An exploratory analysis targeting diagnostic classification of AAC app usage patterns

Adham Atyabi[1,2], Beibin Li[1], Yeojin Amy Ahn[1,2], Minah Kim[1], Erin Barney[1]

[1.] Seattle Children's Innovation & Technology Lab (SCITL)
Seattle Children's Research Institute
Seattle, WA, USA

Frederick Shic[1,2]

[2.] Department of Pediatrics
University of Washington
Seattle, WA, USA

{aatyabi, fshic}@uw.edu

Abstract—

Augmentative and Alternative Communication (AAC) apps are apps that enable non-speech communicative forms. One class of AAC apps are speech-generating devices (SGDs), where icons/pictures are tapped to produce spoken words. These apps are widely used to support communication and language learning for individuals with disabilities such as autism spectrum disorder (ASD). Given that these apps are used in everyday scenarios, they can generate massive streams of data, providing a wealth of information regarding individual usage patterns and for developing usage model profiles. However, the utility and potential of these streams of data has been little explored from a data mining perspective. The objective of this study is to evaluate several feature representations of usage patterns, coupled with data mining and data modelling techniques, for identifying differences in AAC usage patterns between users with and without ASD. The study is conducted using data streams aggregated from an AAC app called FreeSpeech, specifically designed for individuals with learning disabilities and ASD. Several feature representations for modeling usage profiles based on temporal, behavioral and frequency of usage, are investigated. The potential of each usage representation is assessed using a collection of well-known and well-established learning methods such as support vector machine and ensemble learning. While, in general, prediction performance was only slightly above chance in most representations, results from unsupervised class labeling experiments showed promising results regarding the potential of stationary keypress usage representations with bootstrapped ensembles for separating ASD from non-ASD users.

Keywords—Bagging, boosting, classification, ensembles, imbalanced data-sets, Augmentative and Alternative Communication, Autism Spectrum Disorder

I. Introduction

Augmentative and alternative communication (AAC) systems provide an alternative to speech for communication, allowingindividuals without verbal abilities toexpress thoughts, needs, and share ideas; they can also be used to foster the development of communication skills [3]. Rapid technological development has fueled growth of the AAC field [5], demonstrated by the wide use of speech-generating devices (SGDs), electronic communication boards, and AAC applications for mobile technologies.

These highly technological communication systems have multiple benefits: They offer support for exchanging information, developing social relationships, and communicating needs [6], which are all critical aspects of effective social living. Unfortunately, these activities are hindered all too often in individuals with complex communication needs, including, but not limited to, those with motor speech disorders [13], with aphasia, and with autism spectrum disorder (ASD) [14]. High-tech, often portable, AAC systems can serve as powerful tools for individuals who are minimally verbal, helping them to meet their daily needs and participate fully in interactions with others. The potential benefits that AAC systems offer are also observed in interventions that implement AAC for individuals with complex communication needs and other developmental disabilities [14,15,16,17]. For instance, AAC systems have demonstrated a great potential for fostering improvements in communication in children with ASD [7, 8, 18]. With technological advancements in portable electronic devices, AAC mobile apps further increase accessibility and availability to a more diverse population [11].

Many have speculated about the effect of technology on AAC, especially within the autism community [12,11,10,9], but there have been relatively few empirical studies. A review by Alzrayer, Banda, & Koul [6] found support for the effectiveness of an SGD app called Proloquo2Go, and iOS-based SGDs in general. Other studies have compared physical AAC systems like the Picture Exchange Communication System (PECS) with a modern, tablet-based adaptation [19].

The current study investigates data obtained from an in-house made mobile application for AAC, called FreeSpeech. FreeSpeech is a commercially available AAC app with a flexible, user-programmable interface. FreeSpeech was developed to provide its users with more opportunities to communicate directly with others, thereby increasing quality of life. It is composed of a static and dynamic screen with a field of several icons which represent specific words. When an icon is pressed, the word is spoken aloud by the device. This type of speech generating device is particularly useful for individuals who are minimally verbal as it provides a means to communicate through spoken language. To increase user-friendliness of the application, the users could reconfigure the app interface to accommodate their needs, skills, and preferences. They had options to create their own icons by uploading photos, images, words, and sentences.

The present study evaluates various feature representation mechanisms to predict usage patterns of the FreeSpeech application between those with and without ASD. Released in January 2012 and terminated in May 2014, FreeSpeech collected data from 6033 users. Understanding the usage patterns and predicting the types of individuals who appear to benefit from using this app may provide insight into the improvement of AAC apps, allowing developers to better fit their technology to specific users.

II. Problem Statement

Although substantial effort and attention is diverted towards development of AAC applications to assist individuals with ASD, these apps are not designed with usage analytics in mind. Instead, the development of such applications mostly focuses on providing an accessible mechanism for individuals to have daily and routine trainings in order to improve their performances. Analysis of users' performance is usually done as a self-assessment or is conducted by caregivers and medical professionals using psychological/physical/clinical measures. That is, the underlying design principles of these applications are not to collect valuable usage data for posthoc analysis. In fact, to the best of our knowledge, there is no existing study that utilizes patterns of AAC application usages to separate ASD from non-ASD individuals.

