Deep Learning for Cognitive Load Monitoring: A Comparative Evaluation

Andrea Salfinger
Department of Cooperative Information Systems
Johannes Kepler University Linz
Linz, Austria
andrea.salfinger@cis.jku.at

ABSTRACT

The Cognitive Load Monitoring Challenge organized in the UbiTtention 2020 workshop tasked the research community with the problem of inferring a user's cognitive load from physiological measurements recorded by a low-cost wearable. This is challenging due to the subjective nature of these physiological characteristics: In contrast to related problems involving objective measurements of physical phenomena (e.g., Activity Recognition from smartphone sensors), subjects' physiological response patterns under cognitive load may be highly individual, i.e., expose significant inter-subject variance. However, models trained on datasets compiled in laboratory settings should also deliver accurate classifications when applied to measurements from novel subjects. In this work, we study the applicability of established Deep Learning models for time series classification on this challenging problem. We examine different kinds of data normalization and investigate a variant of data augmentation.

KEYWORDS

Cognitive Load; Deep Learning; Time Series; Wearable Sensing

ACM Reference Format:
Andrea Salfinger. 2020. Deep Learning for Cognitive Load Monitoring: A Comparative Evaluation. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC '20 Adjunct), September 12–16, 2020, Virtual Event, Mexico.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3410530.3414433

1 INTRODUCTION

Motivation. Inferring a user's current *cognitive load* by interpreting physiological measurements sensed non-invasively from her body (e.g., heart rate or skin response measurements) offers many promising applications to enhance our interaction with our steadily increasing number of technical devices, such as smartphones and smart watches. For example, this would allow equipping our smartphones with notification management capabilities so that we do not get interrupted while performing cognitively demanding tasks. Especially low-cost wearables (such as smart wristbands) with widespread adoption thus offer promising potential for implementing this goal [7]. However, to realize this vision, we require reliable classification models capable of inferring a user's current cognitive load from such superficial surrogate measurements. To tackle this goal, the *Cognitive Load Monitoring Challenge* has been initiated [17], which promotes a labeled dataset for cognitive load inference from measurements obtained with a Microsoft Band 2 [6]. This dataset comprises the recordings of several subjects participating in an experimental setting measuring their physiological responses to the two experimental conditions of experiencing *cognitive load* vs. *resting*. The resulting time series measurements of the four physiological variables monitored with the wristband have been split into time windows of a length of 30 seconds each, and annotated with a unique identifier (ID) associating each record to the subject the measurements have been obtained from. In the training dataset, each time window has been labeled with the associated experimental condition, i.e., the subject's underlying cognitive state (*cognitive load* vs. *resting*), which should be used for developing a classification model that achieves the best-possible accuracy on predicting the withheld labels of the test set.

Challenges. The key challenge of this evaluation setup lies in the fact that the measurements in training and test set have been collected from mutually exclusive sets of subjects, mimicking realistic scenarios. As we will examine, the physiological responses to experiencing cognitive load seem to expose highly different patterns across different individuals. Hence, this presumably represents a more difficult learning setting than problems from related areas utilizing measurements of objective physical forces, like Activity Recognition based on smartphone sensors (e.g., using gyroscope and accelerometer) [12]. Consequently, our models need to be capable of *cross-subject transfer learning*, i.e., need to be able to extract patterns from the subjects in the training set that successfully generalize to the novel subjects in the test set.

Contributions. While traditional approaches for time series classification hinge on *feature engineering* and thus often require expert knowledge of the signals to be interpreted, we conjecture that the multivariate nature of the problem may be particularly suited for the automated feature learning capability of *Deep Learning* (DL)-based models. Therefore, the goal of the present study is to probe the applicability of DL-based approaches developed for time series classification (TSC) to this cognitive load monitoring problem. We benchmark several state-of-the-art DL for TSC architectures on

This work has been funded by the Austrian Science Fund (FWF) under grant FWF T961-N31.

