function, a batch size of 64, eight hidden layers with a medium network size and was trained by SGD using 0.1 as learning rate. It is also interesting to look at performance metrics with respect to the temporal observation window size. The performance increases for both accuracy and F-measure until four seconds (that represents best window size) and from five seconds starts to get worse. In Figure 3, the trend of both accuracy and loss for both training and validation sets is reported. It is possible to notice that both the accuracy and loss reach stable values starting from 50 epochs.

IV. Conclusions

This paper proposes the application of an efficient classifier based on a deep temporal convolutional neural networks of MD gait features measured by a low-cost low-power FMCW radar. The experiments have been evaluated processing real data consisting of a long temporal series of FMCW measurements. The high identification accuracy confirms the effectiveness of the proposed method. However since the adopted dataset does not allow to perform a controlled experiment evaluating the impact of several key variables (e.g., subjects' physical characteristics, gender, clothes are note taken into account) a wider experiment is surely desiderable.

References

[1] S. Saponara and B. Neri, "Radar sensor signal acquisition and multidimensional fft processing for surveillance applications in transport systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 4, pp. 604–615, April 2017.

[2] B.-s. Kim, Y. Jin, S. Kim, and J. Lee, "A low-complexity fmcw surveillance radar algorithm using two random beat signals," *Sensors*, vol. 19, no. 3, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/3/608

[3] S. Björklund, T. Johansson, and H. Petersson, "Evaluation of a micro-doppler classification method on mm-wave data," in *2012 IEEE Radar Conference*, May 2012, pp. 0934–0939.

[4] V. C. Chen, F. Li, S. . Ho, and H. Wechsler, "Micro-doppler effect in radar: phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 2–21, Jan 2006.

[5] V. Chen, *The Micro-doppler Effect in Radar, 2nd Edition*. Artech House, 2019.

[6] C. Clemente, L. Pallotta, A. De Maio, J. J. Soraghan, and A. Farina, "A novel algorithm for radar classification based on doppler characteristics exploiting orthogonal pseudo-zernike polynomials," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 417–430, January 2015.

[7] L. Du, L. Li, B. Wang, and J. Xiao, "Micro-doppler feature extraction based on time-frequency spectrogram for ground moving targets classification with low-resolution radar," *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3756–3763, May 2016.

[8] X. Bai and F. Zhou, "Radar imaging of micromotion targets from corrupted data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, pp. 2789–2802, December 2016.

[9] P. Addabbo, C. Clemente, and S. L. Ullo, "Fourier independent component analysis of radar micro-doppler features," in *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, June 2017, pp. 45–49.

[10] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[11] H. T. Le, S. L. Phung, and A. Bouzerdoum, "Human gait recognition with micro-doppler radar and deep autoencoder," in *2018 24th International Conference on Pattern Recognition (ICPR)*, Aug 2018, pp. 3347–3352.

[12] G. Garreau, C. M. Andreou, A. G. Andreou, J. Georgiou, S. Dura-Bernal, T. Wennekers, and S. Denham, "Gait-based person and gender recognition using micro-doppler signatures," in *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Nov 2011, pp. 444–447.

[13] P. Cao, W. Xia, M. Ye, J. Zhang, and J. Zhou, "Radar-id: human identification based on radar micro-doppler signatures using deep convolutional neural networks," *IET Radar, Sonar Navigation*, vol. 12, no. 7, pp. 729–734, 2018.

[14] B. Vandersmissen, N. Knudde, A. Jalalvand, I. Couckuyt, A. Bourdoux, W. De Neve, and T. Dhaene, "Indoor person identification using a low-power fmcw radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 7, pp. 3941–3952, July 2018.

[15] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018. [Online]. Available: http://arxiv.org/abs/1803.01271

[16] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1480–1489. [Online]. Available: https://www.aclweb.org/anthology/N16-1174

[17] M. Bernardi, M. Cimitile, F. Martinelli, and F. Mercaldo, "Driver and path detection through time-series classification," *Journal of Advanced Transportation*, vol. 2018, 2018.

[18] M. L. Bernardi, M. Cimitile, F. Martinelli, and F. Mercaldo, "Keystroke analysis for user identification using deep neural networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8.

[19] D. Misra, "Mish: A self regularized non-monotonic neural activation function," arXiv pre-print, 2019.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 448–456. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045167

[21] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Roy. Stat. Soc.*, vol. 36, pp. 111–147, 1974.

[22] S. Mannor, D. Peleg, and R. Rubinstein, "The cross entropy method for classification," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 561–568.

[23] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML'13. JMLR.org, 2013, pp. III–1139–III–1147.

[24] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11. Red Hook, NY, USA: Curran Associates Inc., 2011, p. 2546–2554.

[25] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *CoRR*, vol. abs/1710.05941, 2017. [Online]. Available: http://arxiv.org/abs/1710.05941

[26] T. Schaul, I. Antonoglou, and D. Silver, "Unit tests for stochastic optimization," 2013.

[27] Y. Wang, J. Liu, J. Mišić, V. B. Mišić, S. Lv, and X. Chang, "Assessing optimizer impact on dnn model sensitivity to adversarial examples," *IEEE Access*, vol. 7, pp. 152 766–152 776, 2019.