As a result no clear guidelines exist for dealing with recorded application interactions (if any are recorded). Given that the data structure and the principles behind the design of these apps did not consider collection of a meaningful usage pattern that could be used for training some learning models, making sense out of such ill-designed data structures is a complex and nontrivial task. This study is focuses on exploring the existing data mining and machine learning data analysis procedures, identifying suitable mechanisms for data representation of such usage recordings, and assessing the existing potential of such procedures for prediction of users' medical diagnoses.

III. Methodology

A. FreeSpeech

FreeSpeech was an AAC iPad application employed to foster communication skills and help people with speech and language disorders to communicate with others. The non-verbal communication was provided through the selection of a sequence of words/icons/buttons that shape meaningful phrases. Figure 1 depicts a screenshot of this app in action.

The application was released in January 2012 and terminated in May 2014, during which time 6033 individuals used the application. We assessed the data collected by the app between January 2012 and January 2014, which included 5372 individuals' data. Given that the app's usage information was originally collected for developer's debugging purposes (see figure 2 for a snapshot of the collected data stream) and the application was not meant to be employed as a diagnosis or treatment tool, little attention been given to the data structure and representation during the app development. This resulted in collection of less suitable data for data analysis purposes.

Fig 1. An illustrative screenshot of FreeSpeech app

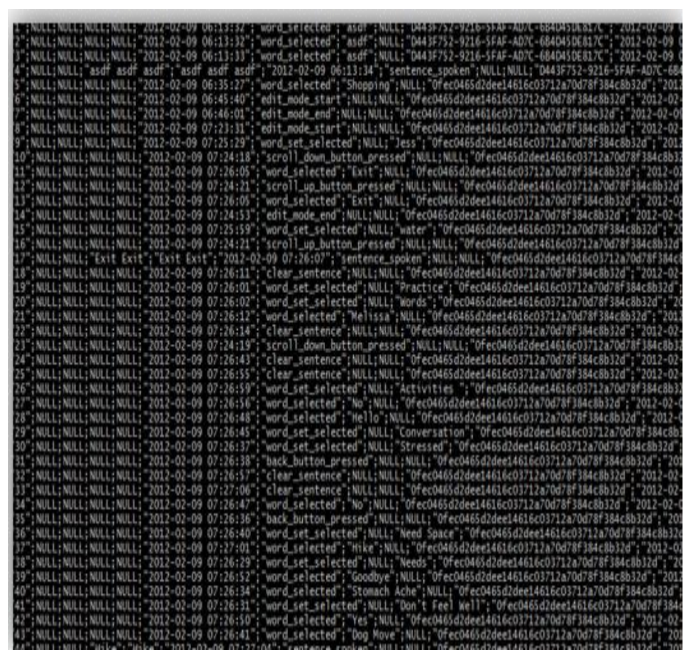

Fig 2. An example of the data stream collected by the FreeSpeech app.

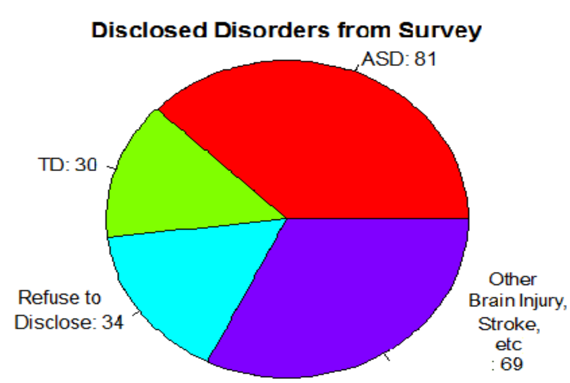

Fig 3. The distribution of FreeSpeech app users based on their self-disclosed diagnosis

The application recorded all key presses, on-screen events, and backend events in a run log. When the iPad was connected to internet, these run logs were automatically uploaded to a server. 214 users filled out a survey for the app, and 180 of them disclosed their disorders, as shown in the Figure 3. Among the 180 users, we labeled the 81 users diagnosed with ASD as the ASD group, and labelled the other 99 users as the non-ASD group for classification.

TABLE I. T-TEST SIGNIFICANT ANALYSIS OF FREESPEECH APP USAGE DATA

	Mean in Group		P-Value
	ASD	**nonASD**	
Length of Each Use (min)	2.14	2.10	0.86
Number of Word Selects per Use	5.6	4.31	0.083
Frequency of Word Selects (#/min)	2.99	2.28	0.049

