Time Series Classification with HA-TCN for Stress Levels

Juan Muneton*, Julia Windham*, Jacob Zeldin*
Brown University, Department of Computer Science

*Each author contributed equally to this paper.

I. INTRODUCTION

With the increase of people diagnosed with anxiety and high stress levels, it has become imperative to study the biological factors around this anomaly. Being able to measure stress, therefore, may help to address this problem. Although stress has a psychological origin, it affects several physiological processes in the human body: increased muscle tension in the neck, change in concentration of several hormones and a change in heart rate (HR) and heart rate variability (HRV).

This research project aims at studying heart rate levels by identifying stages of stress across different time intervals in patients wearing heart rate sensor monitors. In doing so, our goal is to re-implement previous scientific work conducted by Lin et al., 2019 on the paper *Medical Time Series Classification with Hierarhical Attention-Based Temporal Convolutional Networks: a Case Study of Myotonic Dystrophy Diagnosis* [1]. While this invention focuses on the interpretable diagnosis of myotonic dystrophy from analysis of handgrip time series data, we propose that this model can also be used for time-series data for measuring stress levels. Therefore, we present an HA-TCN model capable of classifying levels of stress based on heart rate monitoring at different time-steps, further demonstrating that this model has applicability beyond its current use.

In this project, we take on the challenge of developing a temporal convolutional neural network capable of determining stages of increase heart rate levels with an addition of an attention model that further provides the benefit of identifying relevant behavior in different time-steps while decreasing noise in the data and adjusting importance of relevant patterns for the binary classification of stress levels. We show that our model is capable of learning at a fast rate while also providing an accuracy score beyond 90%, letting us infer that this model is another milestone for the inclusion of artificial intelligent models in the health care field.

II. RELATED WORK

A. Attention Mechanisms

Attention mechanisms in deep learning were were initially developed for recurrent neural networks (RNN) on end-to-end machine translation applications [2]. As of today, there has been a growing application of attention models in temporal sequence analysis [1].

Under an RNN model, the attention framework works by combining a sequence of latent vectors that contain the network activations, forming a context vector. This vector is then passed to a feed-forward layer that will be eventually used for classification. The magnitude of the attention weights is proportional to the relevance of the corresponding segment to the classification decision [1]. Attention models have been integrated to LSTMs in natural language processing (NLP). A group of researchers previously used Attention models on LSTM for extraction of important features in dialogue detection [3]. Their work allowed them to conclude that attention layers played a crucial role in selecting important information, thereby improving label accuracy. In another project, Zhou et al. [4] demonstrated that attention layers in a bi-directional LSTM played a crucial role in prediction of time series classification, a crucial milestone and reason for the development of our current project [5, 6]. The addition of hierarchical attention networks (HAN) has been shown to be useful in choosing relevant encoder hidden states in document classification of words [8]. Additional research has also shown that attention models under a dual-stage RNN for time series prediction successfully identified relevant exogenous time series at each time step and salient encoder hidden states across time steps [7].

B. Temporal Convolution Networks

Previous studies have shown that TCN performs better in prediction and classification models over other recurrent networks such as RNNs and LSTMs [9]. A temporal convolution model carries a greater advantage over these counterparts given that they do not pose the problem of vanishing and exploding gradient issues related that could be otherwise found in RNNs. Moreover, the dilated convolutions of the TCN model allow for an exponentially increasing receptive field dependent on the depth of the model. Lastly, TCN is more parallelizable than RNNs or LSTMs, allowing for a lower requirement of computational resources in more complex scenarios.

III. METHODOLOGY

A. Data

For the modeling of this project, we used the WESAD (Wearable Stress and Affect Detection) Data Set. WESAD is a publicly available dataset for wearable stress and affect detection. This multi-modal dataset features physiological and motion data, recorded from both a wrist- and a chest-worn device, of 15 subjects during a lab study. The following sensor modalities are included: blood volume pulse, electrocardiogram, electrodermal activity, electromyogram, respiration, body temperature, and three-axis acceleration. Moreover, the dataset bridges the gap between previous lab studies on stress and emotions, by containing three different affective states (neutral, stress, amusement). In addition, self-reports of the subjects, which were obtained using several established questionnaires, are contained in the dataset [10]. We split the data into training and testing sets. The training split accounted 70% of the data, while testing only 30%. The WESAD database, however, did not contain the binary classification stress level labels we needed for the implementation of this project. We then decided to further our research exploration in studying heart rate levels and created a threshold that defined whether someone was under stress. Based on previous scientific work, a heart rate above 100 beats can represent stages of major motion activity, which is also correlated with the raise of stress levels. Figure 1 provides an example of a sample of a patient's heart rate within a specific interval. The classification is performed if the heart rate is beyond 100.

