Keystroke-Dynamics for Parkinson's Disease Signs Detection in an At-Home Uncontrolled Population: A New Benchmark and Method

Shikha Tripathi, Teresa Arroyo-Gallego, and Luca Giancardo

***Abstract*—Parkinson's disease (PD) is the second most prevalent neurodegenerative disease disorder in the world. A prompt diagnosis would enable clinical trials for disease-modifying neuroprotective therapies. Recent research efforts have unveiled imaging and blood markers that have the potential to be used to identify PD patients promptly, however, the idiopathic nature of PD makes these tests very hard to scale to the general population. To this end, we need an easily deployable tool that would enable screening for PD signs in the general population. In this work, we propose a new set of features based on keystroke dynamics, i.e., the time required to press and release keyboard keys during typing, and used to detect PD in an ecologically valid data acquisition setup at the subject's homes, without requiring any pre-defined task. We compare and contrast existing models presented in the literature and present a new model that combines a new type of keystroke dynamics signal representation using hold time and flight time series as a function of key types and asymmetry in the time series using a convolutional neural network. We show how this model achieves an Area Under the Receiving Operating Characteristic curve ranging from 0.80 to 0.83 on a dataset of subjects who actively interacted with their computers for at least 5 months and positively compares against other state-of-the-art approaches previously tested on keystroke dynamics data acquired with mechanical keyboards.**

***Index Terms*—Keystroke dynamics, convolutional neural networks, Parkinson's Disease, neurodegeneration.**

Manuscript received 21 February 2022; revised 7 June 2022; accepted 26 June 2022. Date of publication 29 June 2022; date of current version 26 December 2022. The work of Luca Giancardo was supported in part by the Translational Research Institute for Space Health through NASA Cooperative Agreement under Grant NNX16AO69A, in part by NIH under Grants UL1TR003167 and R01NS121154, and in part by the Cancer Prevention and Research Institute of Texas under Grant RP 170668.

Shikha Tripathi is with the Center for Precision Health, School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston, TX 77030 USA.

Teresa Arroyo-Gallego is with the nQ Medical, Cambridge, MA 02142 USA.

Luca Giancardo is with the Center for Precision Health, School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston, TX 77030 USA (e-mail: luca.giancardo@uth.tmc.edu).

Digital Object Identifier 10.1109/TBME.2022.3187309

I. Introduction

PARKINSON'S is a neurodegenerative disorder affecting 6.1 million individuals across the globe according to the Global Burden of Disease Study 2016 [1]. It is characterized by a broad spectrum of motor and non-motor symptoms [2]. Parkinson's can change the quality of an individual's life over a prolonged period with its slow but progressive nature. PD is mostly an idiopathic disease with a complex diagnosis that requires the holistic evaluation of motor and non-motor signs and imaging exams by a movement disorder specialist. PD diagnosis is further complicated in its early stages. Identification of PD as early as possible is essential for the delivery of future neuroprotective drugs that could slow down the disease or to identify potential candidates for clinical trials for such drugs [3]. While new imaging and assay-based tools are being developed [4], [5], [6], they cannot be easily scaled to a population level. Easily deployable tools that could deliver an initial PD screening test at home will be an essential part for developing and delivering disease-modifying therapies.

Wearable devices and inertial measurement units (IMUs) have been used to identify PD signs such as tremors, rigidity, or freezing of gait by employing deep learning [7] and anomaly detectors [8]. Other promising approaches have used human voice [9], [10] to identify vocal cord changes induced by PD. Recently, a new type of method based on keystroke dynamic analysis has emerged as an alternative to create tools able to measure PD signs without the need of dedicated hardware, at the patient's home, and without interfering with their daily routine. The first proof of principle for these methodologies was presented for mechanical keyboards [11], [12] and then extended to software keyboards on touchscreen devices [13]. After that, other research groups have proposed alternative algorithms to detect PD from keystroke dynamics in mechanical keyboards [14], [15], [16], [12], [17] and touchscreen devices [18], [13], [19].

Giancardo *et al.* [12] extract hold time sequences from the keystroke dynamics and train an ensemble of 200 support vector regression models. They calculate an nQi score for every 90-second typing window and produce a final PD motor index by computing the average of the window-level nQi scores over a longer typing sequence. To simplify the classification pipeline proposed in [12], in [14] a univariate model is created to capture the dynamic variation between consecutive keys. Applying the

FRESH [20] algorithm on the sequence of hold times, it calculates the mean absolute consecutive difference (MACD) feature that evaluates the dynamic variation between the consecutive keystrokes, which is important to capture transient bradykinesia and, in turn, significant to identify patients with Parkinson's. While in [14] the univariate model achieves an AUC score of 0.85, methods proposed in [15], [16], [18] approach the classification problem from a different perspective. T. D. Pham *et al.* [15] uses fuzzy recurrence plots to characterize the differences in routine keyboard interactions between Parkinson's and healthy control subjects. In [16] the importance of modeling higher-order interactions in improving the classification performance is discussed. It also reports the challenges involved in such a process in terms of execution time and storage. They approach the problem by converting the exponentially large tensors into a multi-linear problem and applying the FRESH [20] algorithm to extract the relevant features. Iakovakis *et al.* [18] analyze touchscreen typing patterns by representing each typing session as a combination of hold time, flight time, and normalized pressure sequences. It calculates the lower order and higher-order statistics for each of the sequences and trains a two-staged classification pipeline. T. Arroyo-Gallego *et al.* [13] calculate kurtosis and skewness for the higher-order statistics and co-variance to discriminate Parkinson's from healthy control subjects. In [19], D. Iakovakis *et al.* extract the hold time and flight time sequences from the touchscreen typing sessions with the constraint of a minimum of 40 elements in the sequence and trains a convolutional neural network to classify between Parkinson's and healthy control samples. Neil Dhir *et al.* [17] leverage a natural language processing technique to combine the content of the typed text with keystroke dynamics. They use a convolutional neural network long short-term memory network to detect classify PD in a dataset comprised of 230 subjects who transcribed a Wikipedia article.

The methods revised above show that keystroke dynamics are useful to identify PD typing patterns, however, there is still limited work indicating that keystroke dynamics can identify such PD patterns in an uncontrolled environment where data is acquired for long periods at subjects' homes. This is an unmet medical need that would need to be fulfilled to enable a transparent screening test for the general population. In Arroyo *et al.* [21], it is shown that the hold time-based model initially designed for in-clinic purposes [12] generalizes to a controlled patient population of 52 subjects monitored for 7 days achieving an AUC of 0.77 in the separation of PD versus healthy control typing patterns. While these are promising results, they lead to new questions about the generalizability of the model in longer periods, when subjects are bound to use their keyboard less consistently. Additionally, it is not clear how this model would generalize across age groups in larger datasets, which would resemble more clearly a screening setup where the majority of subjects would be PD-free and likely to compile a more heterogeneous representation of typing styles.

To this end, we compare and contrast published methods on 281 subjects with active typing data for at least 5 months, which, to our knowledge, is the largest dataset of this kind. One arm of the dataset was collected as part of a crowdsourced study with self-reported PD status, the other arm was composed of clinically recruited subjects whose PD diagnosis was verified by movement disorder experts and acquired in a study supported by The Michael J Fox Foundation for Parkinson's research (10860) [21]. In addition, we propose a convolutional neural network (CNN) model based on three types of signal representations, all derived from keystroke dynamics. These representations attempt to capture: joint hold time and flight time series, hold time and flight time series as a function of key types and asymmetry in the time series. We compare and contrast the results from analyzing these signal representations coupled with our CNN model with existing published methods that were initially developed for controlled environments, showing that our new approach outperforms the existing methods. Finally, we evaluate the contribution of each feature map for the final classification and how they vary as a function of time.

In summary, the main contributions of this work are:

Benchmark of published methods on a large keystroke dynamics PD dataset acquired by monitoring subjects for a period of 22 months and extracting ∼5 months of active typing data in an uncontrolled environment at the subjects' homes.