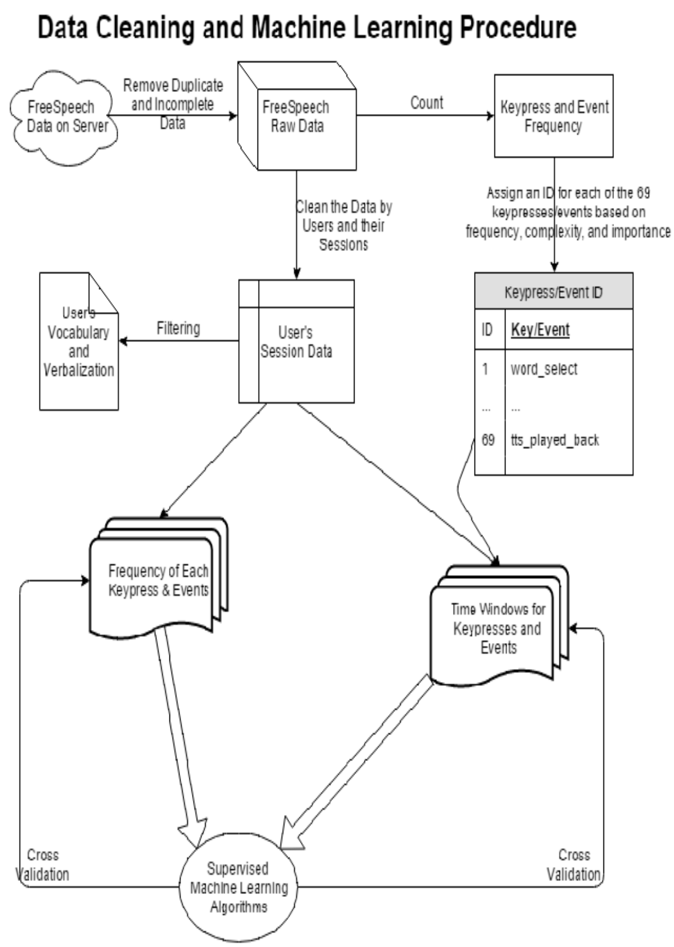

Fig 4. Data cleaning and preprocessing flowchar

To better understand the associated complexities in analyzing the data collected with the app, t-test statistical analysis is performed. The results in Table I reports lack of strong statistical significant across ASD and nonASD groups in the data collected by the app.

B. Preprocessing and usage data manipulation/preparation procedures

Three mainstream methods of feature representations are considered in this study. These methods represent time intervals of key presses, frequency of pressing a given key, and pattern of key presses. The first category reflects the amount of time (in milliseconds) took for a participant to press a new key ($t=t_1-t_0$). The second category indicates how often a particular key was pressed during a session. This representation is extended to categorical key presses as well. That is, first all existing keys/events are distributed to certain prefixed categories and later, for each participant and each session, the frequency of a key from any given category being pressed was extracted to represent category/group frequency of key press for that participant on that particular session.

Overall, 69 events/keys were identified by the app and these events/keys were grouped based on their representative complexity and event type to categories/groups of normal keys, complex keys, normal app events, and complex app events. In this study, a key press captures actions like "selecting a word," "speaking a sentence," "editing mode," etc., while app activity events include actions such as "app becoming active,", "entering background," "memory warning," etc.

The last mainstream feature representation considered in this study extracts patterns of usage from the raw data by identifying sequences of keys that were pressed during each session by each participant. In order to unify this representation, first all existing keys/events were distributed to certain prefixed categories and later, each key in a category was given a unique ID adjacent to other existing keys within that category. These IDs were utilized to capture the usage pattern of the participants. This procedure guaranteed that each key was represented with a unique value and provided the possibility of capturing categorical changes in key presses within the usage pattern.

No spatial information was considered in this analysis since the participants could customize their app interface and shuffle the key locations based on their preferences.

One session of FreeSpeech use was defined by the activities occurring between events where the app became active and then inactive. A session could last from several hours long to few seconds and could contain any number of keypresses. To address the dynamic nature of the recorded sessions, non-overlapping sliding window technique is utilized to segment the session data to manageable, fixed-size samples. This issue is further described in the following sections. The overall procedures employed or data cleaning are presented in figure 4.

To clarify the preprocessing procedure and provide a better realisation of the usage pattern representations considered in this study, the following example is provided:

Assume a sample sequence of keypresses from the raw data (an example of one key press record in the logfile is below). The feature representations in this study convert the raw keypress information to following formats:

- *A single keypress from the logfile:*
"7623";NULL;NULL;NULL;NULL;"2012-02-09_13:36:13";"word_selected";"coffee";NULL;"xxxxxx_userid_xxxxxx";"2012-02-09 13:32:58";"2012-02-09 13:32:58";NULL;NULL;NULL
- *Representation based on keypress/event time intervals with a window size of 10 keypresses:*

User ID	Session	ΔTime 1	ΔTime 2	...	ΔTime 9	ΔTime 10
00001	Date - Time	1	3	...	7	2

- *Representation based on keypress/event frequency:*

User ID	Session	Length of Session	Key 1	...	Key 69
00007	Date - Time	1250 seconds	3	...	2

- *Representation based on groupwise keypress/event frequency:*

User ID	Session	Length of Session	Key/event group 1	...	Key/event group 4
00007	Date - Time	1250 seconds	20	...	55

- *Representation based on keypress/event with a window size of 10 keypresses:*

User ID	Session	Key 1	Key 2	...	Key 9	Key 10
00001	Date - Time	23	16	...	34	52

IV. Experiment Design & Results

A. *Experiment 1. Analysis of usage patterns via various feature representations*

To investigate the potential of gathered usage data, three different feature representations were considered for in-depth analysis. These representations include keypress time intervals, keypress frequency, and keypress patterns. only app usage information of participants with known diagnosis information (labeled samples) were considered in following experiments (experiments 1.1,1.2, 1.3, & 2) in order to eliminate the effect of noisy class labeling that might be caused through using the samples that were originally unlabeled. Experiments 1 and 2 only use data from participants who disclosed their disabilities (180 individuals). However, a broader analysis is conducted in experiment 3 with keypress-based feature representation using the recorded data from all 5372 participants.

