# Enhancing Analysis of Diadochokinetic Speech Using Deep Neural Networks

Yael Segal-Feldman[a], Kasia Hitczenko[b], Matthew Goldrick[c], Adam Buchwald[d], Angela Roberts[e], Joseph Keshet[a]

[a] *Faculty of Electrical and Computer Engineering, Technion–Israel Institute of Technology, Israel*
[b] *Department of Computer Science, George Washington University, Washington DC, USA*
[c] *Department of Linguistics, Northwestern University, IL, USA*
[d] *Department of Communicative Sciences and Disorders, New York University, NY, USA*
[e] *Department of Computer Science and School of Communication Sciences and Disorders, University of Western Ontario, Ontario, Canada*

## Abstract

Diadochokinetic speech tasks (DDK) involve the repetitive production of consonant-vowel syllables. These tasks are useful in detecting impairments, differential diagnosis, and monitoring progress in speech-motor impairments. However, manual analysis of those tasks is time-consuming, subjective, and provides only a rough picture of speech. This paper presents several deep neural network models working on the raw waveform for the automatic segmentation of stop consonants and vowels from unannotated and untranscribed speech. A deep encoder serves as a features extractor module, replacing conventional signal processing features. In this context, diverse deep learning architectures, such as convolutional neural networks (CNNs) and large self-supervised models like HuBERT, are applied for the extraction process. A decoder model uses derived embeddings to identify frame types. Consequently, the paper studies diverse deep architectures, ranging from linear layers, LSTM, CNN, and transformers. These architectures are assessed for their ability to detect speech rate, sound duration, and boundary locations on a dataset of healthy individuals and an unseen dataset of older individuals with Parkinson's Disease. The results reveal that an LSTM model performs better than all other models on both datasets and is comparable to trained human annotators.

*Keywords:* Diadochokinetic speech, DDK, Deep neural networks, Voice onset time, Vowel duration, Parkinson's Disease

## 1. Introduction

Diadochokinetic (DDK) speech tasks are frequently used by clinicians and researchers to assess potential speech motor impairments [1, 2]. Typically, alternating motion rate (AMR) and sequential motion rate (SMR) trials are included. In AMR trials, participants repeat specific nonsense syllables as quickly and accurately as possible(e.g., pa-pa-pa..., ta-ta-ta..., or ka-ka-ka...). On the other hand, in SMR trials participants repeatedly produce a sequence of three different nonsense syllables (e.g., pa-ta-ka-pa-ta-ka...). These tasks help clinicians evaluate the patient's speech-motor control and their ability to make rapidly alternating speech movements. They have been useful as part of detecting impairments, differential diagnosis, and monitoring progress. As a result, they have become a standard component of many speech and neurological assessments [3, 4, 5, 6, 7].

While this task is central to many studies, it is typically analyzed using a simple measure that is subjectively determined by the test administrator, such as estimating speech rate by counting the number of syllables produced within a fixed period of time [6]. This is clearly useful in many clinical settings due to ease of administration and analysis. However, some work has suggested that such impressionistic evaluations of speech have relatively low inter- and intra-rater reliability [5, 8]. Furthermore, the focus on overall speech rate ignores potentially informative measures such as speech rate variability [9]; limiting our analyses to the single dimension of speech rate loses a great deal of potentially relevant information from the speech signal, as temporal and spectral properties of individual speech sounds and syllables. While these more complex measures have been studied via detailed manual annotations [10], it's unclear how this could be implemented in clinical practice.

To address these issues, researchers have developed automated methods to quantify various properties of DDK productions objectively. Previous research has mainly focused on automatically identifying specific speech properties such as speech rate. Some of the studies focus on signal-processing methods to segment the DDK audio. Räsänen et al. [11] proposed using an oscillator-based model for sonority estimation and rhythmic segmentation to create syllable-like objects, while Rong et al. [7] used the spectral envelope of the signal with other signal processing techniques to estimate both DDK rate and regularity. Another approach was to focus on measuring a primary acoustic cue to the initial consonant of each DDK syllable: voice onset time (VOT) [12]). Montaña et al. [13] used energy content and zero-crossing rate to detect VOT, allowing for the derivation of measures such as speech rate.

Other research has leveraged recent advances in deep learning for the detection of syllables or VOT in untranscribed speech. Rozenstoks et al. [7] suggested using an object detection model from the field of computer vision, namely Faster R-CNN; [14] and fine-tuned it to detect speech syllables, while Wang et al.[15] developed two types of convolutional neural networks (CNNs) specifically for syllable segmentation. Arias-Vergara et al.[16] used a bidirectional recurrent neural network (RNN) on manually extracted time and spectral acoustic features to detect VOT. Finally, a recent paper by [17] used a large pre-trained self-supervised transformer base model, wav2vec2 [18], that is known to outperform CNN and LSTM models in ASR, to measure speech rate.

To the best of our knowledge, Novotnỳ et al.[19] stand among previous work by proposing to automatically segment *both* VOT and vowels. This could allow for a wider array of temporal and spectral measures to be calculated at the level of individual segments. Their signal processing-based methodology resembled that of [13].

Our previous work, [20], focused on two types of deep learning models for the automatic segmentation of unannotated DDK speech. Follow-up work using the best-performing system suggested the presence of subtle motor abnormalities in individuals at clinically high risk for psychosis, based on greater speech rate and VOT variability relative to healthy controls [9]. In this work, we expand the scope of our previous study [20] by exploring various deep-learning architectures (CNNs, LSTMs, and transformers). Our models distinguish themselves by enabling precise segmentation of both vowels and VOTs. They offer the flexibility of employing a variable-length processing window and operating directly on the raw waveform. We also explore the utilization of a large pre-trained self-supervised transformer-based model as a base network, which is then fine-tuned on a small dataset to perform our task. Collectively, these endeavors yield five different types of deep learning architectures.

The models were trained on DDK samples from healthy individuals and were tested on healthy individuals and on individuals with Parkinson's Disease (PD). Our models outperform the currently available state-of-the-art and exhibit comparable performance to trained human annotators.

The contributions of the paper are as follows: (i) presenting models that allow precise segmentation of both vowels and VOTs; (ii) utilizing the raw waveform to eliminate the need for a restrictive representation; (iii) exploring the usage of multiple deep-learning architectures for this task; (iv) exploring the utilization of a large pre-trained self-supervised transformer-based model for accurate segmentation rather than just calculating speech rate; (v) testing our model against other models on two datasets and (vi) an open access model. The implementation of our models is available at: `https://github.com/MLSpeech/DDKtor`.

This paper is structured as follows. In Section 2, we formally introduce the problem setting. Our proposed method is presented in Section 3. Section 4 covers our datasets. Sections 5 and 6 detail our experiments and present results on various datasets. We discuss these results and future directions in Section 7. Finally, Section 8 provides concluding remarks.