UbiComp/ISWC '20 Adjunct, September 12–16, 2020, Virtual Event, Mexico

ACM ISBN 978-1-4503-8076-8/20/09…$15.00
https://doi.org/10.1145/3410530.3414433

this dataset, and investigate the problem of *cross-subject transfer learning* by examining the impacts of two different dataset *normalization* strategies. We also contribute our findings on additional training setups we experimented with, notably a variant of data augmentation (*upsampling*).

Structure of the Paper. In the next section, we discuss the state of the art on related classification and recognition problems. We then present the experimental details of the provided dataset and challenge, before introducing our comparative evaluation and concluding with our main findings.

2 RELATED WORK

The increasingly wide-spread adoption of *ubiquitous continual sensing technology* in the form of smartphones and wearables, paired with the possibility of automatically converting these massive amounts of sensor data into useful predictive models using *machine learning*, has been opening up a plethora of novel applications to assess and exploit a user's current personal and environment context. In this vein, probably most closely related to the problem of cognitive load monitoring with wearables would be the problem of *Activity Recognition* (AR), which seeks to exploit data sensed from a user's smartphone to classify their current activity and has received considerable research interest for almost a decade [12]. On related activity recognition challenges [11], state-of-the-art approaches correspond to complex ensembles of classifiers joining both traditional, feature-engineering based machine learning approaches, as well as more recent DL-based architectures [5]. Interestingly, in [5] it has been analyzed that most models reach rather similar accuracies, and combining them into an ensemble yields a low-percentage increase in accuracy. Whereas it seems reasonable that these approaches to AR may provide valuable starting points for addressing the present cognitive load monitoring problem, we also note considerable differences in the nature of the problem: The employed smartphone sensors rather measure objective physical forces (such as acceleration), whereas our wristband-recorded data comprises physiological measurements that should be correlated to an underlying cognitive state, both of which are presumably highly individual, i.e., may show considerable between-subject variation. Hence, this introduces additional complexity into the classification problem.

Technically, this classification problem corresponds to a binary, *multivariate time series classification* (TSC) problem. Historically, TSC has been studied in statistics (e.g., yielding approaches like ARIMA), as well as "classical" machine learning (ML) approaches, i.e., approaches requiring hand-crafting of discriminative features. Up-to-date approaches of such classical ML usually involve complex ensembles of various individual classifiers, such as the Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) [1, 14, 15], representing the state of the art on most datasets from the UCR/UEA Time Series Dataset Repository [4].

As an alternative to ML approaches requiring "manual" feature engineering, the past decade has been marked by the surge of *Deep Learning* (DL), which denotes neural networks that autonomously extract discriminative feature representations across their multiple layers suitable for solving the classification problem. DL has also been excelling on sequence-based data in the form of specialized architectures such as *Recurrent Neural Networks* (RNNs) and 1-D Convolutional Neural Networks (CNNs) [8, 13, 16], which also have demonstrated competitive results on TSC [10]. Interestingly, DL has focused TSC less than other application domains (like natural language processing), which is presumably due to the rather limited size of classic TSC datasets, whereas DL typically requires large datasets allowing it to successfully generalize given its high number of parameters and thus many degrees of freedom.

Based on the findings in [5, 10], in this work we decided to focus on DL-based approaches to this TSC problem, motivated by their feature learning capabilities and inherent ability to deal with multivariate data, whereas most classical TSC approach require dedicated strategies for incorporating multivariate time series, such as concatenating the individual features' time series, or column ensembling [15]. Rather than spending our efforts on feature engineering and selection, we were interested in examining the preprocessing and training steps that would optimally phrase our prediction problem for the DL architectures examined. Essentially, the question we are pursuing is how well such DL approaches can be applied "out-of-the-box" to this dataset and problem, without incorporating any expert knowledge about the signals at hand nor sophisticated signal processing.