Experiment 1.1 keypress temporal features

The temporal usage representation consisted of time differences between subsequent keypresses. Given the differences in the number of keypresses across participants and their multiple sessions using the app, the resultant non-stationary representation is fixated using subsamples of non-overlapping sliding windows of size k keypresses. Each window contains temporal information of keypresses indicating the time differences between two subsequent keypresses. Several window sizes are considered in order to identify the least distorting stationary representation of the underlying pattern (k= 10, 20, 30, 40, 50, and 100 keypresses). Given the imbalanced nature of the dataset caused by differences in the number of sessions each participant used the app, the length of each session, and the number of keys they pressed in each session, an especially designed Leave-One-Out (LOO) scheme is employed in this study. First, all the data recorded from a given participant is separated as test set and the remaining data is considered as training set. The remaining data is passed through 100 bootstraps in which all samples from the less-populated group (ASD or non-ASD) are reserved while an equal number of samples from the other category are drawn randomly from the pool of training samples. This procedure is used to provide pseudo-balancing between the two groups of ASD and non-ASD samples/participants. Two layers of majority voting are used to aggregate the results. That is, first, the decision on multiple subsamples of the same subject are unified (samples each representing a different 20 keypress window of the same session/participant), and later, the decision across the 100 bootstraps are consolidated. Support Vector Machine (SVM) with radial based kernel function (RSVM) is utilized for evaluation. Radial based kernel and SVM are chosen due to their better learning generalization capability in adhoc experiments with the data. The results of this analysis are presented in Figure 5.

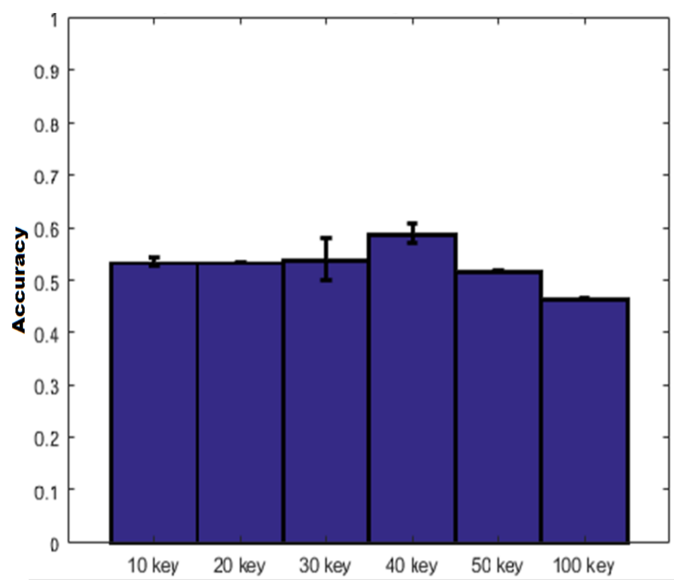

Fig 5. The averaged LOO classification accuracy of keypress time intervals with varying sizes of non-overlapping sliding windows

The results in figure 5 indicate windows sizes of 40 and 100 keypresses as the best and worst window-sizes for representing usage patterns based on temporal information. This is with the understanding that the differences in the overall classification accuracy across various window-sizes are in the order of 5-10%. The lower classification performances achieved by larger window sizes can be due to subsequent lower numbers of training samples, which can result in poor training of SVM. The overall classification performance reported in figure 5 suggests infeasibility of the time-based representation of participants' usage patterns given that the results are only slightly above 50% (chance level). This is noteworthy given that a two-sample t-test analysis of the keypress time interval data across ASD and non-ASD groups revealed lack of any statistical significance with $p > 0.05$ between the ASD and non-ASD groups with any of the window sizes utilized in this experiment.

Experiment 1.2 keypress frequency features

In this experiment, frequency of each key (or key-group) being pressed in a session by a given participant is used to represent the app usage. Like experiment 1.1, LOO with nested 100 bootstrap is used for balancing the training sets between ASD and nonASD groups. Average (over 5 repetitions) classification accuracy of RSVM in the LOO scheme nested by 100 bootstraps is employed as the assessment criteria. The results indicated lack of performance differences across the two types of frequency-based representations utilized in this experiment (e.g., frequency of a given key being pressed versus frequency of a given key-group being pressed). The average classification accuracy is barely above chance e.g., 53.8%.

Experiment 1.3 keypress features

This experiment is designed to investigate various keypress fixed window size usage pattern representations. First, keypress features captured within non-overlapping window sizes of 10, 20, 30, 40, 50, and 100 are considered. This experiment is inspired from previous feature representation studies in which similar windowing and overlapping mechanisms are considered and shown that such methodologies are able to improve pattern discrimination performances with classifiers such as SVM [22-25]. Like previous experiments the combinations of RSVM and 5 repetitions of an especially designed LOO scheme that is nested with 100 bootstraps are employed for evaluation. The results are presented in Figure 6. To better understand the feasibility of this feature representation and to reduce the chances of poor classification performance being achieved due to lack of having adequate number of training samples, 25%, 50%, 75% and 90% overlapping windows are also evaluated (only on window size of 20 keypresses).