## 2. Problem Setting

In the DDK task, we receive an audio signal that contains a repeating pattern of a positive-lag VOT for each voiceless stop consonant, followed by the vowel $\langle a \rangle$. We aim to divide the audio signal into three distinct acoustic objects: VOT, vowel, and other segments representing silence or different sounds. Consequently, our model is designed to take the raw audio as input and produce a sequence that identifies these distinct objects and their corresponding timings.

For a raw audio with a duration denoted as T, we represent the sequence of input samples as $\bar{\mathbf{x}} = (x_1, \ldots, x_T)$, where each $x_t \in \mathbb{R}$ and $1 \leq t \leq T$. The output is structured as a sequence of frames, $\bar{y} = (y_1, \ldots, y_M)$, where $M$ is the number of frames, $y_m \in \mathcal{Y} = \{\text{VOT, Vowel, Other}\}$ for $1 \leq m \leq M$ and each $m$ frame represents the embedded features with $l$-milliseconds (ms) resolution. It is important to note that the number of frames, $M$, is driven from two sources. The first is the signal duration $T$, which can vary between different signals, and the second is the frames' resolution $l$, which is determined by the model architecture. Hence, $M$ is not fixed and may differ case-by-case.

Our models consist of two functions. The first function, called the *encoder*, extracts representation features or embeddings. It is denoted as $f : \mathcal{X} \to E^M$. It operates by taking inputs from the domain $\mathcal{X}$ and producing a sequence of embedding vectors $\bar{e} = (e_1, \ldots, e_M)$. Each vector, $e_m \in E$, represents the acoustic information of the $m$-th frame in the latent representation domain $E \in \mathbb{R}^N$.

The second function, referred to as the *decoder*, is used as a classification function. It is denoted as $g : E^M \to \mathcal{Y}^M$. The function $g$ operates on the sequence of embedding vectors and produces a sequence of $M$ predictions associated with the target objects.

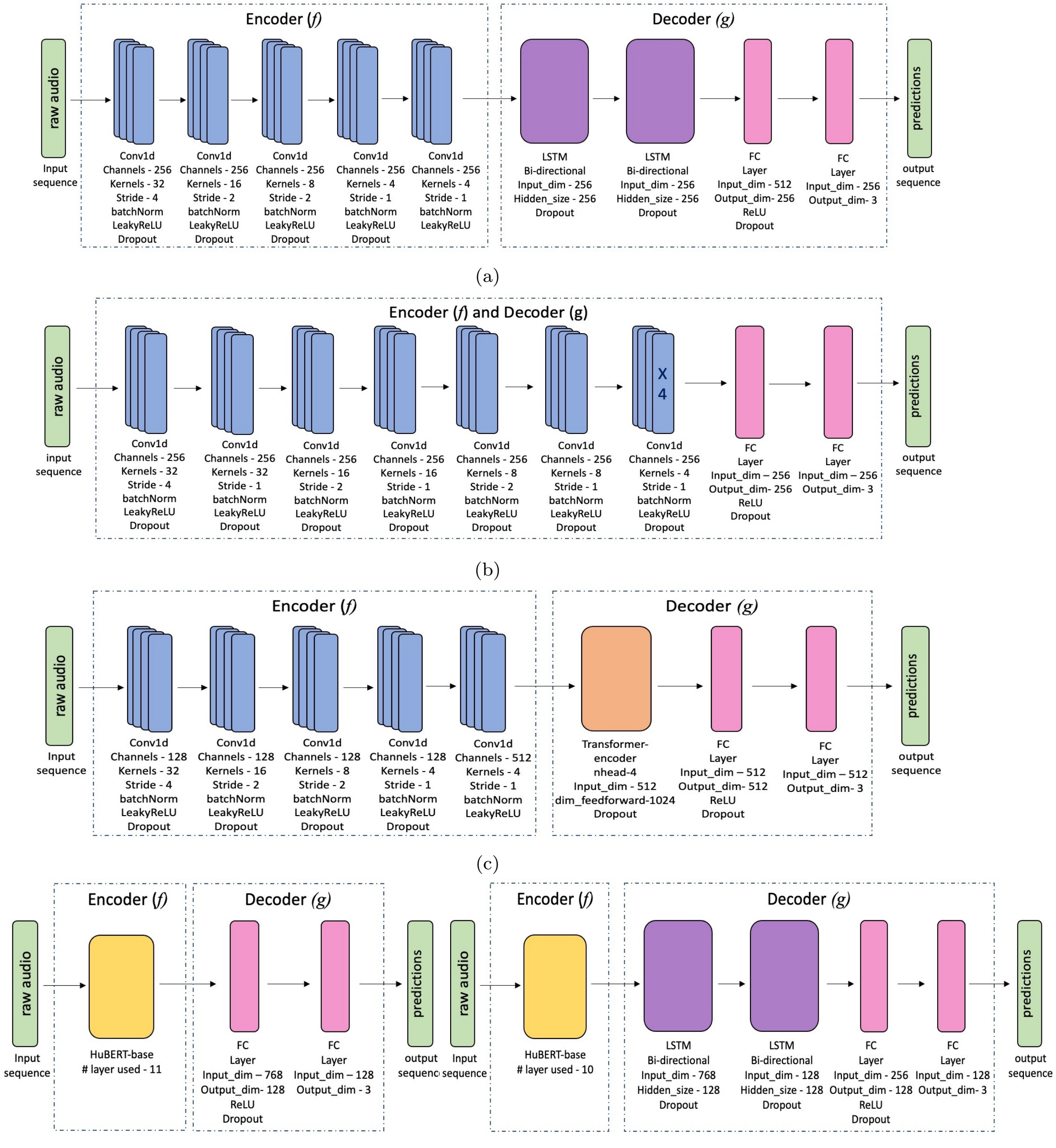
## 3. Models

The paper introduces five different model architectures, each with unique implementations for the encoder $f$ and decoder $g$. Additionally, three of the models produce output frames with a resolution of $l = 1$ ms, while the fourth and fifth models yield output frames with a resolution of $l = 20$ ms due to limitations in their encoder architecture, which will be explained later. All five models employ deep learning architectures for the encoder and decoder. The number of parameters and layers in all models was determined using a validation set, as will be described in the next section.

When working on the raw waveform, a common technique is to use a convolutional neural network (CNN) to replace the classic signal processing features [21, 22]. In addition, the most basic deep neural network architecture that handles sequential data is the Recurrent Neural Network (RNN). Therefore, the first model, denoted as *DDKtor-LSTM* employs a CNN as the encoder and long short-term memory (LSTM), which is a type of RNN as the decoder $g$. The encoder $f$ consists of five 1D convolutional layers with batch normalization, a leaky-ReLU activation function, and a dropout between each layer. The encoder $f$ output is then given as input to the decoder $g$, which is composed of a two-layer bi-directional LSTM and two fully connected (FC) layers.