3 BACKGROUND AND PREPROCESSING

In the provided dataset, four different types of physiological measurements have been recorded from subjects performing tasks under two experimental conditions, *cognitive load* (label 0) vs. *resting* (label 1). Using a Microsoft Band 2, *Galvanic skin response* (GSR), *heart rate* (HR), *RR intervals* (RR) and *skin temperature* (Temp) have been recorded, each sampled at 1Hz. Hence, the dataset consists of 30 second-windows of recordings, which thus comprise a sequence of 30 measurements for each variable. For each subject, 50% of the taken samples correspond to the cognitive load condition and 50% to resting, thus corresponding to a binary classification problem on a balanced dataset. Hence, we are dealing with a multivariate TSC problem characterized by homogeneous time (comprising 30 uniformly distributed time steps), four variables and two classes of cognitive load. The particular challenges lie in the rather small size of the dataset, the rather small window size (30 seconds), as well the per-subject effects, which we will examine in the following. The provided training dataset comprises labeled samples obtained from 18 subjects, whereby between 14 and 41 measurements have been collected per subject (on average 35 measurements per subject, which are balanced - i.e., for each experimental condition, roughly the same amount of samples has been acquired), thus yielding a very small training dataset with only 632 samples in total. The test dataset to be classified contains measurements taken from 5 different subjects. Hence, our model should be capable of transferring the discriminative patterns extracted from the training subject distributions to the unseen test subject distributions.

To mimic this evaluation setting in our model selection process, we thus partitioned the provided training set into the following 66%-17%-17% split required for our model development: We split the 18 training subjects into a development test set (*dev-test*) for estimating the models' generalization performance comprising 3 subjects (we randomly selected subjects '8a1ep', 'b7mrd', and '7swyk'), a

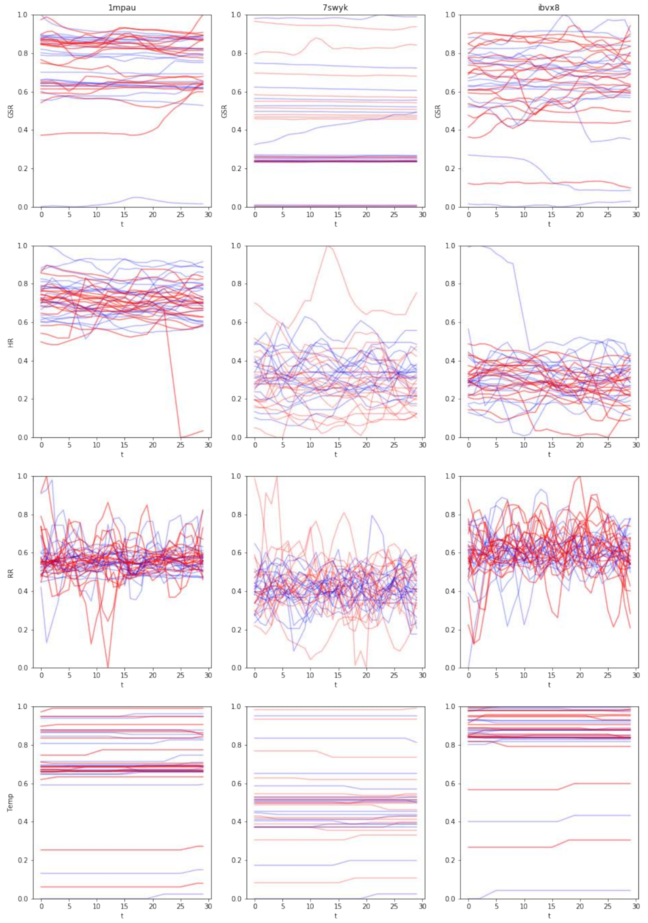


Figure 1: A comparison of three different subjects' (globally normalized) measurement series. Color coding: red: *resting*, blue: *cognitive load*

development validation set (*dev-val*) for parameter tuning comprising 3 subjects (we randomly selected subjects 'yljm5', '5gpsc', and 'f3j25'), and assigned the remaining 12 subjects to our development training set (*dev-train*).

As our physiological measurements are on different scales, we next need to transform each feature into the range [0, 1] to convert our dataset to a numeric range that neural networks can handle well. Since we examined our measurements to be clearly non-Gaussian, we choose to *normalize* our data rather than employ standardization. However, we identified two potential options for normalizing:

– *global normalization*: In our first approach to normalization, we use min-max scaling to convert each time series to the range between zero and one. Instead of directly selecting the minimum and maximum values of *dev-train*, we set larger and lower, physiologically feasible values for minimum and maximum, respectively, to prevent that test subjects may have lower or higher values than previously seen. This *normalization* approach preserves the relative distances between different subjects' measurements, thus allows to assess subject-specific differences on a common scale. Fig. 1 shows the plots for three different subjects' measurement series obtained with this global normalization.