The results in figure 6, representing non-overlapping samples with varying window sizes, are similar to what is reported with temporal usage representation in experiment 1.1, with the best overall classification performance being achieved by a window-size of 40 keypresses on non-overlapping keypress sequences. Like experiment 1.1, a window size of 100 keypress sequences achieved the poorest classification performance.

The subsequent multi-scaled keypress representation using a range of overlapping percentages showed a slight overall prediction improvement.

Two-sample t-test analysis of the keypress pattern data across ASD and non-ASD groups revealed statistically significant differences across the usage patterns of these two groups of participants with $p < 0.05$ across all window sizes and all overlapping percentages (0%, 25%, 50%, 75%, 90%). The presence of statistical significance across the ASD and non-ASD groups with keypress features while the classification accuracy is only above the chance performance suggests that the overall keypress patterns share enough similarities across the two groups to make them difficult for the learners to distinguish between the two groups appropriately. The reported performance with overlapping windows indicates that having more training samples with keypress usage patterns is not the most effective factor for separating diagnostic categories.

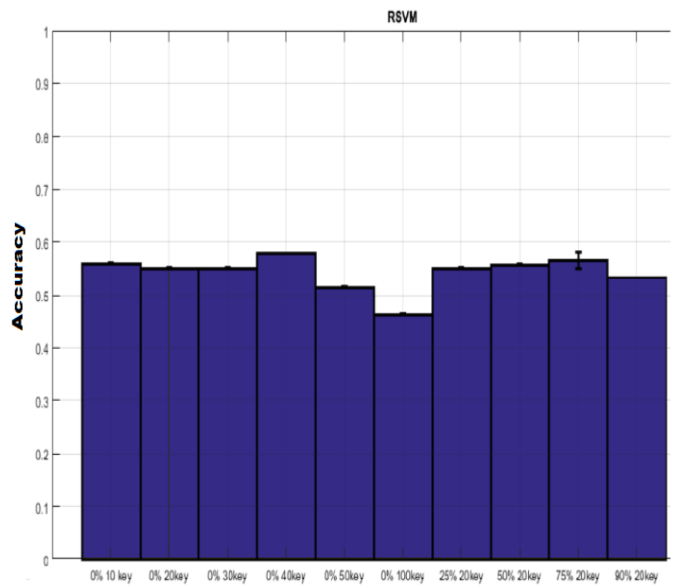

Fig 6. The averaged LOO classification accuracy of keypress features with varying sizes of sliding windows with overlapping ratios of 0% (non-overlapping) to 90%

B. *Experiment 2. Sample multiplication and robust learners' impact on overall diagnosis prediction*

To better understand the underlying factors in prediction of ASD and non-ASD classes from keypress usage patterns, Experiments 1.2 and 1.3 are replicated using more complex and stronger learners including variations of boosting and bagging ensembles (AdaboostM1, RobustBoost, LPBoost, TotalBoost, RusBoost, SubSpace, LogitBoost, GentleBoost and Bagging). In this collection of boosting and bagging ensembles, RobustBoost is considered to investigate possible effects that noisy class labels might have on overall prediction accuracy since the class labeling information is taken from participant self-reports that could be unreliable. LPBoost, Bagging and TotalBoost are considered due to their robustness with dealing with small training samples. The remaining boosting ensembles are utilized based on the potential they showed in previous studies [4]. A review of these methods and their detailed descriptions can be found in [20,21].

The results of classification performance using keypress frequency features are illustrated in Figure 7. The results indicate a clear classification improvement with the best performance being achieved by LogitBoost and keypress frequency. Keypress frequency performs slightly better compared with key group frequency. LogitBoost performance with keypress frequency is closely followed by Bagging and GentleBoost methods. It is noteworthy that except for Rusboost, all other methods report above-chance prediction despite the ill-defined nature of our feature space in keypress frequency category.

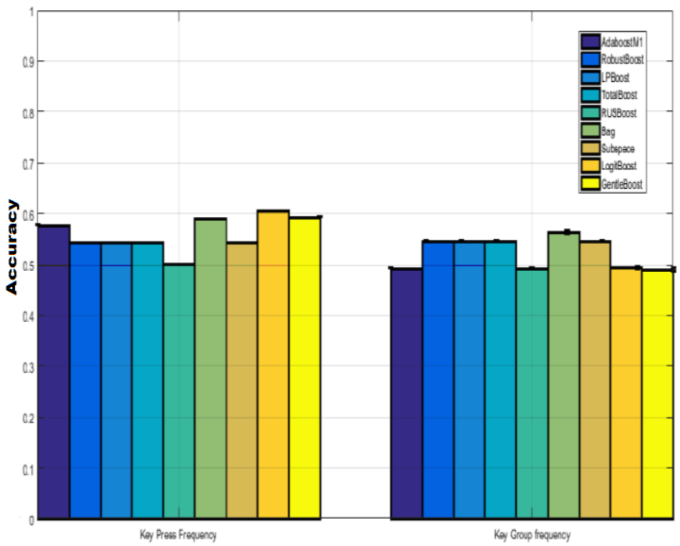

Fig 7. The averaged LOO classification accuracy of keypress frequency feature representation with a collection of strong and well-known ensemble learners