In wav2letter [23], it was shown that CNN architectures could effectively substitute for recurrent models. We leverage this concept and present the second model, denoted as *DDKtor-CNN*, which employs a CNN architecture for the encoder and the decoder. The combined architecture of the encoder and the decoder consists of ten 1D convolutional layers with batch normalization, a leaky-ReLU activation function, and a dropout between each layer. The output of the CNN is forwarded to two FC layers.

A relatively new architecture for sequential data is the Transformer [24]. In our third model, denoted as *DDKtor-Transform*, we utilize CNN as en-


Encoder (f)
Decoder (g)
raw audio
Input sequence
Conv1d Channels - 256 Kernels - 32 Stride - 4 batchNorm LeakyReLU Dropout
Conv1d Channels - 256 Kernels - 16 Stride - 2 batchNorm LeakyReLU Dropout
Conv1d Channels - 256 Kernels - 8 Stride - 2 batchNorm LeakyReLU Dropout
Conv1d Channels - 256 Kernels - 4 Stride - 1 batchNorm LeakyReLU Dropout
Conv1d Channels - 256 Kernels - 4 Stride - 1 batchNorm LeakyReLU
LSTM Bi-directional Input_dim - 256 Hidden_size - 256 Dropout
LSTM Bi-directional Input_dim - 256 Hidden_size - 256 Dropout
FC Layer Input_dim - 512 Output_dim- 256 ReLU Dropout
FC Layer Input_dim - 256 Output_dim- 3
predictions
output sequence
Encoder (f) and Decoder (g)
raw audio
input sequence
Conv1d Channels - 256 Kernels - 32 Stride - 4 batchNorm LeakyReLU Dropout
Conv1d Channels - 256 Kernels - 32 Stride - 1 batchNorm LeakyReLU Dropout
Conv1d Channels - 256 Kernels - 16 Stride - 2 batchNorm LeakyReLU Dropout
Conv1d Channels - 256 Kernels - 16 Stride - 1 batchNorm LeakyReLU Dropout
Conv1d Channels - 256 Kernels - 8 Stride - 2 batchNorm LeakyReLU Dropout
Conv1d Channels - 256 Kernels - 8 Stride - 1 batchNorm LeakyReLU Dropout
X 4
Conv1d Channels - 256 Kernels - 4 Stride - 1 batchNorm LeakyReLU Dropout
FC Layer Input_dim – 256 Output_dim- 256 ReLU Dropout
FC Layer Input_dim – 256 Output_dim- 3
predictions
output sequence
Encoder (f)
Decoder (g)
raw audio
Input sequence
Conv1d Channels - 128 Kernels - 32 Stride - 4 batchNorm LeakyReLU Dropout
Conv1d Channels - 128 Kernels - 16 Stride - 2 batchNorm LeakyReLU Dropout
Conv1d Channels - 128 Kernels - 8 Stride - 2 batchNorm LeakyReLU Dropout
Conv1d Channels - 128 Kernels - 4 Stride - 1 batchNorm LeakyReLU Dropout
Conv1d Channels - 512 Kernels - 4 Stride - 1 batchNorm LeakyReLU
Transformer-encoder nhead-4 Input_dim - 512 dim_feedforward-1024 Dropout
FC Layer Input_dim – 512 Output_dim- 512 ReLU Dropout
FC Layer Input_dim – 512 Output_dim- 3
predictions
output sequence
Encoder (f)
Decoder (g)
raw audio
Input sequence
HuBERT-base # layer used - 11
FC Layer Input_dim – 768 Output_dim- 128 ReLU Dropout
FC Layer Input_dim – 128 Output_dim - 3
predictions
output sequenc
Encoder (f)
Decoder (g)
raw audio
Input sequence
HuBERT-base # layer used - 10
LSTM Bi-directional Input_dim - 768 Hidden_size - 128 Dropout
LSTM Bi-directional Input_dim - 128 Hidden_size - 128 Dropout
FC Layer Input_dim - 256 Output_dim - 128 ReLU Dropout
FC Layer Input_dim - 128 Output_dim- 3
predictions
output sequence

(a)

(b)

(c)

(d)

(e)

Figure 1: Encoder and decoder architectures for all the models. (a) DDKtor-LSTM's architecture,(b) DDKtor-CNN's architecture,(c) DDKtor-Transform's architecture,(d) DDKtor-HuB-Linear's architecture,(e) DDKtor-HuB-LSTM's architecture.

coder $f$, and Transformer as decoder $g$. *DDKtor-Transform*'s encoder $f$ has the same architecture as in *DDKtor-LSTM*, while the decoder $g$ is composed of one layer of transformer encoder followed by two FC layers.

In recent years, self-supervised learning has been a significant method used to train models in the deep learning field. One of the most well-known models is the HuBERT model [25]. It was trained in a self-supervised manner on a large amount of unlabeled data and used as an encoder for speech. The HuBERT model creates an embedding vector at a 20 ms resolution. This resolution is suitable for vowel duration but not precise enough for VOT, where many empirical effects of interest are significantly smaller [26]. While acknowledging this limitation, a comparison of the performance of this model allows us to explore the trade-off between a model trained on a large amount of data but with less precise output versus a model trained on a small amount of data but with more precise output.

The HuBERT model consists of multiple transformer encoder layers, and we want to investigate the need for a sequential base decoder versus a simple classifier that assumes the relevant sequential information has already been encoded. To accomplish this, we introduce two new models. The fourth model, denoted as *DDKtor-HuB-Linear*, uses a pre-trained HuBERT as the encoder and applies a simple decoder composed of two FC layers. The fifth model, denoted as *DDKtor-HuB-LSTM*, uses a pre-trained HuBERT as the encoder $f$ but applies a two-layer bi-directional LSTM followed by two FC layers as a decoder. Note that the resolution of the HuBERT-based models is 20 ms, which is the original frame resolution of HuBERT.

All models' parameters were trained to minimize the cross-entropy loss function. Each model employs a post-processing procedure to convert frame-based predictions to segment-based ones. This post-processing was done by grouping frames with the same object type.

Figure 1 presents the full architecture, including hyper-parameters for each of the five models, and Table 1 presents a detailed summary of each model's number of parameters.