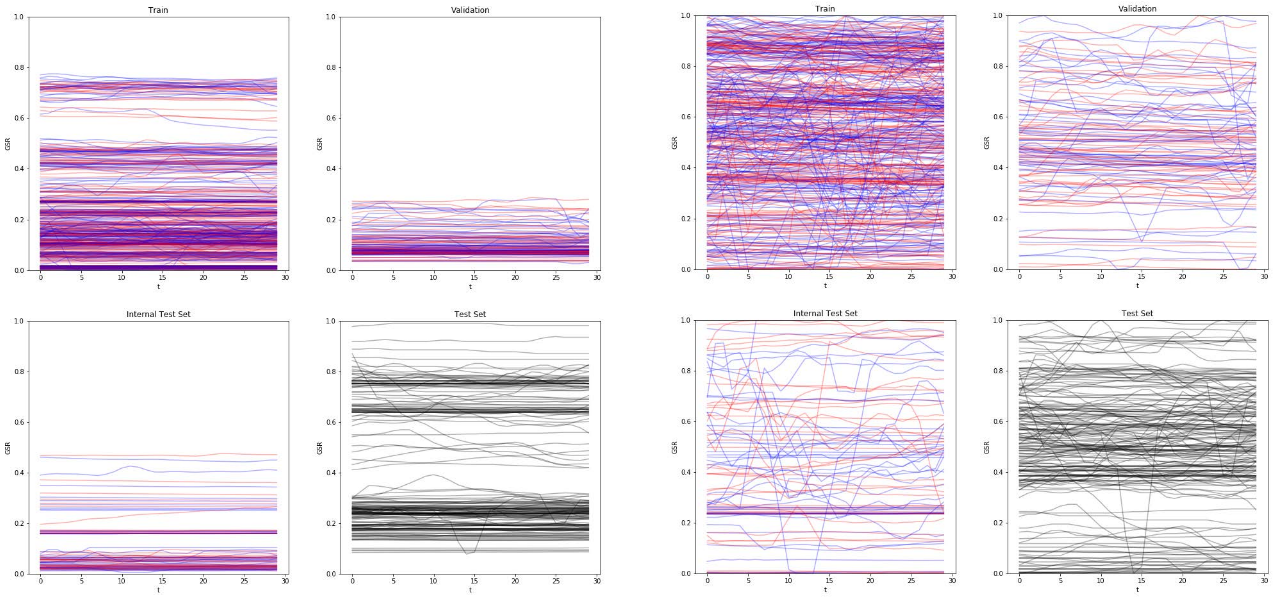
– *per-subject normalization*: As Fig. 1 suggests, there might indeed exist subject-specific patterns. In particular, we observed that different subjects may have different offsets with respect to their measurements' mean values. Since it is unclear whether this may be beneficial for the classification problem, or rather confound the problem, which might depend more on relative variations in the signal irrespective of the actual magnitude, we therefore also investigate the efficacy of using a *per-subject normalization* approach, which performs min-max scaling individually for each subject. This diminishes the effects of the different means of the individual subjects' measurements, thus might help in better discriminating the *relative* differences in the subjects' discriminative patterns (see comparison in Fig. 2).

To increase the size of our rather small dataset and prevent overfitting, we also experimented with *data augmentation* techniques by adding altered versions of the training data: We experimented with *time shifting* by adding variations of the training data where the multi-variate sample sequences have been shifted up to ten time steps back or forth. However, training on this augmented dataset led to deteriorated performance than training on the limited, original data, hence we conclude that this augmentation rather led to confounding the original signal than making the extracted signal more robust. Therefore, in the following we report our results obtained on training on the original, normalized training dataset.