1) Introduction of a new type of keystroke dynamics features using hold time and flight time series as a function of key types and asymmetry in the time series.
2) Introduction of a simple CNN architecture that positively compares against other state-of-the-art methods.
3) Testing of the generalizability of the methodologies using two cohorts, one with clinically recruited patients and another with patients recruited with a crowdsource study.

II. Materials

The work presented in this paper uses the data collected longitudinally over an extended period of 22 months in the context of a study approved by the Massachusetts Institute of Technology, USA (Committee on the Use of Humans as Experimental Subjects approval no. 1412006804) and the University of Texas Health Science Center IRB (no. HSC-SBMI-17-0687). The data was collected by the neuroQWERTY software described in [12] and [21]. Once installed on the subject's laptop, the neuroQWERTY software runs in the background of the device collecting any interaction with the keyboard, regardless of the computer application context. The typing information recorded includes the timestamps for press and release events for every keystroke. In addition, the information about the type of keystroke, namely, right side key, left side key, space, backspace, and special key (e.g., SHIFT) is recorded, providing some level of information about the position of the fingers during the keyboard interaction, but greatly limiting potential privacy concerns.

Subjects in the dataset can be divided into two groups based on the source of their clinical information; subjects with a confirmed diagnosis by a movement disorder specialist (Confirmed diagnosis) and subjects who self-reported a diagnosis while signing up on the neuroQWERTY crowdsource study website (Unconfirmed diagnosis). The dataset produced by a longitudinal collection over 22 months contains 150 PD subjects, 1,216 controls, and 480 subjects who self-reported conditions other than PD, such as arthritis or dementia. We apply the following inclusion/exclusion criteria to select the subjects included in this analysis:

1) Subjects with a confirmed and self-reported diagnosis are included.
2) Subjects with active typing data of at least 5 months are included.
3) Subjects who self-reported conditions other than PD are excluded.

The above inclusion/exclusion criteria resulted in a total of 281 evaluable subjects. The demographic information of the subjects is shown in Table I.

In this work, we aim to evaluate the keystroke dynamics data collected during heterogeneous interactions with a mechanical keyboard, such as web browsing, email writing, word processing, game playing, or any other activity a computer user typically performs on their device. To preserve privacy, our data collection software did not record the applications used, nor the content of the text typed, however, we believe the data captured the majority of the possible typing activities that a user might perform on their computer by collecting their typing activity over a prolonged period and by discarding subjects that did not reach the activity threshold we set as valid weekly keyboard interactions.

We defined a valid weekly keyboard interaction as 300 keystrokes in a 7-day time window. To be included in the analysis, the threshold for valid weekly keyboard interactions was not enforced continuously over the observation period as it allowed for activity gaps of “non-valid” weekly keyboard interaction when there was no recorded user’s activity. This allowed simulating a real-world scenario, where subjects are not expected to continuously use their device over a prolonged number of months to enable the proposed PD screening test to work.

III. Methods

We start this section by describing existing methods that used keystroke dynamics on mechanical keyboards to detect parkinsonian signs. These methods will be benchmarked against our approach that couples a new feature representation, using hold time and flight time series as a function of key types and asymmetry, with a custom CNN model. We proceed by describing a general keystroke dynamic feature extraction and how the resulting data structures are converted to the specific Feature Maps used as input to the CNN model (GEN, COG, and ASYM Feature maps). Next, we describe the CNN architecture and how we evaluate the performance of the benchmarked methods. Finally, we describe our strategy to interpret the association between feature maps and PD predictions based on attribution maps.

A. Existing Approaches

Giancardo *et al.* [12] applied ensemble learning to the hold-time sequences to create a classification tool for differentiating the PD from healthy control subjects. Milne *et al.* [14] proposed a simpler feature extraction approach to classify PD and healthy control samples. Both manuscripts used the same dataset which was collected in the clinic and captured the mechanical keyboard typing data generated by subjects as they transcribed a folk tale that was randomly selected from a text sample collection. Subjects were allowed to correct their mistakes and were instructed to type as they would normally do at home. The dataset was included data from two groups namely, Early-PD (18 PD, 13 CNT) and De-novo (24 PD, 30 CNT). Subjects involved in the experiment typed over 14 minutes on average. For each subject data, the classification method divides the hold-time (HT) sequences in 90-second non-overlapping windows and generates that is transformed into a 7-dimensional feature vector. It trains an ensemble of 200 support vector regression models using the window-level feature vectors as input and the normalized version of the UPDRS-III [22] score as a supervised reference to generate a set of neuroQWERTY indices (nQi). The window-level nQi scores are grouped by computing the average of the scores obtained for the whole typing sample to generate a single score per subject. The AUC score achieved by this method is 0.81.

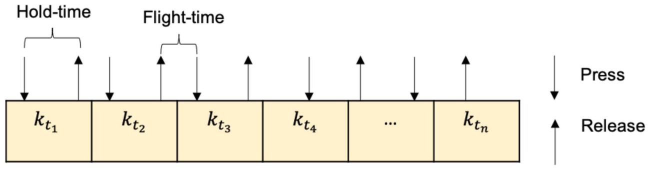

Fig. 1. Keystroke dynamics features.

Milne *et al.* [14] proposed and tested multiple methods on the same dataset. First, they show that by uniquely computing the standard deviation of the HT they achieved an AUC of 0.82. Then, they compute the mean of the absolute difference in HT between consecutive keystrokes, thereby obtaining another straightforward univariate model that achieves an AUC of 0.85. Finally, they leveraged the FRESH [20] algorithm to extract general-purpose time-series features and compute the 5 most predictive which are then used to train a logistic regression model that achieves an AUC of 0.80.

All of these methods have been developed on data coming from short typing sessions that might not be representative of the heterogeneity of multiple usage patterns of a computer. Only nQi was tested on a controlled population typing at home for 7 days [21]. However, it is still unknown how these metrics would perform over longer periods.

B. Keystroke Dynamics Feature Extraction

Let $K_{t_j} = k_{t_1}, k_{t_2}, k_{t_3}, \ldots, k_{t_j}, \ldots, k_{t_n}$ be the keystroke dynamics sequence recorded on a given subject. $'k_{t_j}'$ represents the key (k) recorded at time $'t_j'$ (j = position of the key in the keystroke sequence) and $'\mathrm{n}'$ represents the total number of keys in the keystroke sequence. Hold time (HT) is the time elapsed between the press and release event.

Flight time (FT) is the time elapsed between the release event of key ‘$k'_{t_{j-1}}$ and the press event of key $'k'_{t_j}$.

Fig. 1 shows how the keystroke dynamics features, hold time and flight time, are calculated from key press and release events. As the crowdsource data was collected over a prolonged period, with discontinuity between the typing sessions, we apply a threshold on the hold-time and flight-time sequences to remove any invalid values. We set the default thresholds for the hold time sequence values within the range [0.02, 0.8) and flight time sequence values within the range [-0.5, 5). They are inspired by earlier works [23] and [12] with a wider range. A wider range of thresholds ensures that we capture an adequate amount of typing activity in our experiments.

We use the preprocessed HT/FT sequences to calculate the features of interest. We resample the HT/FT sequence at a non-overlapping interval of 7-days. Let us assume we have a HT sequence $'ht'_X$ for subject 'X' calculated from the key press and release events recorded during the interaction of subject 'X' with the device keyboard.

$ht_X = ht_{\mathrm{X}1}\ ,\ ht_{\mathrm{X}2},\ ht_{\mathrm{X}3},\ \ldots,\ ht_{Xn};'\mathrm{n}'$ represents the last hold time value calculated for 'X'.

Let $H(ht_X)$ be the function that calculates the feature matrix from the hold-time sequences.