## 4. Datasets

In our experiments, we used two datasets. The first is used for training and evaluation and the second is used only for evaluation. We start by describing each of them.

| Model | Total # params | $f$ # params | $c$ # params |
|---|---|---|---|
| DDKtor-LSTM | 4.87 | 2.11 | 2.76 |
| DDKtor-CNN | 6.37 | 6.37 | 0 |
| DDKtor-Transform | 3.09 | 0.73 | 2.36 |
| DDKtor-HuB-Linear | 94.47 | 94.37 | 0.1 |
| DDKtor-HuB-LSTM | 95.72 | 94.37 | 1.35 |

Table 1: # of parameters for each model (in millions)

| Split | # Participants | Range (years) | Median (years) | #Female(%) |
|---|---|---|---|---|
| Train | 55 | 18-38 | 22 | 29 (52.7%) |
| Validation | 18 | 18-28 | 22.5 | 12 (66.6%) |
| Test | 19 | 18-39 | 25 | 12 (63.2%) |

Table 2: Demographic information summarization of the participants by split type for Younger NT Adults dataset

***Younger NT Adults*** is the first dataset that was used. It includes speech from the AMR and SMR subtasks for 92 neurotypical (NT) adult participants, collected in a laboratory environment as pre-test data in speech motor learning experiments [27, 28]. The speech signals, sampled at 44.1 kHz with 16-bit resolution, were annotated by two independent annotators for VOTs and vowel durations and boundaries. The onset of the syllable/VOT, was identified by onset of a spike in the waveform and high-amplitude energy across frequencies in the spectrogram. The offset of VOT was determined as the positive zero crossing of the onset of periodic energy in the waveform, corresponding to the appearance of formant structure in the spectrogram. The syllable offset was marked as the positive zero crossing of the last periodic cycle in the vowel, where the formant structure remained visible in the spectrogram. This dataset was used to both training and evaluating our models; therefore participants were randomly split into training ($N = 55$, AMR ∼9 minutes, SMR ∼3 minutes), validation ($N = 18$, AMR ∼ 3 minutes, SMR ∼1 minutes), and test ($N = 19$, AMR ∼4 minutes, SMR ∼2 minutes). Table 2 summarizes the demographic information of the participants by the split type.

***ONDRI*** is the second dataset. it was used to evaluate the algorithm's capacity for generalization to laboratory speech among individuals experiencing motor speech impairments. It was created by the Ontario Neurodegenerative Disease Research Initiative (ONDRI), a longitudinal, multi-site, observational cohort study, using a transdisciplinary approach to characterizing deep endophenotypes in neurodegenerative disorders and their relationship to cerebrovascular disease [29, 30]. As part of a larger study protocol, participants completed a series of motor speech tasks. Speech tasks for participants with Parkinson's disease were completed during the optimal ON state of participants' levodopa medication. Participants completed both AMR and SMR tasks in a fixed order. Speech samples were collected using an AKG 520C head-worn microphone connected to a PC laptop via a Scarlett 2i2 pre-amplifier. Audio files were recorded at a sampling rate of 44.1 kHz and 16-bit. The participants were instructed to perform the AMR and SMR tasks as rapidly and regularly as possible in one breath. Although participants were instructed to perform each task only once, occasionally, participants stopped the task prematurely ($<$ 3 seconds) and were prompted by the examiner to repeat the trial. Each participant was recorded twice: once at the beginning of the study and once a year later. At the beginning of the study, there were 147 participants ($N = 147$) aged 55-85, and after a year, 25 participants were excluded, resulting in 122 participants ($N = 122$). AMR and SMR task files were edited manually to ensure uniform file duration for automated analysis and equivalent sampling window sizes across participants. When disruptions in the task occurred ($>$ 200 ms), the longest syllable train produced was used in the analysis. Once the longest syllable string was identified, the first and last syllables were deleted from the utterance, which resulted in segments of speech of approximately 5 seconds long per task. The SMR tasks were edited so no syllable was dropped from a trisyllable sequence train. No file had less than 6 syllables (i.e., two trisyllabic sequence trains).

Once the files were trimmed, a custom Praat [31] script was used to segment individual syllables. The script output was manually inspected by trained raters. If it created extraneous syllables or omitted syllables, each rater was instructed to add or remove syllable boundaries when necessary, using information from the audio file, spectrogram, and waveform to determine the correction (n.b. inter-annotator agreement is not available). The Praat script and the annotators segmented *ONDRI* utterances only on a syllable-level basis, so this dataset does not provide information on VOTs and vowel durations.

***Sub-ONDRI***: To further evaluate our system for VOT and vowel durations, beyond mere rate calculations on the ONDRI  dataset, a subgroup of five ($N = 5$) participants, referred to as *Sub-ONDRI*, aged 59-77 years

| Model | BS | dropout | LR | No. of HuBERT Layer |
|---|---|---|---|---|
| DDKtor-LSTM | 32 | 0.3 | 0.0001 | - |
| DDKtor-CNN | 32 | 0.3 | 0.0001 | - |
| DDKtor-Transform | 32 | 0.4 | 0.0007 | - |
| DDKtor-HuB-Linear | 32 | 0.4 | 0.0003 | 11 |
| DDKtor-HuB-LSTM | 32 | 0.4 | 0.0003038 | 10 |

Table 3: The selected batch size (BS), dropout, learning rate (LR), and the number of the HuBERT layer whose outputs were used as features, for each model.

old, was selected from the full ONDRI dataset. Two independent annotators annotated this subset for VOT and vowel durations and boundaries using the same criteria as the Younger NT Adults dataset, allowing boundaries calculations and the computation of the gold-standard inter-annotator agreement.

Note that both ONDRI dataset and Sub-ONDRI sub-set were used only to evaluate our models and not for training.

## 5. Experiments

### *5.1. Experiments details*

We resampled the audio files at 16 kHz and divided long audio files into one-second segments for all five models. To enhance the algorithm's performance, we employed data augmentations using the *WavAugment* package [32]. These augmentations included using the original audio utterances without noise, introducing noisy speech[1] with signal-to-noise ratios of 5, 10, 15 dB, and applying band-reject filtering to the speech by eliminating randomly selected spectral components. To create varying input lengths, we randomly shifted the starting point of each one-second frame input.

We conducted a hyper-parameters search, using the sweep tool of the Weights& Biases AI platform[2] to determine the optimal batch size, dropout, learning rate (LR), and the HuBERT layer number (out of 12 transformer-encoder layers) whose outputs were used as features to the decoder $g$, for each model. The chosen hyper-parameters are outlined in Table 3. The

[1] The car noise from the [33] package was included, resembling the air conditioning noise commonly present in the datasets. However, other noises yielded similar outcomes.

[2] https://wandb.ai/site/sweeps

input and output dimensions of the decoder, the number of LSTM or transformer encoder layers, and the number of channels when the encoder $f$ is composed of CNN layers were also determined by hyper-parameters search. The selected values are specified in Figure 1. We use the base HuBERT architecture (with $\sim$ 95.M parameters) pre-trained on the Librispeech 960 dataset to both DDKtor-HuB-Linear and DDKtor-HuB-LSTM and fine-tuned them on the *Younger NT Adults* dataset.