4 MODELS AND COMPARATIVE ANALYSIS

In terms of *accuracy*, the baseline to beat would be 51%, representing the majority class (1) in our *dev-test* set. We also include a conventional ML classifier from the `sktime` package [15] to examine how the DL-based models compare to it, by training a time series forest (TSF) classifier (comprising 200 random forests), which represents an interval-based TSC approach. We use column concatenation to adapt the TSF to our multivariate problem. TSF classifiers provide a highly suited baseline here, as these can directly operate on the *raw* data without requiring normalization, thus allowing to include a classifier which does not depend on a suitable normalization choice.

Next, we employ simple versions of established types of *Recurrent Neural Networks* (RNNs) [8] to get an impression of the overall difficulty of the problem. We start with small architectures using a limited number of hidden units, to account for our small training dataset size. We evaluate several basic configurations comprising RNNs using Gated Recurrent Units (GRUs) [2] as well as RNNs using LSTM cells [9]. As the results indicated roughly comparable performance, we did not attempt at performing a systematic grid search on the optimal network architecture configuration. We train with standard *keras* [3] hyperparameter settings (unless explicitly stated), and train each model on *dev-train* for 1100 epochs (unless explicitly stated). During training, we evaluate the model's accuracy on the hold-out *validation set* (*dev-val*) in each epoch, and use *model checkpointing* to save the best model obtained in this training run. Table 1 reports the results we obtained on *dev-test* after training *n* models for different model configurations in this fashion, which seeks to smooth out the stochastic components of training and



(a) ***global normalization***

(b) ***per-subject normalization***

Figure 2: Comparison of different normalization strategies on the GSR measurements.

thus should reveal the overall model capacities of different model architectures on this problem.

We observed that LSTM-based models generally obtained higher accuracies on the *dev-train* set (> 80%) than GRU-based models, but both achieved similar accuracies on *dev-test*, possibly suggesting that LSTMs tend to overfitting to the training data.

Except for these popular RNNs, we also probed the two top-performing DL approaches for TSC identified in the comparative evaluation conducted in [10]. We based upon their open source implementation offered in the Python package `sktime-dl` to evaluate the architectures Fully Convolutional Network (FCN) and Residual Network (ResNet), also motivated by the superior performance of a ResNet-based approach reported for a similar AR problem in [5]. Technically, these two architectures represent entirely different approaches to the previously evaluated recurrent architectures, as these aggregate temporal information via convolution operations.

Table 1 presents the results obtained with these models, for both the *global* as well as the *per-subject* normalization. As we can see, *global normalization* in general yields better predictive performance. Thus, trying to alleviate the inter-subject differences simply by means of performing normalization on the subject-level seems to be an insufficient approach, which would probably require more sophisticated processing for shifting different subjects' measurements into a common frame of reference. In general, the results across the different classes of models surprisingly appear relatively comparable. ResNet, one of the most complex models tested, achieves close to 100% accuracy on the training data, but interestingly, does not beat simpler classifiers by a considerable margin.

Whereas FCN and ResNet represent quite deep and complex architectures, which require comparatively long training times (hence, we could only evaluate a smaller number of models in Table 1), the evaluated RNNs represent compact models with a small memory footprint, comparatively fast training and competitive results, which represent clear advantages over the convolutional architectures (note that the comparative TSC evaluation conducted in [10] did not include RNN-based models). However, upon inspecting the models' confusion matrices (see Table 2 for an excerpt), it becomes clear that most models, in particular the RNN-based ones (but also the TSF), seem to exhibit a systematic bias towards the *cognitive load* class, by consistently classifying a large portion of samples to this class. This represents a rather surprising finding given the carefully balanced dataset, which comprises an equal share of both cognitive load and resting samples for each subject. Hence, we conclude that apparently, the characteristics of the resting class somehow might be more difficult to learn for the model. We thus hypothesize that by stratifying the data such that resting samples are more frequently shown to the model during training, the optimization problem could be more forced towards considering the resting class. In the simplest setting, we can achieve this effect by simply duplicating all resting samples and reshuffling, which practically means that each resting sample will be shown twice to the model. We denote this artificial replication of samples of one class as "*upsampling*" this class. As shown in the last column block of Table 1, this indeed might lead to a slight increase in accuracy – however, this trick has to be exercised with caution, and needs a careful model selection process based on the test dataset (*dev-test*). Whereas some training runs yield improved and more balanced confusion matrices (see Table 3), others might result in an artificial bias for the resting class, induced by the upsampling of this class. As expected, increasing upsampling by choosing an even higher