Being ht_{Xt1} the sequence start point recorded at time $t1$, $H(ht_X)$, computes the probability density function of the keys falling within the 7-day interval following $t1$. The resulting probability density function is then transformed into a 20-bin histogram using a 20-dimensional sampling vector in HT range 0 to 0.4 seconds.

$$H\ (ht_{X1}) = \begin{cases} hist\left(pdf\left(ht_{X1}, ht_{X2}, \ldots, ht_{Xk}\right)\right),\ len\left(samples\right) > 300 \\ \qquad\qquad none,\ otherwise \end{cases} \tag{1}$$

The HT sequence is resampled by selecting three different start points in the time axes (corresponding to the sequence of recorded keys. This resampling method is applied to account for the lack of temporal consistency in users' keyboard interactions over the observation period. Thus, resampling at three random start points helps us better simulate the concept of uncontrolled data acquisition. After creating the histogram vectors for each of the key sets in the 7-day interval over three random starting points, we sort the histogram vectors to match their natural order on the time axis.

The sorted histogram vectors are concatenated in their natural order creating a feature matrix of shape $(h_X x20)$, where width is 20 and height is h_X for subject X. Since subjects were not restricted to any specific level and duration of typing activity, the resulting height of the feature matrix varies among subjects. Finally, we transpose the output feature matrix into a final structure of shape $(20\ x\ h_X)$.

We perform the same feature extraction process on the FT sequence such that $F(ft_X)$ is the function that calculates the FT feature matrix. Given

$ft_X = ft_{X1}\ ,\ ft_{X2},\ ft_{X3},\ \ldots,\ ft_{Xn};'\mathrm{n}'$ represents the last FT value calculated for 'X'. ft_1 is the sequence start point recorded at time $t1$,

$$F\ (ft_{X1}) = \begin{cases} hist\left(pdf\left(ft_{X1}, ft_{X2}, \ldots, ft_{Xk}\right)\right),\ len\left(samples\right) > 300 \\ \qquad\qquad none,\ otherwise \end{cases} \tag{2}$$

Where 'k' is the position of the last available element in the interval between $t1$ and $t1+$ 7 days, '*pdf*' represents the probability density function, $'hist'$ calculates the histogram and 'len' denotes the length of the samples.

We used $h_X =\ 21$ which would be equivalent to ~5 months of active typing. This value was chosen before any method development, to balance the need of including a sufficient number of subjects in the experiment and a long enough activity observation period to capture the heterogeneity of the computer activities performed by the subjects.

As some of the subjects provided data over a period that is longer than the interval established by the h_X threshold, we apply data augmentation as follows (Fig. 2):

1) If a subject's output feature matrix's width is equal to h_X , we use that subject without any further processing.
2) If a subject's output feature matrix's width is greater than h_X, we divide the feature matrix into blocks of shape $(h_X x21)$. In the event where the block is not a multiple of 21, we take the floor value of the integer division of the width of the matrix and the desired width h_X. The floor value represents the number of blocks accepted from the subject's feature matrix.
3) Every HT/FT sequence converted to these blocks can be seen as a list of fixed shape matrices for that sequence.
4) Once all the sequences have been converted to such blocks for any feature map, matrices with the same indices from across the sequences are stacked along the depth dimension. The final dimension of every feature map is decided by this process of stacking. Table II summarizes the HT/FT sequence as the function of the key types of interest (KTI).

C. GEN, COG, ASYM Feature Maps

Table II summarizes the three feature maps proposed in this work: GEN, ASYM, and COG. These have been designed to capture specific facets of PD signs in the keystroke dynamics patterns.

Motivated by the known effects of bradykinesia (i.e., generalized slowness of movement) in PD motor skills [2], the GEN feature map is constructed from all the key press and release events present in the keystroke signal. Thus, the HT sequence for the GEN feature map is computed as $ht_X = ht_{\mathrm{X}1}\ ,\ ht_{\mathrm{X}2},\ ht_{\mathrm{X}3},\ \ldots,\ ht_{Xn}$ where, $'\mathrm{n}'$ denotes the last hold time value calculated for subject $'\mathrm{X}'$. Using this HT sequence, we follow the feature extraction explained in Section B to construct the GEN feature map. Similarly, the FT sequence is defined as $ft_X = ft_{X1}\ ,\ ft_{X2},\ ft_{X3},\ \ldots,\ ft_{Xn}$ where, $'\mathrm{n}'$ denotes the last FT value calculated for subject $'\mathrm{X}'$. By including all keys in the entire sequence, we aim to generate a global representation of the impact of PD motor impairment in the keystroke patterns.

Motivated by the characteristic asymmetrical onset PD [24], we construct the ASYM feature map as a function of the key type information. As shown in Table II, this feature map employs the HT sequence calculated from two specific key type sets, namely right (rg) and left (lf). It calculates the absolute difference between the HT of the right and left keys. For the

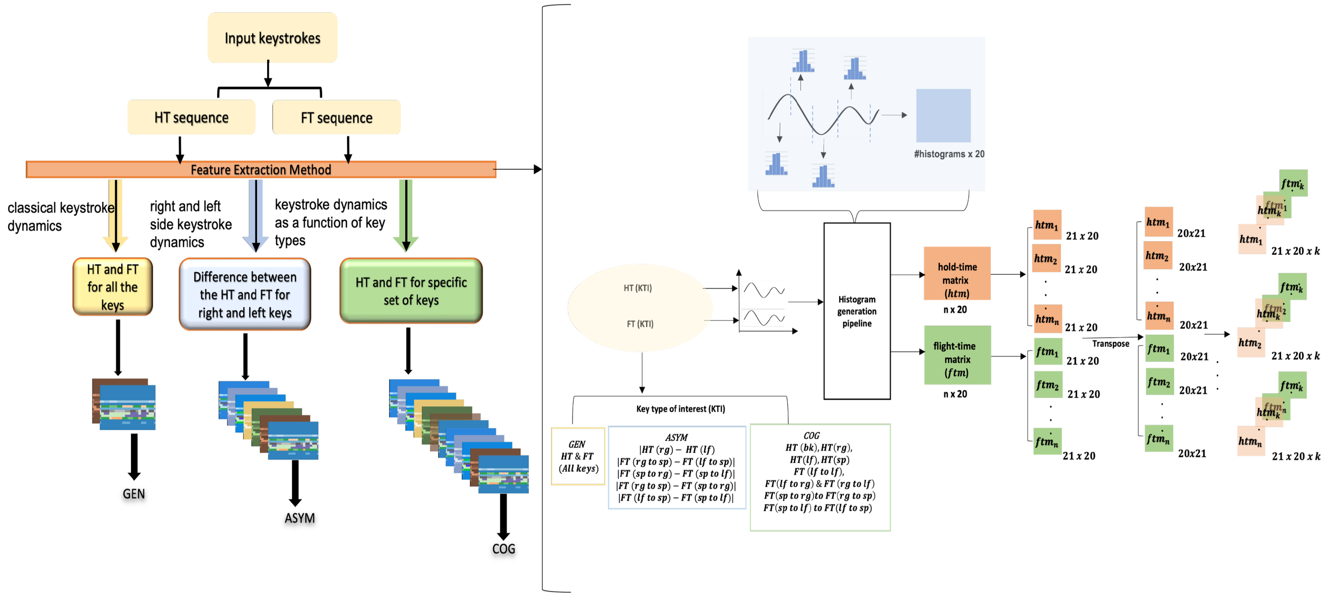
Input keystrokes
HT sequence
FT sequence
Feature Extraction Method
classical keystroke dynamics
right and left side keystroke dynamics
keystroke dynamics as a function of key types
HT and FT for all the keys
Difference between the HT and FT for right and left keys
HT and FT for specific set of keys
GEN
ASYM
COG
#histograms x 20
HT (KTI)
FT (KTI)
Histogram generation pipeline
hold-time matrix (htm)
n x 20
flight-time matrix (ftm)
n x 20
Key type of interest (KTI)
GEN
HT & FT
(All keys)
ASYM
|HT (rg) − HT (lf)
|FT (rg to sp) − FT (lf to sp)|
|FT (sp to rg) − FT (sp to lf)|
|FT (rg to sp) − FT (sp to rg)|
|FT (lf to sp) − FT (sp to lf)|
COG
HT (bk), HT(rg),
HT(lf), HT(sp)
FT (lf to lf),
FT(lf to rg) & FT (rg to lf)
FT(sp to rg)to FT(rg to sp)
FT(sp to lf) to FT(lf to sp)
htm1 21 x 20
htm2 21 x 20
htmn 21 x 20
ftm1 21 x 20
ftm2 21 x 20
ftmn 21 x 20
Transpose
htm1 20x21
htm2 20x21
htmn 20x21
ftm1 20x21
ftm2 20x21
ftmn 20x21
21 x 20 x k
ftmk
htmk

Fig. 2. The figure above shows the complete pipeline of the feature extraction process. The HT/FT sequences are used to generate the features maps GEN, ASYM, and COG using the key type of interest (KTI). The feature extraction method is expanded and shows the transformation of HT/FT signals to the binned probability densities (20) in the form of matrices. These matrices are broken down into fixed size matrices of dimension 21 x 20. The fixed size hold-time matrices (htm) are stacked with the fixed size flight-time matrices (ftm) along the depth. The stacking is performed for every time a HT and FT sequence is processed, determining the depth of the final feature matrices.