1. Data under PCA

We implemented feature dimensionality reduction using the Principal Analysis Component to preserve information from features beyond heart rate, such as blood volume pulse, electrocardiogram, electrodermal activity, electromyogram, respiration, body temperature, and three-axis acceleration. We used the SKlearn PCA with a dimension reduction of 1.

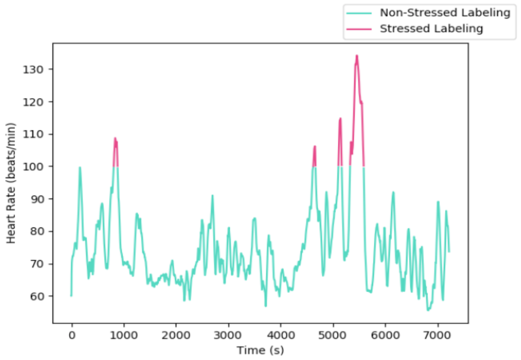

FIG. 1: Sample of a patient's heart rate

2. Temporal Convolutional Neural Networks (TCNs): Model and Training

We use a neural-network based model for stress, in particular, the Temporal Convolutional Network (TCN) [1]. This model is useful because it helps capture temporal dependencies in the historical time series in the WESAD data. More concretely, suppose we are given an input sequence $x_0, ..., x_T$ and want to predict some corresponding outputs $y_0, ..., y_T$, where y (0,1) at each time denoting a binary key of stress or no stress. The key constraint is that, to predict the output y_t for some time t, we can only use the inputs that have previously been observed: $x_0, ..., x_t$. To accomplish this, the TCN model uses **causal** convolutions, convolutions where an output at time t is only convolved with elements from time t and earlier in the previous layer. The second principle that TCN is built off of is that the network produces an output of the same length as the input, and consequently all the convolutional layers follow the 1D fully-convolutional network (FCN) architecture.

A simple causal convolution is limited in terms of the amount of history it can use to generate its outputs. To address this, the TCN uses **dilated convolutions**, where the dilation factor increases exponentially with each layer. In our model, we have 1 input layer, 6 convolution hidden layers, 6 attention layers, and 1 output layer with dilation factors $d = 1, 2, 4$. The filter size k also determines how many elements from the previous layer we use to determine the outputs at the next layer. Our filter size decreases at each convolution, but overall this represents each element depends on exactly on the number of elements each filter contains from the previous layer. By adjusting for the dilation factor and filter size, the outputs at the top level are able to capture a wider range of inputs in TCN.

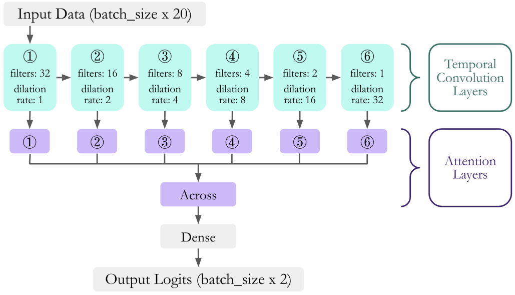
Input Data (batch_size x 20)
① filters: 32 dilation rate: 1
② filters: 16 dilation rate: 2
③ filters: 8 dilation rate: 4
④ filters: 4 dilation rate: 8
⑤ filters: 2 dilation rate: 16
⑥ filters: 1 dilation rate: 32
Temporal Convolution Layers
①
②
③
④
⑤
⑥
Attention Layers
Across
Dense
Output Logits (batch_size x 2)

FIG. 2: The TCN framework

In our implementation of TCN, we couple each convolution layer with an relu Activation layer. Finally, we add a series of Dense layers after the convolution layers to reduce the output to size 2 for the volume prediction. The network parameters are optimized using Adam with learning rate 0.05. Additionally, we split the model training in to two steps, where the first trains the model without regularization and the second with regularization.