Model Type		Global Normalization				Per-subject Normalization				Per-subject Norm. & 2:1 upsampling				Global Norm. & 2:1 upsampling			
	n	μ	σ	min.	max.	μ	σ	min.	max.	μ	σ	min.	max.	μ	σ	min.	max.
Stacked LSTM: H_1 10 LSTM cells, H_2 5 LSTM cells, no regularization	10	0.53	0.04	0.47	0.59	0.49	0.05	0.42	0.58	0.53	0.04	0.48	0.63	0.55	0.05	0.44	0.63
Stacked LSTM: H_1 10 LSTM cells (20% dropout), H_2 5 LSTM cells (20% dropout)	10	0.55	0.05	0.47	0.63	0.50	0.03	0.45	0.57	0.52	0.04	0.45	0.58	0.54	0.05	0.48	0.62
GRU: H_1 (5 GRUs), no regularization	10	0.59	0.04	0.52	0.64	0.55	0.04	0.50	**0.64**	0.52	0.02	0.49	0.56	0.53	0.01	0.51	0.55
Stacked GRUs: H_1 10 GRUs, H_2 5 GRUs, no regularization	10	0.59	0.05	0.50	0.66	0.53	0.05	0.44	0.59	0.51	0.03	0.47	0.56	0.61	0.03	0.57	**0.67**
sktime-dl: FCN (max. 2000 epochs)	3	0.60	0.04	0.55	0.64	0.58	0.02	0.57	0.60	**0.62**	0.03	0.57	**0.65**	0.61	0.03	0.58	0.65
ResNet (max. 300 epochs)	5	**0.64**	0.04	0.56	**0.70**	**0.60**	0.03	0.55	**0.64**	0.59	0.03	0.56	0.64	**0.63**	0.04	0.55	0.66
		No Normalization				No Normalization				No Normalization				No Normalization			
		μ	σ	min.	max.	μ	σ	min.	max.	μ	σ	min.	max.	μ	σ	min.	max.
sktime: TSF (200 random forests)	10	0.56	0.02	0.52	0.60	0.56	0.02	0.52	0.60	0.56	0.02	0.52	0.60	0.56	0.02	0.52	0.60
Baseline:		0.51				0.51				0.51				0.51			

Table 1: Mean (μ), standard deviation (σ), min. and max. of accuracy obtained with different model configurations (averaged over n models trained for each configuration). H_i denotes *hidden layer* i.

replication factor for the resting class definitely deteriorates performance again, as this introduces an artificial bias towards the resting class.

5 CONCLUSION & LESSONS LEARNED

In the present work, we compared the performance of different types of DL-based TSC approaches on a publicly available cognitive load monitoring dataset. As our assessment revealed, most model classes yielded relatively comparable results, whereby we conclude that architectures based on GRUs deliver the most competitive results when factoring in the trade-offs between prediction accuracy, model size and complexity and training times. With careful model selection, their performance is roughly on par with the far more complex ResNets, which presumably can be further tuned by evaluating a wider range of hyperparameter settings and architecture configurations (e.g., identifying the optimal number of hidden neurons and layers). The fact that the best-performing model only achieved an accuracy of 70% suggests that this cognitive load monitoring task represents a considerably hard problem, most likely due to the between-subject variance and limited dataset size, which renders this problem particularly difficult for DL-based approaches. However, our evaluated approaches have shown to be competitive with a classical ML approach tested (a time series forest classifier). We studied the effects of two different *normalization* approaches for preprocessing the data on the resulting accuracy, finding that a *global* normalization yields higher predictive accuracy than a *subject-based* normalization strategy. We further experimented with *upsampling* the samples of the resting class by a factor of two, in order to mitigate the models' inherent bias. Most models evaluated, in particularly the RNN-based ones, exposed a systematic bias towards the cognitive load class on the balanced dataset, possibly suggesting that the underlying characteristics of the resting class seemed to be more difficult to extract. We sought to mitigate this effect by adopting a two-fold "upsampling" strategy for the resting samples, i.e., showing the models the resting samples twice as often during training, which, however, only yields marginal improvements on some type of models only and demands careful model selection to weed out models biased towards the upsampled class. Further increasing the "upsampling" proportion, as to be expected, conversely led to deteriorated results, which changed the dataset characteristics and problem too substantially, thus inducing a wrong bias for the resting class.