TABLE I
SUBJECTS CHARACTERISTICS

Unconfirmed Diagnosis	PD	Controls	Statistical Significance
n (total n=217)	19	198	N/A
Women # (%)	2	75	not sig. p=1.00
Unknown # (%)	1	0	not sig. p=1.00
Avg. Age (std)	60.158 (13.897)	38.949 (13.199)	***p<0.001
Avg. Num. of Keypresses (std)	470,443 (613,049)	680,935 (938,572)	not sig. p=0.18
Avg. Num. of Days with Active Typing (std)	139.3 (111.8)	144.2 (111.3)	not sig. p=0.82
Avg. Hold-Time (std)	0.127 (0.03)	0.11 (0.022)	**p<0.01
Avg. Flight-Time (std)	0.6 (0.186)	0.413 (0.129)	***p<0.001
Confirmed Diagnosis			
n (total n=64)	31	33	
Women # (%)	17	18	not sig. p=1.00
Unknown # (%)	0	0	N/A
Avg. Age (std)	60.6 (12.0)	60.8 (10.7)	not sig. p=0.72
Avg. Num. of Keypresses (std)	549,580 (781,514)	366,743 (282,573)	not sig. p=0.69
Avg. Num. of Days with Active Typing (std)	148.2 (74.2)	142.6 (59.8)	not sig. p=0.95
Avg. Hold-Time (std)	0.129 (0.028)	0.105 (0.028)	**p<0.01
Avg. Flight-Time (std)	0.522 (0.194)	0.517 (0.26)	not sig. p=0.35
n (total n=64)	31	33	
Women # (%)	17	18	not sig. p=1.00

The evaluation of the statistical significance as calculated using the Mann-Whitney U shows the statistical difference between the two groups namely, Parkinsons and Controls with respect to hold-time, flight-time, and age in the combined dataset. The confirmed diagnosis refers to the subjects who were diagnosed by clinical experts and unconfirmed diagnosis refers to people who self-reported their diagnosis.

TABLE II
FEATURE MAPS GEN, COG, AND ASYM WITH KEY SUBSETS

Feature Map	Hold Time	Flight Time
GEN	HT (all)	FT (all)
COG	HT (bk)	FT (bk to bk)
COG	HT (rg)	FT (rg to rg)
COG	HT (lf)	FT (lf to lf)
COG	HT (sp)	FT (lf to rg, rg to lf)
COG		FT (sp to rg, rg to sp)
COG		FT (sp to lf, lf to sp)
ASYM	abs (diff ((HT (rg), HT (lf)))	abs (diff (FT (rg to sp), FT (lf to sp)))
ASYM		abs (diff (FT (sp to rg), FT (sp to lf)))
ASYM		abs (diff (FT (rg to sp), FT (sp to rg)))
ASYM		abs (diff (FT (lf to sp), FT (sp to lf)))

Key subsets are used to derive the HT and FT sequences which are then used to construct the final feature maps. $abs(.)$ represents the absolute operation and $diff(.)$ represents the difference between the values.

FT sequence, we use the right (rg), left (lf), and space (sp) keys. Then, we calculate the absolute difference between the FTs values for each possible combination of key type pairs. For example, we calculate the absolute difference between the FT from right to space and FT from left to space. The HT and FT absolute differences are then used to build the ASYM feature map following the process described in Section B.

Parkinson's can cause cognitive decline in subjects [25], this fact motivated us to create a feature map focused on identifying the patterns in motor skills mediated by cognition. The COG feature map derives the HT sequences from backspace (bk), right

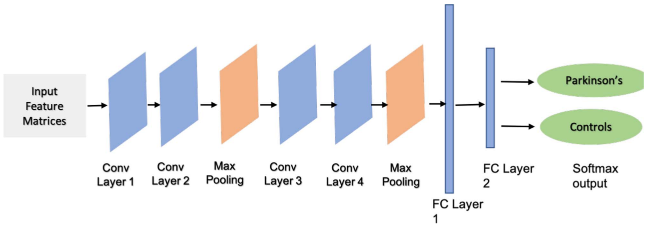

Fig. 3. The convolutional neural network architecture consists of 4 convolutional layers namely, Conv Layer 1, 2, 3, and 4. Conv Layer 1 and 3 have a filter of height 2 and width 3. Conv Layer 2 and 4 have a filter of height 2 and width 5. The output classes are Parkinson's and Control.

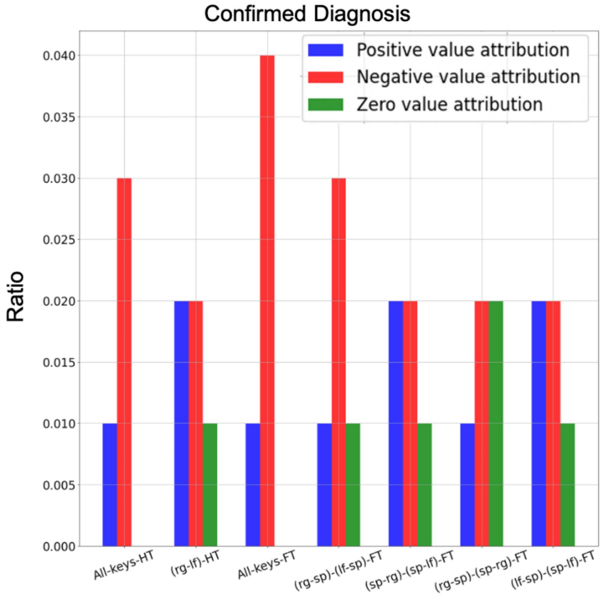

Fig. 4. The attributions (positive, negative, and zero) for the confirmed diagnosis set. The positive (blue) attributions are obtained after setting a threshold of greater than 0 in the attribution matrix. Similarly, negative attributions (red) are generated after setting a threshold of less than 0 in the attribution matrix. Zero valued attributions are the values in the attribution matrix equal to 0. "All-keys-HT" and "All-keys-FT" do not have any zero-valued attributions since they project the sum of attributions from all the keys present in the set and thus unless all the attributions are 0 the total of the attribution is a non-zero value. Attribution matrix is obtained using integrated gradients [26].

TABLE III
Features Maps Used in Each Model

	Feature Map			
Model	GEN	COG	ASYM	CNN Parameters
GEN-NET	Y	N	N	1,425,818
GEN-ASYM-NET	Y	N	Y	1,426,778
GEN-COG-NET	Y	Y	N	1,427,738
GEN-ASYM-COG-NET	Y	Y	Y	1,428,698

The total number of trainable parameters in each of the models which increases as we increment the number of individual features maps used to construct each model input.

(rg), left (lf), and space (sp) keys. The FT sequences consist of FT from keys backspace to backspace, right to right, left to left, left to right and right to left, space to right and right to space, and space to left and left to space. The HT and FT sequences are used to build the COG feature map using the process described in Section B.