We compared our models against the state-of-the-art model Arias-Vergara et al. [16], denoted as AV-GRU. We trained it on the *Younger NT Adults* dataset without using data augmentation, as it dramatically reduced performance. We also compared our model against Räsänen et al. [11], which presents an algorithm for the segmentation of syllable-like objects. Hence, we compare it only for the DDK speech rate task, using the best parameters they reported in their study ($f0$ = 8Hz, Q = 0.8, and $\delta = 0.01$). Implementations of [15, 7, 13, 19, 17] were not available.

For the three models with the resolution of $l = 1$ ms, we also mark short VOTs (less than five ms) and short vowels (less than 20 ms) as silence and then convert a brief silence (less than 20 ms) between two VOT segments to a single VOT segment.

## 6. Results

We evaluated the models' performances of DDK speech rate and VOT and vowel segments' duration and boundaries. These assessments compare the models against the gold-standard derived from the manual annotations. Additionally, as a benchmark for performance evaluation, we include metrics reflecting the agreement among annotators, referred to as *annotators*.

### *6.1. Diadochokinetic speech rate*

The DDK rate quantifies syllable production by dividing the total number of syllables produced by the overall articulation time. Specifically, it represents the duration between the VOT onset of the initial produced syllable and the vowel offset of the last produced syllable. Each participant's DDK rate comprises four distinct rates: one for each syllable type within the AMR task (i.e., pa, ta and ka) and one for the SMR task.

The DDKtor models output VOTs and Vowels rather than syllables, which requires post-processing to extract the rate. The post-processing involves combining VOT and vowel into a single syllable when the gap between

| Model | Younger NT Adults | Sub-ONDRI | ONDRI |
|---|---|---|---|
| DDKtor-LSTM | **0.930(0.19)** | **0.994(0.05)** | **0.924(0.17)** |
| DDKtor-CNN | 0.918(0.29) | 0.894(0.41) | 0.707(0.57) |
| DDKtor-Transform | 0.852(0.85) | 0.818(0.86) | 0.635(0.77) |
| DDKtor-HuB-Linear | 0.882(0.88) | 0.965(0.32) | 0.918(0.18) |
| DDKtor-HuB-LSTM | 0.910(0.91) | 0.970(0.43) | 0.910(0.19) |
| Arias-Vergara et al. [16] | 0.865(0.865) | 0.903(0.464) | 0.793(0.39) |
| Räsänen et al. [11] | 0.601(0.991) | 0.932(0.511) | 0.775(0.5) |
| Annotators | 0.994(0.04) | 0.994(0.03) | - |

Table 4: Correlations (r) between model and annotator DDK rates (mean absolute errors in parentheses). The result on Younger NT Adults refers to the performance on the test set. Bold indicates the best-performing model within each column. All correlations are significant with p < 1e-5.

them is less than 25 ms (this gap was selected through optimization on the validation set). Additionally, we increment the syllable count whenever the predicted vowel duration is more than twice the participant's average vowel duration. This addresses the issue of models sometimes combining two adjacent syllables (treating flapped /t/s as a part of the vowel).

The model of Arias-Vergara et al. [16] sometimes skips VOTs. Therefore, in post-processing, we incremented the syllable count whenever the time elapsed between two VOTs was more than twice the average inter-VOT duration for that DDK trial. It is important to note that because Arias-Vergara et al. [16] estimates only VOT, it cannot calculate total articulation time (which requires a value for full syllable length, including VOT and vowel duration).

To ensure a fair and consistent comparison across all models, we calculated the total articulation time using a manually annotated window that covers the entire set of syllables.

Table 4 presents the correlations (r) and mean absolute errors between the models and the annotator. All our models produced highly correlated results with the annotator for Younger NT Adults and Sub-ONDRI datasets. The DDKtor-LSTM model presents the highest correlation across all the datasets. It is interesting to note that while all our models perform better than AV-GRU and the Räsänen et al.[11] model on the test set of the Younger NT Adults dataset, this isn't the case on the Sub-ONDRI and ONDRI datasets, where DDKtor-CNN and DDKtor-Transform have lower correlation than all

|  | Younger NT Adults |  | Sub-ONDRI |  | ONDRI |
|---|---|---|---|---|---|
| Model | VOT F1 | Vowel F1 | VOT F1 | Vowel F1 | Syllable F1 |
| DDKtor-LSTM | **0.978** | **0.983** | **0.994** | **0.997** | **0.984** |
| DDKtor-CNN | 0.956 | 0.962 | 0.968 | 0.963 | 0.942 |
| DDKtor-Transform | 0.948 | 0.954 | 0.925 | 0.955 | 0.927 |
| DDKtor-HuB-Linear | 0.936 | 0.973 | 0.958 | 0.975 | 0.980 |
| DDKtor-HuB-LSTM | 0.942 | 0.976 | 0.959 | 0.969 | 0.979 |
| Arias-Vergara et al. [16] | 0.933 | - | 0.938 | - | 0.884 |
| Annotators | 0.998 | 1 | 1 | 1 | - |

Table 5: F1-scores for VOT and vowel segments prediction

other models. Additionally, the DDKtor-CNN and DDKtor-Transform models exhibit a considerable decrease in performance when tested on the ONDRI dataset. On the other hand, DDKtor-LSTM, DDKtor-HuB-Linear, and DDKtor-HuB-LSTM show a moderate reduction in the results for the same dataset. Overall, these models demonstrate high accuracy in predicting DDK rates from unannotated DDK samples across different datasets and populations.

### *6.2. Segments duration and boundaries*

In this section, the models' performances are measured at the segment level, and we analyze the accuracy of the predicted boundaries and the durations of both VOT and vowel segments. Recall that the outputs of our models are frames that we group together by their type in order to predict a single-segment object.

To evaluate the models' performance, we established a matching scheme for prediction-target segment pairs. This involved pairing each predicted segment with the target segments possessing an overlapping time, closest start or end times. The selected match represented the target assignment with the most significant overlap[3]. Instances of missed detection occurred when these target segments were omitted within the overlapping region. False alarms were computed accordingly in alignment with this process, allowing

[3] Here, predicted sections are matched with target sections based on the highest intersection-over-union (IOU) from the targets with overlapping time, closest start or end times. This is slightly different from [20], where the predicted segment is only matched with targets having the closest start and end times.

the computation of an F1 score. Table 5 presents the F1 scores for VOTs and vowels segment detection for each model by dataset. All the models present high F1 scores across all the datasets. Again, the DDKtor-LSTM model presents the best results, which are very close to the Annotators' correlations.