We hope that these findings on optimizing the preprocessing for and training of established DL approaches contributes to the understanding of their suitability for cognitive load monitoring. As directions for future work, we note that additional improvements on predictive accuracy could be obtained by creating an ensemble of several classifiers, as well as a more systematic evaluation of different neural architecture configurations.

REFERENCES

[1] Anthony Bagnall et al. 2020. A tale of two toolkits, report the third: on the usage and performance of HIVE-COTE v1.0. http://arxiv.org/pdf/2004.06069v2

[2] Kyunghyun Cho et al. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *CoRR* abs/1406.1078 (2014). http://arxiv.org/abs/1406.1078

[3] François Chollet et al. 2015. Keras.

[4] Hoang Anh Dau, Eamonn Keogh, et al. 2018. The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[5] Martin Gjoreski, Vito Janko, et al. 2020. Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors. *Information Fusion* 62 (2020), 47–62.

[6] Martin Gjoreski, Tine Kolenik, et al. 2020. Datasets for Cognitive Load Inference Using Wearable Sensors and Psychological Traits. *Applied Sciences* 10, 11 (2020), 3843.

[7] Martin Gjoreski, Mitja Luštrek, and Veljko Pejović. 2018. My Watch Says I'm Busy: Inferring Cognitive Load with Low-Cost Wearables. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (UbiComp'18)*. Association for Computing Machinery, New York, NY, USA, 1234–1240.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[10] Hassan Ismail Fawaz et al. 2019. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery* 33, 4 (2019), 917–963.

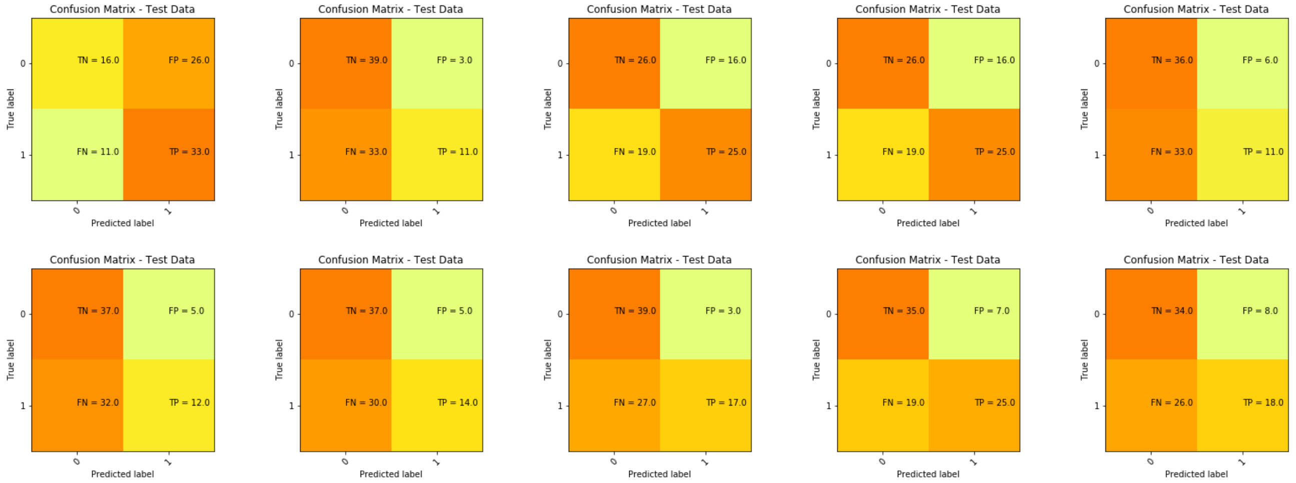


Table 2: Confusion matrices obtained on 10 trained stacked GRUs models, on the globally normalized data without upsampling. As we observe, it is rarely the case that the mass is concentrated along the diagonal (i.e., the plots show the darker colors along the diagonal) – rather, we observe a bias towards one class, most frequently class 0 (cognitive load), indicated by the darker colors in the left column of the confusion matrices.