D. CNN Architecture

Using the GEN, COG, and ASYM Feature Maps, we train a custom CNN to classify PD from control subjects as shown in Fig. 3. The boundaries of the input matrix are replicated to pad the input matrix before feeding it to the CNN. The CNN architecture consists of 2D convolutional layers. The convolutional kernel size for layers 1 and 3 is $(2x3)$ and for layers 2 and 4 is $(2x5)$ respectively. We construct a unit such that a convolutional kernel $(2x3)$ is followed by a $(2x5)$ kernel followed by a $(2x2)$ max pooling and ReLU activation layer. The CNN consists of two of such units. The choice of the filter size was dictated by the need of creating filter kernels able to capture typing activity patterns at different time scales (represented by the second dimension of the filter) and at the same time smooth neighboring values in the first filter kernel dimension.

The output of the 2nd unit is flattened and passed through two linear layers. A SoftMax layer is placed after the 2nd linear/fully connected layer to generate the probability scores for the two classes, namely PD (Parkinson's) and CNT (control). We use stochastic gradient descent as optimizer with a 0.0001 learning rate and a binary cross-entropy loss function. The model was trained in 700 epochs.

The four models proposed in this work differ on the type and shape of the input feature map they use. A GEN feature map of shape $(20x21x2)$ that comprises of the hold-time and flight-time matrix for all the key types is used as the input feature map in the GEN-NET model. Likewise, the model GEN-COG-NET inputs a structure of shape $(20x21x12)$ that is obtained by combining the GEN and COG feature maps. The GEN-ASYM-NET input shape is $(20x21x7)$, which is obtained as a combination of GEN and ASYM feature maps. Likewise, GEN-ASYM-COG-NET input shape is (20x21x17) obtained by combining GEN, ASYM, and COG feature maps. Here, the process of combining the features is done by stacking the feature matrices along the depth dimension to create a 3-D input feature matrix in the case of GEN-COG-NET, GEN-ASYM-NET, and GEN-ASYM-COG-NET.

The height of the feature matrices represents the 20-bin histogram representing the weekly keystroke probability density functions, whereas the width represents the longitudinal 7-day block intervals of key interactions captured over the observation period. The two units of convolution operation followed by max pooling and an activation layer differ in the width of the convolution kernel.

The total number of trainable parameters is shown in Table III. As apparent from the combination of the feature maps, the number of parameters increases in the order of GEN-NET, GEN-ASYM-NET, GEN-COG-NET, and GEN-ASYM-COG -NET.

E. Evaluation

We train and evaluate the models on the crowdsource study (unconfirmed diagnosis arm) using a 5-fold cross-validation strategy where each fold contains a train/validation and test split. Then, we use the clinically recruited patients (confirmed diagnosis arm) as a fully independent test set.

Since we use data augmentation for our training set which differs from that of our test set, the total number of subjects generated for training from the unconfirmed diagnosis after augmentation is 869. Within every fold, the training data is partitioned into test set and train set. The train set is further split into 75% training data and 25% validation data. Thus, within every fold, the actual number of training data is composed of 398 data samples. The loss on the validation set was used to employ the early stopping. The model training is stopped after 20 epochs if there is no improvement in the validation loss. This helps to prevent the overfitting of the training set. We use mini-batch stochastic gradient descent to optimize our models. The batch size used for training is 70 subjects at a time, empirically selected during the experiments.

Because of the stochastic nature of our models, we observed some deviations in the performance of the models. Thus, we repeated the process 30 times. In Table IV we show the performance of the proposed models' performance compared to the models presented in [12], [14]. We report the average (standard deviation) of the performance of the models for the separate sets namely Confirmed Diagnosis and Unconfirmed Diagnosis. We combine the two sets and report a weighted average and standard deviation. The weights are calculated as the proportion of subjects present from each of the sets in the combined set.

Finally, we use integrated gradients [26] to calculate the attribution maps in order to understand our CNN's decisions.

TABLE IV
CLASSIFICATION PERFORMANCE COMPARISON

	Method	Confirmed Diagnosis, n=64 AUC (std)	Unconfirmed Diagnosis, n=217 AUC (std)	Confirmed & Unconfirmed Diagnosis n=281 AUC (std)
Baseline	Logistic Regression	0.64(0.038)	0.81(0.119)	0.71(0.117)
CNN Based	GEN-ASYM-NET	0.80(0.037)	0.83(0.077)	0.81(0.058)
	GEN-COG-NET	0.75(0.025)	0.81(0.121)	0.77(0.084)
	GEN-NET	0.76(0.017)	0.7(0.135)	0.73(0.09)
	GEN-ASYM-COG-NET	0.76(0.029)	0.82(0.105)	0.78(0.075)
Other methods	nQi [12]	0.71(0.038)	0.73(0.094)	0.72(0.067)
	Stdev [14]	0.78(0.0)	0.7(0.121)	0.75(0.087)
	FRESH [14]	0.66(0.052)	0.71(0.137)	0.68(0.094)
	MACD [14]	0.82(0.0)	0.64(0.122)	0.75(0.117)

The table shows the AUCs (average (standard deviation)) for each of the models proposed in this work (CNN based and Baseline) and the reference models from the literature [12], [14]. For every model, the AUCs have been averaged over the 5-folds. Since each experiment is repeated 30 times, we also average the AUCs over these 30 iterations. The column Confirmed Diagnosis refers to subjects with a PD diagnosis confirmed by an expert, whereas in the Unconfirmed Diagnosis set, clinic-demographic information was self-reported. The performance of the models on the combined set of Confirmed and Unconfirmed Diagnosis is reported in the rows of the last column. There is a difference in the number of subjects in Confirmed and Unconfirmed set. Thus, we report the weighted average and weighted standard deviations for the combined set. The weights reflect the proportion of subjects for each type of diagnosis in the combined set.

IV. RESULTS

This section presents the results of the proposed methods compared to the results achieved using the methods described in [12] and [14]. The evaluation approach is the same for both proposed and literature methods. We successfully reproduce the nQi results as reported in [12] and the FRESH-based approach described in [14]. Table IV shows the average AUC (standard deviation) for each of the methods. We first introduce a baseline approach that calculates the average hold-time and flight-time for each subject and performs binary classification. This baseline approach achieves an average AUC of 0.64 on confirmed and 0.81 on unconfirmed diagnosis cohort. Thus, we create feature sets that serve as inputs to the four of our models namely, GEN-ASYM-NET, GEN-COG-NET, GEN-NET, and GEN-ASYM-COG-NET. We compare the performance of the four proposed models in Fig. 8, both in the confirmed and unconfirmed diagnosis sets using a receiver operating characteristic curve (ROC) analysis. GEN-ASYM-NET shows the best overall performance with an average AUC of 0.8 in the confirmed diagnosis set and 0.83 in the unconfirmed diagnosis set.

As mentioned above, we use the nQi [12] and MACD [14] methods as anchor references. MACD applies the FRESH algorithm to compute a univariate feature set that is used for the binary classification of PD versus healthy control subjects. When we run the same method using our evaluation strategy, MACD obtains an average AUC of 0.82 on confirmed and 0.64 on unconfirmed diagnosis sets. GEN-ASYM-NET performs consistently better on both subject types when compared to MACD. Similar consistency is observed in the models GEN-COG-NET, GEN-NET, and GEN-ASYM-COG-NET. nQi proposes a feature set that is relatively more complex than the approach used in MACD. When replicating our evaluation pipeline, the nQi method obtains an average AUC of 0.71 on confirmed and 0.73 on unconfirmed diagnosis sets. Although nQi's performance is consistent in both subject sets, it shows significantly lower performance than the one achieved by our models.