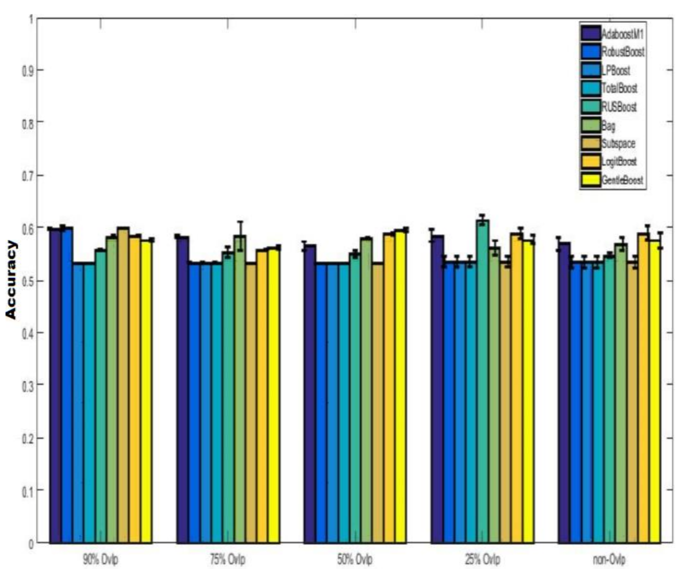

Fig 8. The averaged LOO classification accuracy of keypress feature representation using sliding window size of 20 keypresses with various overlapping rations ranging from 0% to 90%. A collection of strong and well-known ensemble learners are used for assessment.

The averaged LOO classification accuracy of keypress feature representation using a sliding window size of 20 keypresses with various overlapping ratios ranging from 0% to 90%. A collection of strong and well-known ensemble learners are used for assessment.

The results in Figure 8 highlight the impact of using strong learning in overall classification accuracy with the best overall performance being achieved by RusBoost and 25% overlapping keypress windows of size 20 keys.

C. *Experiment 3 Autism diagnosis from non-stationary app usage patterns*

This experiment is aimed at evaluating the predictive value of main feature representation of usage patterns. As mentioned earlier, a substantial number of the aggregated usage patterns lack diagnostic classes (only 180 participants out of 5372 disclosed their conditions) which makes this evaluation more complicated. To address this issue, the following mechanism is considered to predict the diagnostic classes from users' keypresses.

First, key press informations is cataloged and each key is given a unique value based on the overall category to which it belongs (see preprocessing section). Given the differences in the number of keys pressed by each user in each session, a non-overlapping sliding window of 20 keypresses is utilized and all shorter window data entries are ignored.

Self Organizing Map (SOM) and K-means clustering algorithms are utilized to relabel the samples. The class relabeling process is accomplished using:

- the 1st 20 keypresses of the first session of each participant resulting in a dataset with a single sample (20 keypress) per participant
- all 20 keypress windows from all sessions resulting in a dataset with more than one sample per participant (unless the participant only used the app once and only pressed 20 to 39 keys).

The results are presented in Table II. In this procedure, the outcome of the clustering methods are assessed based on the percentage of samples from known diagnosis group (e.g., usage data of 180 participants) of ASD and non-ASD groups that disagree with the cluster-labeling decisions of the majority. The results (averaged across 100 and 10 repetitions in Kmeans and SOM respectively) indicate a better overall performance with lower error rates when all of the sliding windows of 20 keypresses are used for the assessment. The high percentage of error rates reported by the non-ASD group can be attributed to the fact that this group includes a collection of individuals with varying learning/language disabilities in addition to typically developing users.

TABLE II. THE RESULTS OF DIAGNOSIS PREDICTION ON SAMPLES WITH KNOWN DIAGNOSIS USING CLUSTERING AND REGRESSION METHODS

	ASD Error Rate	**non-ASD Error Rate**
Class label prediction using all 20 keypress samples of each participant		
SOM	21.8129%	53.3854%
K-Means	21.8245%	53.3854%
Class label prediction using only the 1st 20 keypresses of the first session in each participant		
SOM	39.1304%	58.9744%
K-Means	39.1304%	58.9744%

he overall suitability of the best relabeled set (labels identified by k-means clustering method using all subsamples of 20 keypresses) is evaluated using an specially designed Leave-One-Out (LOO) scheme that contains 100 bootstraps of the training set for pseudo-label-balancing. This procedure is eliminates any effect that might be caused by the imbalanced nature of our dataset. The nested-LOO scheme is designed in a way to eliminate any chance of subsequent samples of the same participant being presented in both training and testing simultaneously. It should be noted that although the current dataset is imbalanced in principle, the added bootstrap component in the LOO scheme generated a set of pseudo-balanced training sets in which all of the samples from the smallest class population are reserved and an equivalent number of samples from the other class are randomly chosen from the existing pool. The 100 repetitions of the bootstrap is used to eliminate the chance of randomly choosing a well-tuned or badly-tuned training set which can result in exceptionally high or poor classification performances. It is noteworthy that the utilized LOO scheme contains two layers of majority voting in which, first a majority voting is applied over the predictions of multiple sliding windows originating from the same participant's app usage information, and later a secondary majority voting is conducted to unify the decision across the 100 bootstraps that each has a different training sample combination.