3. Attention Layers

When using TCN for classification tasks, the last sequential activation from the deepest layer is typically used. However, this approach often condenses information extracted from the input sequence into one vector, which is problematic because a singular vector may have information in a largely abbreviated scale. Given this problem, we attempted to add a hierarchical attention mechanism across each hidden layer. More specifically, we create a within layer attention stage after each hidden layer. Suppose HA-TCN has k layers, H_i is a matrix consisting of convolutional activations at each layer i, i= 0,1,2,...K; H_i = [h_0, h_1, ... h_T], H_i $\mathbb{R}^{\mathrm{C*T}}$, where C is the number of kernel filters at each later. Then, we move to calculate an alpha weight at each hidden convolution layer α_i $\mathbb{R}^{1*T}$ as follows:

$$\alpha_i = softmax(\tanh(w_i^T H_i))$$

where w_i $\mathbb{R}^{\mathrm{CX1}}$ is a trained parameter vector and H_i represents the *ith* hidden layer, and w_i^T is its transpose . Then, the combination of convolutional activations for layer i is calculated as follows:

$$\gamma_i = ReLU(H_i \alpha_i^T)$$

After executing each within layer attention layer, the convolutional activations are now transformed into a new vector, $M = [\gamma_0, \gamma_1, \gamma_2, ..., \gamma_i, ...\gamma_k]$, where k is the total number of hidden layers. Similarly, the across attention layer takes M as the input to calculate the final sequence representation:

$$\alpha = softmax(\tanh(W^T M))$$

$$\gamma = ReLU(\tanh(M\alpha^T))$$

This final gamma representation will output a dimensionality of *batchsize x* 2 representing the outcome of the logits which then will be used to extract the highest cell indexed value that denotes the binary outocome for prediction in the classification model.

Figure 2 provides a representation of the TCN model with attention layers, where for each hidden layer there is a within layer attention that calculates the α_i's. We combine these scores along with the convolutional activations, and, finally, estimate the attention score. With this approach in mind, we were able to account for the most relevant information in the input data. Attention layers can help reduce the amount of noise and, eventually, improve our overall predictive performance.

IV. RESULTS

The model performs consistently at around 0.955% accuracy with the chosen learning rate of 0.005, window size of 20, batch size of 64, and no additional features besides heart rate. Figure 3 shows the distribution of accuracies between batches of test data, showing that the spread is fairly small.

We performed several experiments to determine differences between these options.

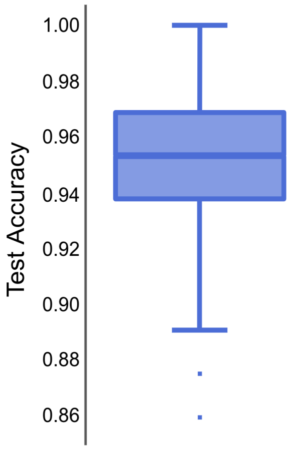

FIG. 3: Test accuracy of model over batches

A. Training & Learning Rates

Table 1 shows the mean and final loss for training using various learning rates, and Figure 4 displays this for each batch. Large learning rates like 0.05 and small learning rates like 0.0005 both had poor mean and final loss, showing that they did not learn well and did not quickly reach a steady loss. Losses in between these two extremes more appropriately started off high and decreased by the end of our single epoch.

TABLE I: Model training loss with different learning rates

Learning Rate	Mean Loss	Final Loss
0.0005	677.824	0.394
0.001	4.667	0.383
0.005	46.165	0.235
0.01	44.155	0.098
0.05	1756.82	0.686

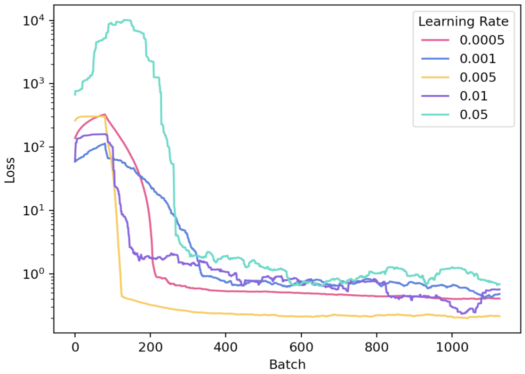

FIG. 4: Training loss with different learning rates

B. Window Size & Model Performance

Table 2 shows the mean and maximum accuracies across batches of test data in models with different window sizes. They are all nearly identical, showing us that there is no particular difference in the way the model performs between window sizes. This is somewhat surprising: we had assumed that with a larger window size, it would be easier to classify.