To better understand the underlying patterns captured by our best-performing model we construct attribution maps. The attribution maps were calculated for GEN-ASYM-NET as it was the best performing out of the proposed models for both the unconfirmed subjects and confirmed subjects. We outline some of the important details from the attribution maps that we have

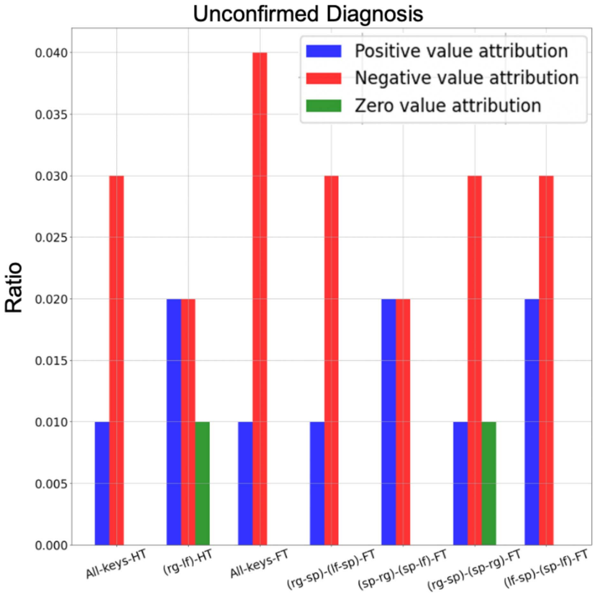

Fig. 5. The attributions (positive, negative, and zero) for the unconfirmed diagnosis set. The positive (blue) attributions are obtained after setting a threshold of greater than 0 in the attribution matrix. Similarly, negative attributions (red) are generated after setting a threshold of less than 0 in the attribution matrix. Zero valued attributions are the values in the attribution matrix equal to 0. "All-keys-HT" and "All-keys-FT" do not have any zero-valued attributions since they project the sum of attributions from all the keys present in the set and thus unless all the attributions are 0 the total of the attribution is a non-zero value. Attribution matrix is obtained using integrated gradients [26].

observed in our experiments. Fig. 5 and Fig. 6 represent projections of the positive valued (blue), negative valued (red), and zero-valued (green) attributions representing the individual contributions of each of the feature maps for confirmed and unconfirmed diagnosis sets, respectively. The attributions presented here are normalized to produce the ratio of positive, negative, and zero contributions. For the features that include hold time and flight time for all types of key categories, we observe an overall high negative contribution towards the predictions in confirmed and unconfirmed diagnosis cohorts (Fig. 5 and Fig. 6). Whereas, for the feature derived from the absolute difference between the flight time from right key to space key category and flight time from left key to space key category, the positive and negative contribution is at the same level for the confirmed and unconfirmed diagnosis. However, for the same set of feature maps, we do not see any zero contributions in the unconfirmed diagnosis, unlike what we see in the confirmed diagnosis arm. The highest negative contributions are seen on the flight time of all key types for unconfirmed diagnosis. There are no zero attributions under the features represented by $(sp-rg)-(sp-lf)-FT$ and $(lf-sp)-(sp-lf)-FT$ for the unconfirmed diagnosis unlike what we see in the confirmed diagnosis set. An interesting observation is that there are no zero attributions in the "All-keys-HT" and "All-keys-FT", which implies that every element in these feature maps contributes (positively or negatively) towards the predictions. These numbers can help us create an overall picture of how the feature maps are working on their own even when they are jointly used as input to the model.

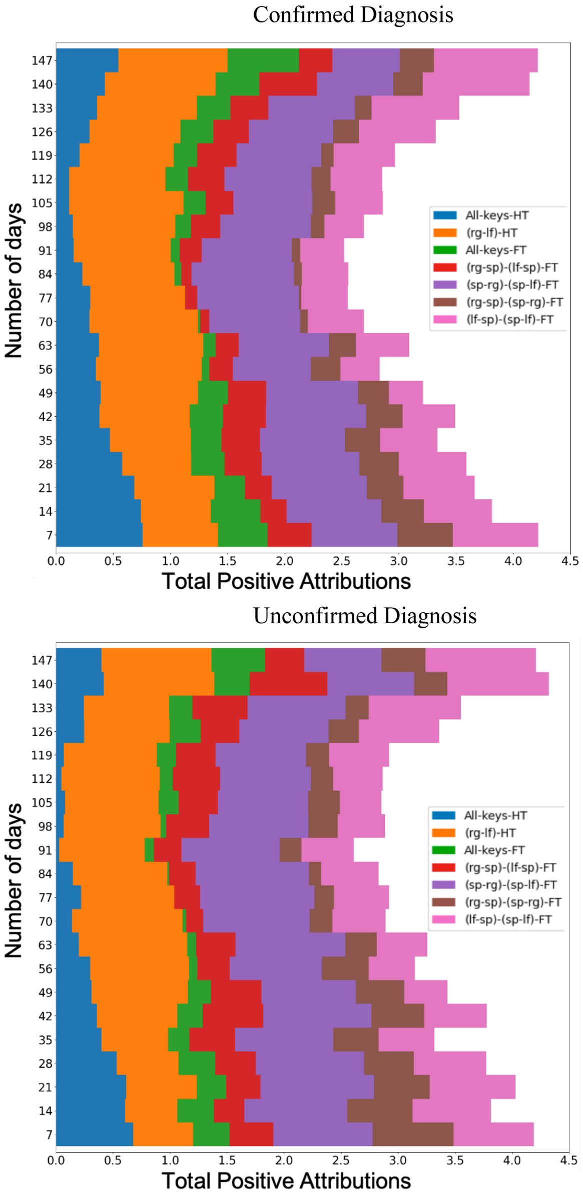

Fig. 6. The figure shows a stacked bar chart for the positive attributions for each of the valid windows summed across all subjects in both the confirmed and unconfirmed diagnosis arms. The feature representations used to train GEN-ASYM-NET are color-coded and labeled as shown in the figure legend. The y-axis represents the number of days over which the keystroke activity was evaluated. The x-axis represents the values of positive attributions summed across all subjects. The positive attributions are obtained after setting a threshold of greater than 0 in the attribution matrix obtained after applying integrated gradients [26] on the input feature matrix. Positive attributions represent the components of the feature matrix that contribute positively during training.

Fig. 7 shows the positive attributions for each feature map type. The x-axis represents the total number of positive attributions and the y-axis represents the number of days over the $\sim$5-month observation period (154 days). Note that each feature map has a 0 to 1 range.

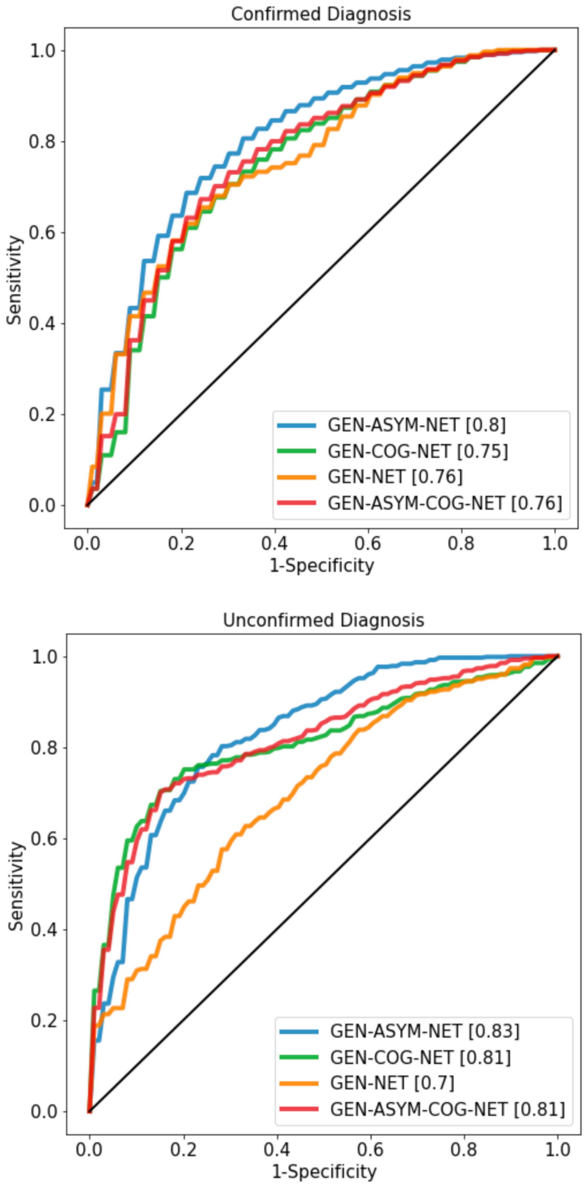

Fig. 7. Receiver operating characteristic curves (ROC) showing the comparison of the performances of methods GEN-COG_NET, GEN-ASYM-NET, GEN-NET, presented in the paper. The performance of the methods shown in the figure above is evaluated by averaging the AUC scores over 5-folds and 30 repetitions for the unconfirmed and confirmed diagnosis sets. After averaging the AUC scores, we see the GEN-ASYM-NET outperforms the rest of the proposed methods.