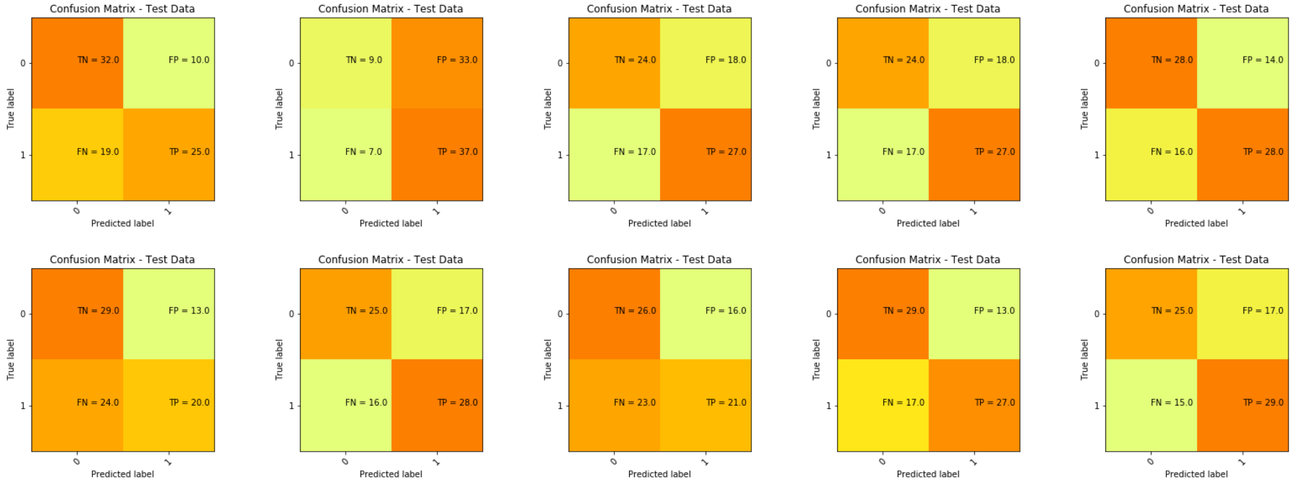


Table 3: Confusion matrices obtained on 10 trained stacked GRUs models, on the globally normalized data with upsampling. As we observe, for the GRU-based models, this upsampling strategy induces a bias towards the resting class in most of the resulting models, indicated by the darker colors in the right column of the confusion matrices, but generally leads to more balanced confusion matrices.

[11] Vito Janko, Mitja Luštrek, et al. 2018. A New Frontier for Activity Recognition - The Sussex-Huawei Locomotion Challenge. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (UbiComp '18)*. Association for Computing Machinery, New York, NY, USA, 1511–1520.

[12] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. 2011. Activity Recognition Using Cell Phone Accelerometers. *SIGKDD Explor. Newsl.* 12, 2 (2011), 74–82. https://doi.org/10.1145/1964897.1964918

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[14] Jason Lines, Sarah Taylor, and Anthony Bagnall. 2016. HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles for Time Series Classification. In *16th IEEE International Conference on Data Mining*. IEEE.

[15] Markus Löning, Anthony Bagnall, et al. 2019. sktime: A Unified Interface for Machine Learning with Time Series. In *Workshop on Systems for ML at NeurIPS 2019*.

[16] Jürgen Schmidhuber. 2015. Deep learning in neural networks: an overview. *Neural networks : the official journal of the International Neural Network Society* 61 (2015), 85–117.

[17] Niels van Berkel, Anja Exler, Martin Gjoreski, Tine Kolenik, Tadashi Okoshi, Veljko Pejovic, Aku Visuri, and Alexandra Voit. 2020. UbiTtention 2020: 5th International Workshop on Smart & Ambient Notification and Attention Management. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC '20 Adjunct)*. to appear.