TABLE II: Model testing accuracy with different window sizes

Window Size	Mean Acc	Max Acc
1	0.955	1.0
2	0.955	1.0
10	0.955	1.0
20	0.955	1.0
30	0.955	1.0
50	0.955	1.0
100	0.954	1.0

C. Number of Features & PCA

We also tested the use of multiple features by comparing the performance of the main heart rate time series data with the use of additional PPG data (another measure of blood flow). Since the time series data needed to have one float per time-step, we used PCA of 1 component to reduce the values of heart rate and PPG to one value for each time-step in the time series for an individual. The labels remained the same: whether or not the next sequential time step has a heart rate reading of greater than 100. Table 3 displays the mean and maximum values of mean test accuracy across 10 models for each of the heart rate data and the PCA heart rate / PPG data. We can see that there does not exist any significant difference with this additional feature, perhaps because it is not well correlated.

TABLE III: Model testing accuracy with and without additional features PCA

Model Type	Mean Acc	Max Acc
Heart Rate Only	0.955	0.955
Heart Rate & PPG (with PCA)	0.955	0.955

V. CHALLENGES

In the development of this project, we encountered some challenges that were addressed with time and extensive research:

- Our goal in this project was to re-implement *Lin et al., 2019* paper by considering it in a different scenario: predicting stress based on heart levels. However, we had to spend sometime understanding how to create the appropriate labels in our data to differentiate an observation from stress and no stress. This challenge was resolved by researching and understanding thresholds of heart rate and other variables. In doing so, we created a model labels that were assigned for each timestamp based on the heart rate behavior within a particular period.

- Convolution models tend to become messy and difficult to understand as our data gets passed into several convolving layers. Our task was to appropriately look into how to accurately represent our data at each step, while preserving its meaning for final prediction. Our consideration was to understand how our data was transformed by looking into dimensionality and appreciating how the model behaved through each convolution.

VI. REFLECTION

The purpose of this project was to build a classification model for predicting states of stress. We found that the outcomes of the model exceeded our expectations, with accuracies beyond 90%, further emphasizing that our intentions to create a prediction model were successful. In order to create such successful project, we attempted to understand correctly the number of filters needed at each layer to be able to evaluate and hold the necessary information at each pass. One aspect that we could have delved more into could have been developing more rigorous approaches for classifying stress beyond one feature, heart rate. The dataset that we used contain more variables that could have given us a better picture of what stress is, potentially making the model more or less accurate. If we had had more time, we would have continued identifying ways to label our data and also look into approaches in feature engineering such as clustering and PCA.

Finally, the following are our takeaways of this project:

- understanding how Temporal Convolution Neural Networks can be used for time-dependent data in a rigorous format, compared to other more convenient approaches like LSTMs and RNNs.

- building successful approaches for reducing noise in our data such as attention layers show to have been effective for increasing the accuracy of our model

- the relevance of Deep Learning in the medical field is crucial, and this project demonstrates one more time how small, but meaningful ideas can have an impact on different fields.

VII. BIBLIOGRAPHY

[1] Lei Lin, Beilei Xu, Wencheng Wu, Trevor Richardson, Edgar A. Bernal. (2019, March 18). Medical Time Series Classification with Hierarchical Attention-based Temporal Convolutional Networks: A Case Study of Myotonic Dystrophy Diagnosis.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[3] Shen and H.-y. Lee. Neural attention models for sequence classification: Analysis and application to key term extraction and dialogue act detection. arXiv preprint arXiv:1604.00077, 2016.

[4] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu. Attention-based bidirectional long short-term memory net- works for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Lin- guistics (Volume 2: Short Papers), volume 2, pages 207–212, 2016.

[5] Y. Tang, J. Xu, K. Matsumoto, and C. Ono. Sequence-to- sequence model with attention for time series classification. In 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), pages 503–510. IEEE, 2016.

[6] P. Vinayavekhin, S. Chaudhury, A. Munawar, D. J. Agra- vante, G. De Magistris, D. Kimura, and R. Tachibana. Fo- cusing on what is relevant: Time-series learning and under- standing using attention. In 2018 24th International Con- ference on Pattern Recognition (ICPR), pages 2624–2629. IEEE, 2018.

[7] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cot- trell. A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint arXiv:1704.02971, 2017.

[8] 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 Amer- ican Chapter of the Association for Computational Lin- guistics: Human Language Technologies, pages 1480–1489, 2016.

[9] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271, 2018

[10] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger and Kristof Van Laerhoven. 2018. Introducing WESAD, a multimodal dataset for Wearable Stress and Affect Detection. In 2018 International Conference on Multimodal Interaction (ICMI â€™18), October 16â€“20, 2018, Boulder, CO, USA. ACM, New York, NY, USA, 9 pages.