From Fig. 7 we can observe the feature matrix "$All - keys - ht$" (blue) is somewhat less curved when compared to the positive attributions of the other feature maps. (rg-lf)-HT feature type holds the greater number of positive attributions in unconfirmed diagnosis in comparison to confirmed diagnosis cohorts. (rg-sp)-(lf-sp)-FT (red) holds greater positive contributions in confirmed diagnosis when compared to the unconfirmed diagnosis. (rg-sp)-(sp-rg)-FT (brown) has greater positive attribution in the confirmed than in unconfirmed cohorts. (sp-rg)-(sp-lf)-FT (purple) is the only feature set that appears to hold almost the rate of positive attributions for confirmed and unconfirmed diagnosis cohorts. The positive attributions take a U-like shape over the 5-month period represented by the y-axis, which indicates that the beginning and the end of the keystroke dynamic signals have more weight in the classifier decision.

V. Discussion

While clinical diagnosis of PD will inevitably require a clinical expert to examine the patient, easily deployable tools that could deliver an initial PD screening test at home will be an essential part for developing and delivering disease-modifying therapies. Techniques based on the remote and passive monitoring of keystroke dynamics, in conjunction with other approaches to confirm the PD signs, have the potential of playing a key role in such screening, given their extremely low burden on the subject and the ease of data acquisition.

In this work, we generate a set of feature maps to capture PD relevant features in natural keystroke data monitoring subjects for a period of 22 months and extracting ~5 months of active typing data in an uncontrolled environment at the subjects' homes. The feature maps use a combination of key hold-time and flight-time sequences calculated form the typing activity of the subjects. Hold time is considered typically independent of typing skills as it is chiefly influenced by subcortical mechanisms [12]. This attribute of the hold time dynamics can be leveraged to model bradykinesia, a generalized slowness of movement characteristic of early PD that be reflected in the press-release intervals as the patient is typing. Flight time defined as the elapsed time between releasing a key and pressing the next key can also help us understand inter-digit dynamics and fine motor coordination as finger transition from key to key. Their complementary characteristics justify their inclusion in the feature maps. We train four CNN-based models that use different combination of input feature maps. GEN-ASYM-NET outperforms the other three models. After further evaluation of GEN-ASYM-NET using the ratio of the positive, negative, and zero attributions we observed that some of the constituents of the ASYM feature map contribute equally, positively, and negatively towards the classification process. Whereas the same is not true about the GEN feature map as the negative attribution ratio takes the lead. This indicates the usability of ASYM feature map in learning potential patterns that can detect PD signs. In addition to the ratio of the positive, negative, and zero attributions, we narrow down our understanding of the patterns by looking into the positive attributions over ~5 months of active typing data. We observe more positive attributions towards the beginning and the end of the typing data. A possible explanation could be due to the progressive neurodegenerative nature of PD, which can be better detected by focusing the classifier's attention on changes between start and end of data acquisition. In summary, the attribution ratios, and the positive attribution scores for both the cohorts exhibit similar behavior. This strengthens our belief that GEN-ASYM-NET is not overfitting to the training data (unconfirmed diagnosis) rather is learning patterns that distinguish PD from an HC.

In addition to the a cross-validation strategy used for the unconfirmed diagnosis cohort, we kept the confirmed diagnosis cohort as external testing set. This allowed us to test the generalizability of the models with a cohort with less confounding variables and ensuring that no data leakage was present during the cross-validation procedure. Also, it proves that even that data acquired with a crowdsource strategy can be an excellent source to be used for training purposes.

In addition to test our new feature representation coupled with a CNN model, we experimented with other models previously presented in the literature specifically designed for keystroke dynamics with mechanical keyboards for PD signs detection. A comparison to the MACD [14] reference models reveal the necessity for feature representation beyond standard statistical features to maintain the performance consistency over both cohorts. Compared to MACD, which proposes a univariate feature approach, our feature set is designed to capture richer feature patterns from the keystroke dynamics data. MACD's feature works well in the confirmed diagnosis cohort, which may be due to the limited demographic and typing homogeneity observed in this subject group. Whereas its poor performance on the unconfirmed diagnosis can be attributed to the fact that a single feature is not able to capture the heterogeneity of the cohort captured with a crowdsource data acquisition design. The nQi [12] model does not suffer from the inconsistency over both types of data. However, its lower classification performance compared to the proposed methods might suggest that it could benefit from the additional information captured by the GEN-ASYM-NET-based model.

We utilize ∼5 months of active typing data collected in an uncontrolled environment to build our models. Whereas MACD and nQi in their original work use a dataset collected with much shorter typing length. Prior to evaluating their performance on our dataset, we retrain them using our training set. This provides an unbiased evaluation and comparison to our models and indicates that GEN-ASYM-NET achieves better performance overall.

There are limitations associated with our work. In this work, as other approaches based on keystroke dynamics, PD signs affecting lower limbs are unlikely to be captured, still, it can be complementary to other systems focusing specifically on lower limbs that are typically based on Inertia Measurement Units. Another limitation revolves around our sample of 281 subjects, which might still not be not enough be a representative of the general populations, however, by passively monitoring keystroke dynamics for such a long period of time, we are likely to have captured a significant amount typing styles and computer activities, such as word processing, playing browsing the web, etc., not previously recorded in other studies, as it is shown by the lower performance of other methods that we find in our dataset compared to the original publications. The crowdsource cohort has inherent limitations: all subjects self-reported their diagnosis and age is a potential confounder. However, our testing strategy used the confirmed diagnosis cohort (not used during the training phase) which has well matched PD / control design and a diagnosis confirmed by a movement disorder specialist both for PDs and Control subjects. As such, a model that overfits potential confounding variables like age would have good performance on the unconfirmed diagnosis cohort and poor performance on the confirmed diagnosis cohort. In our analysis, the only approach showing this behavior is the baseline logistic regression model. Another limitation of the crowdsource data is that we cannot be sure of its accuracy due to its underlying population of self-reported (unconfirmed diagnosis cohort). However, our model trained on this population does perform well on the confirmed diagnosis cohort, which indicates its potential usefulness regardless of the noise in the data.

Future work will include a greater number of subjects in our experiments and inclusion of other diseases that might affect keystroke dynamics and testing models using self-supervised or semi-supervised approaches and integration of other wearable sensors.

VI. Conclusion

We propose a benchmark of published methods on a large keystroke dynamics PD dataset acquired by monitoring subjects for a period of 22 months and extracting ∼5 months of active typing data in an uncontrolled environment at the subjects' homes. Then, we introduce of a new type of keystroke dynamics features using hold time and flight time series as a function of key types and asymmetry in the time series coupled with a simple CNN architecture. Finally, we test of the generalizability of the methodologies using two cohorts, one with clinically recruited patients and another with patients recruited with a crowdsource study finding that our approach positively compares against other methods previously tested on keystroke dynamics data acquired with mechanical keyboards.