We will now assess the precision of the boundaries and the durations. To achieve this, we must eliminate all the miss-detected and false positive segments from the models. Our analysis will focus solely on segments mutually identified by each model and the annotator. For the AV-GRU model, only the VOT onset and offset were calculated. Additionally, following [26, 20], we excluded outliers (the top 5% and bottom 2% of duration values) from both the models' and annotator's predictions. Due to the fact that the ONDRI dataset was annotated at the syllable level and lacks precise VOT and vowel annotations, it was excluded from the assessment of boundaries and durations.

| | Younger NT Adults | | Sub-ONDRI | |
|---|---|---|---|---|
| Model | VOTs | Vowels | VOTs | Vowels |
| DDKtor-LSTM | **0.919(4)** | **0.922(7)** | **0.682(7)** | **0.92(8)** |
| DDKtor-CNN | 0.874(4) | 0.905(8) | 0.566(8) | 0.882(11) |
| DDKtor-Transform | 0.864(5) | 0.901(8) | 0.585(9) | 0.911(9) |
| DDKtor-HuB-Linear | 0.771(8) | 0.872(10) | 0.592(8) | 0.90(9) |
| DDKtor-HuB-LSTM | 0.798(8) | 0.857(11) | 0.6(8) | 0.915(8) |
| Arias-Vergara et al. [16] | 0.807(7) | - | 0.482(8) | - |
| annotators | 0.929(3) | 0.960(5) | 0.75(5) | 0.959(4) |

Table 6: Correlations between model and annotator durations by dataset (mean absolute errors in ms in parentheses). Bolded values represent the best-performing models within each column (VOT or vowel in each dataset). All correlations are significant with p < 0.0001

Table 6 presents the correlation between each model's predicted duration, the annotated duration, and the mean absolute error rates in milliseconds. DDKtor-LSTM achieves the highest correlations with the annotator across test sets. The DDKtor-HuB-Linear and DDKtor-HuB-LSTM display good correlation for the vowels in both Younger NT Adults and Sub-ONDRI even though they output prediction in a 20-millisecond resolution. Note the correlations for the duration of VOTs in the Sub-ONDRI dataset are significantly lower than those in the Younger NT Adults dataset. However, this pattern also holds for the VOT correlation among the annotators themselves in the Sub-ONDRI dataset. Nevertheless, the LSTM model performs at a level close to gold-standard.

| | Model | VOT Onset | VOT Offset | Vowel Onset | Vowel Offset |
|---|---|---|---|---|---|
| Younger NT Adults | DDKtor-LSTM | 2.09 | **2.90** | 3.14 | **6.41** |
| | DDKtor-CNN | **2.08** | 3.27 | **2.77** | 7.39 |
| | DDKtor-Transform | 2.40 | 3.39 | 3.16 | 6.66 |
| | DDKtor-HuB-Linear | 6.85 | 6.77 | 7.47 | 9.40 |
| | DDKtor-HuB-LSTM | 6.58 | 6.60 | 7.34 | 9.73 |
| | Arias-Vergara et al. [16] | 4.10 | 4.78 | - | - |
| | annotators | 1.42 | 2.46 | 2.42 | 2.98 |
| Sub-ONDRI | DDKtor-LSTM | 3.64 | 6.39 | **6.43** | **3.45** |
| | DDKtor-CNN | 3.22 | 6.77 | 7.63 | 5.54 |
| | DDKtor-Transform | **2.97** | 7.91 | 7.00 | 3.57 |
| | DDKtor-HuB-Linear | 6.77 | 7.68 | 8.55 | 7.12 |
| | DDKtor-HuB-LSTM | 6.68 | 7.17 | 7.38 | 6.661 |
| | Arias-Vergara et al. [16] | 5.65 | **5.81** | - | - |
| | annotators | 2.50 | 3.21 | 3.06 | 1.90 |

Table 7: Mean absolute deviation of boundary onsets and offsets (milliseconds) by dataset.

Table 7 presents the mean absolute deviation of boundary location across boundary types (VOT onset, VOT offset, vowel onset, and vowel offset) of the models and annotators. For the Younger NT Adults test set, the DDKtor-LSTM and DDKtor-CNN models shows the best performance. They perform comparably to annotators on VOT boundaries, with 2 to 3.39 ms deviations. However, the models demonstrate slightly higher mean absolute deviations, approximately 6 ms, for the vowel offset.

When evaluating the Sub-ONDRI dataset, the results are slightly worse, and no single model outperforms the others across all boundary types. However, DDKtor-LSTM still performs well, with mean absolute deviations ranging from around 3-6 ms. Note as well that the DDKtor-HuB-Linear and DDKtor-HuB-LSTM present very good deviation results for both the VOT and vowel, considering that their output resolution is 20 milliseconds.

## 7. Discussion

In this section, we examine the earlier-stated outcomes and our exploration of various model architectures. While the DDKtor-CNN model demonstrates strong performance on the Younger NT Adults and Sub-ONDRI datasets, its efficacy noticeably declines when applied to the ONDRI dataset. The assessment of the DDKtor-Transform model uncovers the lowest DDK correlation rate across all models and datasets, with a considerable reduction for the ONDRI dataset. However, the findings remain inconclusive regarding segment durations and boundaries, sometimes displaying relatively lower outcomes for the DDKtor-CNN, DDKtor-HuB-Linear, and DDKtor-HuB-LSTM models, depending on the specific measure.

Overall, the DDKtor-LSTM model consistently outperforms others across most metrics and datasets. This suggests that while full CNN or transformer architectures have merits, they may not entirely supersede recurrent neural networks (RNN) for sequential tasks, particularly when training from scratch on a smaller dataset.

We also want to explore the usage of large models, which were pre-trained unsupervised on unlabeled data. Both the DDKtor-HuB-Linear and DDKtor-HuB-LSTM models present high DDK rate correlation across all datasets. While differences between the two models exist, they aren't significant. At times, DDKtor-HuB-LSTM outperforms, while in other scenarios, DDKtor-HuB-Linear does better, hinting that the transformer-encoder layers of the HuBERT model capture most of the necessary information.

Additionally, both DDKtor-HuB-Linear and DDKtor-HuB-LSTM produce outputs at a 20-millisecond resolution, suitable for vowel duration segments. Notably, for Sub-ONDRI , these vowel correlations exceed even those of DDKtor-CNN, demonstrating a higher correlation. For VOT segments, lower correlations in duration are evident in the Younger NT Adults dataset. However, in the Sub-ONDRI dataset, both DDKtor-HuB-Linear and DDKtor-HuB-LSTM outperform DDKtor-CNN and DDKtor-Transform, despite their 1-millisecond output resolution. This disparity is particularly pronounced in the ONDRI dataset, where DDKtor-HuB-Linear and DDKtor-HuB-LSTM show the lowest mean absolute deviation for syllable offset.