The classification performance achieved using the logistic regression method showed 70% accuracy in medical diagnosis prediction. This is noteworthy since evaluations with pre-known diagnosis samples (app usage information from 180 individuals out of 5372 participants) under similar evaluation schemes with much stronger learners (SVM and variations of bagging and boosting) reported near chance levels of classification accuracy. While this result may be due in part to smoothing out of diagnostic variability due to our label assignment technique, it also highlights the potential feasibility of using patterns of key presses to predict diagnosis when enough training samples are utilized even with weak learners such as logistic regression.

V. CONCLUSION

This study was focused on the implications of various feature representation approaches for capturing the differences in usage patterns of a group of typically developing individuals and others who suffer from autism spectrum disorder, language impairment, and learning disability. Using several machine learning and data mining methods, the aim was to predict the self-disclosed diagnosis (ASD vs non-ASD) through an individual's keypress patterns in their use of an AAC app. The study took advantage of the recorded data from an AAC app called FreeSpeech and investigated three mainstreams of feature representation based on the patterns of keypresses. These feature representations captured aspects of app usage such as i) the time intervals between consecutive keypresses, ii) frequency that a key or the group that the key belonged to was pressed, and iii) which combinations of keys were pressed. Given the differences in the usage patterns across multiple sessions and participants, sliding window representation was considered in the study. The study investigated various window sizes with varying degrees of overlapping percentages and the results suggested that using keypress patterns represented either as frequency or sliding windows of 20 keypresses with 25% overlapping used in combination with strong learning methods such as RusBoost is able to provide adequate predictions.

To better understand the limits of the feature representations discussed in this study, additional recorded data without a self-disclosed diagnosis is utilized. First, using k-means clustering approach, the unlabeled dataset is labeled to two groups of ASD and non-ASD and later, using combinations of i) leave one out nested with 100 bootstraps and two layers of majority voting and ii) a logistic regression algorithm, a diagnosis is predicted based on the keypress patterns. The achieved 70% classification accuracy indicated the potential of such a feature representation mechanism even with weak learning algorithms such as logistic regression in addition to highlighting the importance of utilizing an adequate number of training samples especially when such ill-captured and ill-designed app usage data is being used.

Similar data representation and data mining methodologies could be used to characterize other complex interactions using technology, e.g. interactions with robots [26]. These methods can also be utilized for characterizing other HCI applications such as those with video games [27] and to characterize the usage of other apps for children with autism operating at various time scales, from apps designed to teach skills to special populations in intensive sessions [28,29] or with those used to promote positive habits of behavior [30].

ACKNOWLEDGMENT

This study received funding and support from NIH awards K01 MH104739, R21 MH103550, the NSF Expedition in Socially Assistive Robotics #1139078, IES EDIES13C0046 I+II, the FAR Fund, and the Associates of the Child Study Center.

References

[1] R.W. Schlosser, and R. K. Koul, “Speech output technologies in interventions for individuals with autism spectrum disorders: a scoping review”. Augmentative and Alternative Communication, vol 0 no. 0, pp. 1–25. doi: 10.3109/07434618.2015.1063689. 2015.

[2] A. Alliano, K. Herriger, A.D., Koutsoftas, and T.E. Bartolotta, “A review of 21 iPad applications for augmentative and alternative communication purposes”, Perspectives on Augmentative and Alternative Communication, vol. 21, no. 2, pp. 60. doi: 10.1044/aac21.2.60, 2012.

[3] P. McNairn, and C. Shioleno. “Augmentative communication—Part 1: Can we talk? Parents’ perspectives on augmentative and alternative communication”. The Exceptional Parent, vol. 30, no. 2, pp. 72–73, 2000.

[4] S. Valencia, M.V. Smith, A. Atyabi, and Frederick Shic, “Mobile ascertainment of smoking status through breath: a machine learning approach”, IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, In press, 2016.

[5] D. McNaughton, and J. Light, “The iPad and mobile technology revolution: Benefits and challenges for individuals who require augmentative and alternative communication”, Augmentative and Alternative Communication, vol. 29, no. 2, pp. 107-116, 2013.

[6] N. Alzrayer, D.R. Banda, and R.K. Koul, “Use of iPad/iPods with individuals with autism and other developmental disabilities: A meta-analysis of communication interventions”, Review Journal of Autism and Developmental Disorders, vol. 1, no.3, pp. 179-191, 2014.

[7] J.B. Ganz, T.L. Earles-Vollrath, R.A. Mason, M.J. Rispoli, A.K. Heath, and R.I. Parker, “An aggregate study of single-case research involving aided AAC: Participant characteristics of individuals with autism spectrum disorders”. Research in Autism Spectrum Disorders, vol. 5, no. 4, pp. 1500-1509, 2011.

[8] S. Sennott, and A. Bowker, “Autism, aac, and proloquo2go”, SIG 12 Perspectives on Augmentative and Alternative Communication, vol. 18, no. 4, pp. 137-145, 2009.