Acknowledgment

We would like to thank the Michael J Fox Foundation for Parkinson's research who supported in part the data acquisition with grant number 10860 and acknowledge the whole neuroQWERTY team at the Massachusetts Institute of Technology and Hospital 12 de Octubre, Spain for the creation of the dataset used in this work. Specifically, we would to thank Martha L. Gray, Alvaro Sánchez-Ferro, Ian Butterworth, Carlos S. Mendoza, Raúl San José Estépar, Maria J. Ledesma-Carbayo, Peter Hansen, Petra Krauledat, Paloma Montero, Veronica Puertas-Martin and Michele Matarazzo. Luca Giancardo is also an inventor on a patent currently licensed to nQ Medical Inc. in the same general research area of this work

References

[1] E. R. Dorsey et al., "Global, regional, and national burden of Parkinson's disease, 1990–2016: A systematic analysis for the global burden of disease study 2016," *Lancet Neurol*, vol. 17, no. 11, pp. 939–953, Nov. 2018, doi: 10.1016/S1474-4422(18)30295-3.

[2] F. Paolini Paoletti, L. Farotti, and L. Parnetti, "Progression of symptoms in Parkinson's disease," in *Diagnosis and Management in Parkinson's Disease*. New York, NY, USA: Elsevier, 2020, pp. 3–20, doi: 10.1016/B978-0-12-815946-0.00001-6.

[3] A. E. Lang, "Clinical trials of disease-modifying therapies for neurodegenerative diseases: The challenges and the future," *Nature Med.*, vol. 16, no. 11, pp. 1223–1226, Nov. 2010, doi: 10.1038/nm.2220.

[4] N. F. Haq et al., "Parkinson's disease detection from fMRI-derived brainstem regional functional connectivity networks," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, vol. 12267, A. L. Martel et al., Eds., New York, NY, USA: Springer, 2020, pp. 33–43, doi: 10.1007/978-3-030-59728-3_4.

[5] S. K. Khoo et al., “Plasma-based circulating MicroRNA biomarkers for Parkinson’s disease,” *J. Parkinson’s Dis.*, vol. 2, no. 4, pp. 321–331, 2012, doi: 10.3233/JPD-012144.

[6] S. Manne et al., “α-Synuclein real-time quaking-induced conversion in the submandibular glands of Parkinson’s disease patients,” *Movement Disord.*, vol. 35, no. 2, pp. 268–278, Feb. 2020, doi: 10.1002/mds.27907.

[7] J. Camps et al., “Deep learning for freezing of gait detection in Parkinson’s disease patients in their homes using a waist-worn inertial measurement unit,” *Knowl.-Based Syst.*, vol. 139, pp. 119–131, Jan. 2018, doi: 10.1016/j.knosys.2017.10.017.

[8] T. T. Pham *et al.*, “Freezing of gait detection in Parkinson’s disease: A subject-independent detector using anomaly scores,” *IEEE Trans. Biomed. Eng.*, vol. 64, no. 11, pp. 2719–2728, Nov. 2017, doi: 10.1109/TBME.2017.2665438.

[9] R. A. Shirvan and E. Tahami, “Voice analysis for detecting Parkinson’s disease using genetic algorithm and KNN classification method,” in *Proc. 18th Iranian Conf. Biomed. Eng.*, 2011, pp. 278–283, doi: 10.1109/ICBME.2011.6168572.

[10] E. Vaiciukynas et al., “Detecting Parkinson’s disease from sustained phonation and speech signals,” *PLoS One*, vol. 12, no. 10, Oct. 2017, Art. no. e0185613, doi: 10.1371/journal.pone.0185613.

[11] L. Giancardo et al., “Psychomotor impairment detection via finger interactions with a computer keyboard during natural typing,” *Sci. Rep.*, vol. 5, no. 1, Sep. 2015, Art. no. 9678, doi: 10.1038/srep09678.

[12] L. Giancardo et al., “Computer keyboard interaction as an indicator of early Parkinson’s disease,” *Sci. Rep.*, vol. 6, no. 1, Dec. 2016, Art. no. 34468, doi: 10.1038/srep34468.

[13] T. Arroyo-Gallego et al., “Detection of motor impairment in Parkinson’s disease via mobile touchscreen typing,” *IEEE Trans. Biomed. Eng.*, vol. 64, no. 9, pp. 1994–2002, Sep. 2017, doi: 10.1109/TBME.2017.2664802.

[14] A. Milne, K. Farrahi, and M. A. Nicolaou, “Less is more: Univariate modelling to detect early Parkinson’s disease from keystroke dynamics,” in *Discovery Science*, L. Soldatova et al., Eds., New York, NY, USA: Springer, 2018, pp. 435–446, doi: 10.1007/978-3-030-01771-2_28.

[15] T. D. Pham, “Pattern analysis of computer keystroke time series in healthy control and early-stage Parkinson’s disease subjects using fuzzy recurrence and scalable recurrence network features,” *J. Neurosci. Methods*, vol. 307, pp. 194–202, Sep. 2018, doi: 10.1016/j.jneumeth.2018.05.019.

[16] O. M. J. Hooman, J. Oldfield, and M. A. Nicolaou, “Detecting early Parkinson’s disease from keystroke dynamics using the tensor-train decomposition,” in *Proc. 27th Eur. Signal Process. Conf.*, 2019, pp. 1–5, doi: 10.23919/EUSIPCO.2019.8902562.

[17] N. Dhir et al., “Identifying robust markers of Parkinson’s disease in typing behaviour using a CNN-LSTM network,” in *Proc. 24th Conf. Comput. Natural Lang. Learn.*, 2020, pp. 578–595, doi: 10.18653/v1/2020.conll-1.47.

[18] D. Iakovakis et al., “Touchscreen typing-pattern analysis for detecting fine motor skills decline in early-stage Parkinson’s disease,” *Sci. Rep.*, vol. 8, no. 1, Dec. 2018, Art. no. 7663, doi: 10.1038/s41598-018-25999-0.

[19] D. Iakovakis et al., “Early Parkinson’s disease detection via touchscreen typing analysis using convolutional neural networks,” in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2019, pp. 3535–3538, doi: 10.1109/EMBC.2019.8857211.

[20] M. Christ, A. W. Kempa-Liehr, and M. Feindt, “Distributed and parallel time series feature extraction for industrial big data applications,” 2017. Accessed: Nov. 01, 2020. [Online]. Available: http://arxiv.org/abs/1610.07717

[21] T. Arroyo-Gallego et al., “Detecting motor impairment in early Parkinson’s disease via natural typing interaction with keyboards: Validation of the neuroQWERTY approach in an uncontrolled at-home setting,” *J. Med. Internet Res.*, vol. 20, no. 3, Mar. 2018, Art. no. e89, doi: 10.2196/jmir.9462.

[22] C. G. Goetz et al., “Movement disorder society-sponsored revision of the unified parkinson’s disease rating scale (MDS-UPDRS): Scale presentation and clinimetric testing results: MDS-UPDRS: Clinimetric assessment,” *Movement Disord.*, vol. 23, no. 15, pp. 2129–2170, Nov. 2008, doi: 10.1002/mds.22340.

[23] A. Papadopoulos et al., “Unobtrusive detection of Parkinson’s disease from multi-modal and in-the-wild sensor data using deep learning techniques,” *Sci. Rep.*, vol. 10, no. 1, Dec. 2020, Art. no. 1, doi: 10.1038/s41598-020-78418-8.

[24] R. Djaldetti, I. Ziv, and E. Melamed, “The mystery of motor asymmetry in Parkinson’s disease,” *Lancet Neurol.*, vol. 5, no. 9, pp. 796–802, Sep. 2006, doi: 10.1016/S1474-4422(06)70549-X.

[25] D. Verbaan et al., “Cognitive impairment in parkinson’s disease,” *J. Neurol., Neurosurg., Psychiatry*, vol. 78, no. 11, pp. 1182–1187, Nov. 2007, doi: 10.1136/jnnp.2006.112367.

[26] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” 2017. Accessed: Oct. 28, 2020. [Online]. Available: http://arxiv.org/abs/1703.01365