### *7.1. Sub-ONDRI annotation versus ONDRI annotation*

In Section 6, our models excelled with the Younger NT Adults and Sub-ONDRI datasets but weren't compared to the ONDRI dataset as it has only

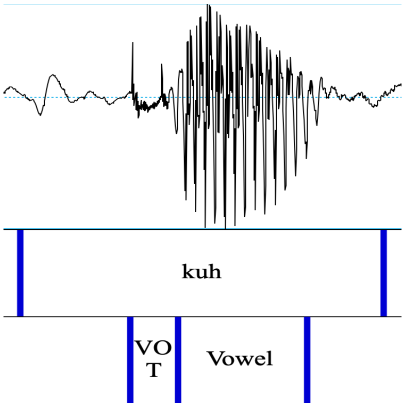


Figure 2: Syllable example of ONDRI dataset. The first layer shows syllable boundaries by the ONDRI annotator, while the second layer depicts boundaries by our DDKtor-LSTM model.

| Measure | Annotator 1 | Annotator 2 |
|---|---|---|
| rate correlation | 0.997 | 0.997 |
| rate syllable error | 0.1 | 0.1 |
| duration correlation | 0.845 | 0.821 |
| duration error | 23.16 (ms) | 24.91(ms) |
| VOT onset | 6.76(ms) | 7.36(ms) |
| Vowel offset | 21.21(ms) | 21.9(ms) |

Table 8: Comparison of the annotations of the same utterances in Sub-ONDRI and ONDRI datasets. The Table presents the correlation rate, syllables mean absolute error, duration correlation, duration mean absolute error (in ms), VOT onset mean absolute deviation (in ms) and Vowel offset mean absolute deviation (in ms).

syllable level annotation. As Sub-ONDRI was annotated by two human annotators for both VOTs and vowels, with higher boundaries accuracy, it is interesting to analyze the difference in the annotation.

In Section 6, our models excelled with the Younger NT Adults and Sub-ONDRI datasets but were not compared to the ONDRI dataset as it only has syllable-level annotations. The Sub-ONDRI dataset was annotated by two human annotators for both VOTs and vowels, with high accuracy in annotating boundaries, making it interesting to analyze the differences in the annotation.

A comparison of shared utterances in both datasets, presented in Table 8, revealed a high correlation rate, as expected, but considerable differences in duration and boundaries. The mean absolute duration errors for annotator 1 and annotator 2 are 23.16 ms and 24.91 ms, respectively. Since the vowel offset deviation is approximately 21 ms for both annotators, we can deduce that most of the duration error comes from this source. Upon analyzing the DDKtor-LSTM model on the ONDRI dataset, we found a duration error of 29.7 ms, a mean absolute deviation of 10.81 ms for VOT onset, and 23.24 ms for vowel offset. Interestingly, the DDKtor-LSTM model's duration error is close to that of the annotators on the Sub-ONDRI dataset, with a difference of only 5 ms. As observed with the annotators, most of the DDKtor-LSTM model's duration errors originate from the vowel offset.

To illustrate further, Figure 2 includes a syllable example from the ONDRI dataset. The first layer shows syllable boundaries according to the ONDRI annotation, while the second layer depicts boundaries identified by our DDKtor-LSTM model.

These results suggest that our DDKtor-LSTM model has the potential for high performance on larger datasets, as evidenced by its close alignment with the Sub-ONDRI annotation compared to the ONDRI dataset.

### *7.2. Future directions*

While the DDKtor-LSTM model exhibits good performance, there are several areas where it could be improved. The model is not robust to certain kinds of phonetic variation. We added in post-processing measures to correct for cases where the model fails to detect syllables. An especially challenging context (occurring in a small number of cases) are syllables where /t/ is flapped. Inspection of the results also shows that the model has difficulty (in a small number of participants) with high degrees of creaky phonation. We believe both of these reflect the low frequency of occurrence of such phonetic variants in our training materials. A more diverse set of materials, deliberating selecting for the occurrence of low frequency variants (flaps, creaky voice) may improve performance. This may be critical for extending to a variety of clinical conditions that leads to greater distortions of the phonetic properties of the syllable sequences. It will also likely be critical to extend training and testing to less controlled acoustic environments. Here, we utilized lab speech, recorded under near-ideal conditions. The use of this tool for clinical applications will require demonstrating robustness to variation in these recording conditions.

Finally, we have limited our training and testing to voiceless consonant-initial syllables. In some contexts, voiced consonants are used as well (e.g., ba da ga instead of pa ta ka). Future work can extend our approach to these other syllables.

## 8. Conclusions

The automated examination of complex acoustic features in diadochokinetic speech holds the potential to offer novel insights into speech motor disorders, alleviating the burdens on both clinicians and researchers.

We found that the DDKtor-LSTM model consistently outperforms others across most metrics and datasets. This suggests that while full CNN or

Transformer architectures have their advantages, they not completely surpass recurrent neural networks (RNNs) for sequential tasks. Additionally, although models based on HuBERT, trained on large amounts of unlabeled data, show high DDK rate and duration correlations across all datasets, the DDKtor-LSTM model still outperforms them.

Overall, DDKtor-LSTM achieved state-of-the-art performance on untranscribed, unannotated speech, performing almost as well as human annotators across all the datasets. This algorithm can allow for more detailed automatic analyses of DDK samples, providing new insights into motor speech behavior.

## 9. Acknowledgments

This work is supported by the Ministry of Science & Technology, Israel (Y. Segal); U.S. National Institutes of Health (NIH; grants R21MH119677, K01DC014298, R01DC018589); U.S. National Science Foundation (NSF; grant DRL2219843); the Binational Science Foundation (BSF; grant 2022618); and the Ontario Brain Institute (an independent non-profit corporation, funded partially by the Ontario government). Matching funds provided by participating hospitals, the Baycrest Foundation, Bruyère Research Institute, Centre for Addiction and Mental Health Foundation, London Health Sciences Foundation, McMaster University Faculty of Health Sciences, Ottawa Brain and Mind Research Institute, Queen's University Faculty of Health Sciences, the Sunnybrook Foundation, the Thunder Bay Regional Health Sciences Centre, the University of Ottawa Faculty of Medicine, and the Windsor/Essex County ALS Association. The Temerty Family Foundation provided infrastructure matching funds. The opinions, results, and conclusions are those of the authors, and no endorsement by the Ontario Brain Institute or NIH is intended or should be inferred. Thanks to Hung-Shao Cheng, Rosemary Dong, Katerina Alexopoulos, Camila Hirani, and Jasmine Tran for their help in data collection and processing. The authors also acknowledge ONDRI's clinical coordinators, including Catarina Downey, Heather Hink, Donna McBain, Lindsey McLeish, and Alicia J. Peltsch.

## References

[1] R. D. Kent, J. F. Kent, J. C. Rosenbek, Maximum performance tests of speech production, Journal of speech and hearing disorders 52 (4) (1987) 367–387.