[9] H.C. Shane, E.H. Laubscher, R.W. Schlosser, S. Flynn, J.F. Sorce, and J. Abramson, “Applying technology to visually support language and communication in individuals with autism spectrum disorders”. Journal of autism and developmental disorders, vol. 42, no. 6, pp. 1228-1235, 2012.

[10] P. Mirenda, “Promising innovations in AAC for individuals with autism spectrum disorders”. SIG 12 Perspectives on Augmentative and Alternative Communication, vol. 18, no. 4, pp. 112-113, 2009.

[11] J. Light, and D. McNaughton, “Supporting the communication, language, and literacy development of children with complex communication needs: State of the science and future research priorities”, Assistive Technology, vol. 24, n. 1, pp. 34-44, 2012.

[12] J. Bradshaw, “The use of augmentative and alternative communication apps for the iPad, iPod and iPhone: an overview of recent developments”, Tizard Learning Disability Review, vol. 18, no. 1, pp. 31-37, 2013.

[13] E.K. Hanson, D.R. Beukelman, and K.M. Yorkston, “Communication support through multimodal supplementation: A scoping review”. Augmentative and Alternative Communication, vol. 29, no. 4, pp 310-321, 2013.

[14] D. Beukelman, and P. Mirenda, Augmentative and alternative communication: Supporting children and adults with complex communication needs (3rd ed.) Baltimore: Paul H. Brookes, 2005.

[15] J.B. Ganz, “AAC interventions for individuals with autism spectrum disorders: State of the science and future research directions”, Augmentative and Alternative Communication. vol. 31, no. 3, pp. 203-214, 2015.

[16] J. Stephenson, “We need to know more about providing effective support for communication partners of children who use augmentative and alternative communication”. Evidence-Based Communication Assessment and Intervention. vol. 10, no. 1, pp. 1-6, 2016.

[17] L.A. Van der Meer, and M. Rispoli, “Communication interventions involving speech-generating devices for children with autism: a review of the literature”, Developmental Neurorehabilitation, vol. 13, n. 4, pp. 294-306, 2010.

[18] J. Light, and K. Drager, “AAC technologies for young children with complex communication needs: state of the science and future research directions”, Augmentative and Alternative Communication. vol. 23, no. 3, pp. 204-216, 2007

[19] J.B. Ganz, E.R Hong, and F.D. Goodwyn, “Effectiveness of the PECS phase III app and choice between the app and traditional PECS among preschoolers with ASD”, Research in Autism Spectrum Disorders. vol. 7, no. 8, pp. 973-983.

[20] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, pp. 1-22, 2011

[21] D. Guan, W. Yuan, Y. W. Lee, M.M.K. Rasel, "A Review of Ensemble Learning Based Feature Selection", IETE Technical Review 31(3):190-198, 2014.

[22] A. Atyabi M. Luerssen, S.P. Fitzgibbon, and D.M.W. Powers, “Biasing the overlapping and non-overlapping sub-windows of EEG recording”, 2012 IEEE World Congress on Computational Intelligence (IEEE WCCI 2012), pp. 741-746, 2012.

[23] A. Atyabi, S.P. Fitzgibbon, and D.M.W. Powers, “Multiplication of EEG samples through replicating, biasing, and overlapping”, Brain Informatics BI’12, Springer Lecture Notes in Computer Science, pp. 209-219, 2012.

[24] A. Atyabi and D.M.W. Powers, “The impact of segmentation and replication on non-overlapping windows: an EEG study”. Second International Conference on Information Science and Technology, ICIST’12, pp. 668-674, 2012.

[25] A. Atyabi, S.P. Fitzgibbon, and D.M.W. Powers, “Multiplying the mileage of your dataset with subwindowing”. Brain Informatics BI’11, Springer Lecture Notes in Computer Science, Vol. 6889, pp. 173-184, 2011.

[26] B. Li *et al.*, “Human Robot Activity Classification based on Accelerometer and Gyroscope,” in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2016, pp. 423–424, 2016.

[27] M. Weng, C. Wall, E. Kim, L. Whitaker, M. Perlmutter, Q. Wang, E. Lebowitz, and F. Shic, “Linking Volitional Preferences for Emotional Information to Social Difficulties: A Game Approach using the Microsoft Kinect”. Proceedings of Affective Computing and Intelligent Interaction (ACII2015), Xian, China: ACM, 2015.

[28] E.S. Simmons, R. Paul, and F. Shic, “Brief Report: A Mobile Application to Treat Prosodic Deficits in Autism Spectrum Disorder and Other Communication Impairments: A Pilot Study”. Journal of Autism and Developmental Disorders, 1–8. doi: 10.1007/s10803-015-2573-8, 2015

[29] L.M. Rice, C.A. Wall, A. Fogel, and F. Shic, “Computer-Assisted Face Processing Instruction Improves Emotion Recognition, Mentalizing, and Social Skills in Students with ASD”. Journal of Autism and Developmental Disorders, 1–11. doi:10.1007/s10803-015-2380-2, 2015.

[30] B. S. Popple, C.A. Wall, L. Flink, K.K. Powell, D.B. Keck, M. Mademtzi, F.R. Volkmar, and F. Shic, “Brief Report: Remotely Delivered Video Modeling for Improving Oral Hygiene in Children with ASD: A Pilot Study”. Journal of Autism and Developmental Disorders. (In press), 2016