[2] M. Nishio, S. Niimi, Comparison of speaking rate, articulation rate and alternating motion rate in dysarthric speakers, Folia Phoniatrica et Logopaedica 58 (2) (2006) 114–131.

[3] P. Rong, Y. Yunusova, J. Wang, J. R. Green, et al., Predicting early bulbar decline in amyotrophic lateral sclerosis: A speech subsystem approach, Behavioural neurology 2015 (2015).

[4] R. Baken, R. Orlikoff, Speech movements, San Diego: Singular Thomson Learning (2000) 511–557.

[5] M. Gadesmann, N. Miller, Reliability of speech diadochokinetic test measurement, International Journal of Language & Communication Disorders 43 (1) (2008) 41–54.

[6] R. D. Kent, Y. Kim, L.-m. Chen, Oral and laryngeal diadochokinesis across the life span: A scoping review of methods, reference data, and clinical applications, Journal of Speech, Language, and Hearing Research 65 (2) (2022) 574–623.

[7] K. Rozenstoks, M. Novotny, D. Horakova, J. Rusz, Automated assessment of oral diadochokinesis in multiple sclerosis using a neural network approach: Effect of different syllable repetition paradigms, IEEE Transactions on Neural Systems and Rehabilitation Engineering 28 (1) (2020) 32–41. doi:10.1109/TNSRE.2019.2943064.

[8] F. Karlsson, E. Schalling, K. Laakso, K. Johansson, L. Hartelius, Assessment of speech impairment in patients with parkinson's disease from acoustic quantifications of oral diadochokinetic sequences, The Journal of the Acoustical Society of America 147 (2) (2020) 839–851.

[9] K. Hitczenko, Y. Segal, J. Keshet, M. Goldrick, V. A. Mittal, Speech characteristics yield important clues about motor function: Speech variability in individuals at clinical high-risk for psychosis, Schizophrenia 9 (1) (2023) 60.

[10] W. Ziegler, Task-related factors in oral motor control: Speech and oral diadochokinesis in dysarthria and apraxia of speech, Brain and language 80 (3) (2002) 556–575.

[11] O. Räsänen, G. Doyle, M. C. Frank, Pre-linguistic segmentation of speech into syllable-like units, Cognition 171 (2018) 130–150.

[12] A. S. Abramson, D. H. Whalen, Voice onset time (vot) at 50: Theoretical and practical issues in measuring voicing distinctions, Journal of phonetics 63 (2017) 75–86.

[13] D. Montaña, Y. Campos-Roca, C. J. Pérez, A diadochokinesis-based expert system considering articulatory features of plosive consonants for early detection of parkinson's disease, Computer methods and programs in biomedicine 154 (2018) 89–97.

[14] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

[15] Y. Y. Wang, K. Gao, A. M. Kloepper, Y. Zhao, M. Kuruvilla-Dugdale, T. E. Lever, F. Bunyak, Deepddk: A deep learning based oral-diadochokinesis analysis software, in: 2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), IEEE, 2019, pp. 1–4.

[16] T. Arias-Vergara, P. Arguello-Velez, J. C. Vásquez-Correa, E. Nöth, M. Schuster, M. C. Gonzalez-Rátiva, J. R. Orozco-Arroyave, Automatic detection of voice onset time in voiceless plosives using gated recurrent units, Digital Signal Processing 104 (2020) 102779.

[17] P. Kadambi, G. M. Stegmann, J. Liss, V. Berisha, S. Hahn, Wav2ddk: Analytical and clinical validation of an automated diadochokinetic rate estimation algorithm on remotely collected speech, Journal of Speech, Language, and Hearing Research 66 (8S) (2023) 3166–3181.

[18] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, Advances in neural information processing systems 33 (2020) 12449–12460.

[19] M. Novotnỳ, J. Rusz, R. Čmejla, E. Růžička, Automatic evaluation of articulatory disorders in parkinson's disease, IEEE/ACM Transactions on Audio, Speech, and Language Processing 22 (9) (2014) 1366–1378.

[20] Y. Segal, K. Hitczenko, M. Goldrick, A. Buchwald, A. Roberts, J. Keshet, Ddktor: Automatic diadochokinetic speech analysis, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, Vol. 2022, 2022, pp. 4611–4615.

[21] S. Schneider, A. Baevski, R. Collobert, M. Auli, wav2vec: Unsupervised pre-training for speech recognition, arXiv preprint arXiv:1904.05862 (2019).

[22] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748 (2018).

[23] R. Collobert, C. Puhrsch, G. Synnaeve, Wav2letter: an end-to-end convnet-based speech recognition system, arXiv preprint arXiv:1609.03193 (2016).

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[25] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, A. Mohamed, Hubert: Self-supervised speech representation learning by masked prediction of hidden units, IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2021) 3451–3460.

[26] M. Goldrick, J. Keshet, E. Gustafson, J. Heller, J. Needle, Automatic analysis of slips of the tongue: Insights into the cognitive architecture of speech production, Cognition 149 (2016) 31–39.

[27] A. Buchwald, H. Calhoun, S. Rimikis, M. S. Lowe, R. Wellner, D. J. Edwards, Using tdcs to facilitate motor learning in speech production: The role of timing, Cortex 111 (2019) 274–285.

[28] H.-S. Cheng, A. Buchwald, Does voicing affect patterns of transfer in nonnative cluster learning?, Journal of Speech, Language, and Hearing Research (2021) 1–18.

[29] P. M. McLaughlin, K. M. Sunderland, D. Beaton, M. A. Binns, D. Kwan, B. Levine, J. B. Orange, A. J. Peltsch, A. C. Roberts, S. C. Strother, et al., The quality assurance and quality control protocol for neuropsychological data collection and curation in the ontario neurodegenerative

disease research initiative (ondri) study, Assessment 28 (5) (2021) 1267–1286.

[30] K. M. Sunderland, D. Beaton, S. R. Arnott, P. Kleinstiver, D. Kwan, J. M. Lawrence-Dewar, J. Ramirez, B. Tan, R. Bartha, S. E. Black, et al., Characteristics of the ontario neurodegenerative disease research initiative cohort, Alzheimer's & Dementia 19 (1) (2023) 226–243.

[31] P. Boersma, D. Weenink, Praat: doing phonetics by computer [Computer program](version 6.2.06) (1992-2022).
URL https://www.praat.org

[32] E. Kharitonov, M. Rivière, G. Synnaeve, L. Wolf, P.-E. Mazaré, M. Douze, E. Dupoux, Data augmenting contrastive learning of speech representations in the time domain, in: 2021 IEEE Spoken Language Technology Workshop (SLT), 2021, pp. 215–222. doi:10.1109/SLT48900.2021.9383605.

[33] A. Varga, H. J. Steeneken, Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems, Speech Communication 12 (3) (1993) 247–251.