

# Motion-Print: A Biometric for Real-Time Pilot Identification using Hierarchical Temporal Memory

Sam Heiserman <sup>1</sup>, Kirill Zaychik <sup>2</sup>, and Timothy Miller <sup>2</sup>

<sup>1</sup>Watson School

<sup>2</sup>Affiliation not available

October 30, 2023

## Abstract

This study presents a novel biometric approach to identify operators, given only streams of their control movements within a manual control task setting. In the present task subjects control a simulated, remotely operated robotic arm, attempting to dock onto a satellite in orbit. The proposed methodology utilizes the Hierarchical Temporal Memory (HTM) algorithm to distinguish operators by their unique control behaviors. Results presented compare the identification performance of HTM with Dynamic Time Warping (DTW) and Edit Distance on Real Sequences (EDR), in both static and real-time data settings. The HTM method outperformed both DTW and EDR in the real-time setting, and matched DTW in the static setting. Observed superior performance of the HTM algorithm lays the foundation for the extension of the proposed methodology to other motion-monitoring applications, such as real-time workload assessment, motion/simulator sickness onset or distraction detection.

The data gathered in the study was posted to IEEE-dataport, DOI: 10.21227/wpyf-r927

# Motion-Print: A Biometric for Real-Time Pilot Identification using Hierarchical Temporal Memory

Sam Heiserman, *Member, IEEE*, Kirill Zaychik, and Tim Miller

**Abstract**—This study presents a novel biometric approach to identify operators, given only streams of their control movements within a manual control task setting. In the present task subjects control a simulated, remotely operated robotic arm, attempting to dock onto a satellite in orbit. The proposed methodology utilizes the Hierarchical Temporal Memory (HTM) algorithm to distinguish operators by their unique control behaviors. The HTM neural network is an online learning model for sequential data, inspired by incremental Hebbian learning found in the neocortex. The efficacy of the proposed method is based on the hypothesis that individual subjects show consistent and idiosyncratic control behavior patterns. The authors speculate that once trained by a particular operator’s movements an HTM network can later individuate them, by detecting more anomalies from all other operators whose movements are more foreign to the model. Results presented in this paper compare the identification performance of HTM with Dynamic Time Warping (DTW) and Edit Distance on Real Sequences (EDR), in both static and real-time data settings. These algorithms were also tested across all control channel combinations (field-sets) and a range of data granularity levels within the static setting. The HTM method outperformed both DTW and EDR in the real-time setting, and matched DTW in the static setting. Observed superior performance of the HTM algorithm lays the foundation for the extension of the proposed methodology to other motion-monitoring applications, such as real-time workload assessment, motion/simulator sickness onset or distraction detection.

## NOMENCLATURE

<i>HTM</i>	Hierarchical Temporal Memory
<i>NuPIC</i>	Numenta Platform for Intelligent Computing
<i>SP</i>	Spatial Pooler
<i>TM</i>	Temporal Memory
<i>NAS</i>	Nupic Anomaly Score
<i>MNAS</i>	Mean Nupic Anomaly Score
<i>RS</i>	Rank Score
<i>MRP</i>	Matching Rank Position

## I. INTRODUCTION

Automated user authentication has traditionally been done largely with knowledge and possession-based methods [1]. Knowledge-based systems store information like passwords, Personal Identification Number(s) (PIN) or answers to security questions for all members enrolled, granting access when the

input provided matches what’s stored. Possession-based systems grant access when presented with certain physical tokens, such as tickets or transponders [2]. Both of these systems come with a critical weakness inherent to all knowledge and physical items, they can be stolen. In these cases there may be no other mechanism to ensure that the provider of the correct password or token is not an impostor, and several major password leaks in recent years have shaken public confidence in the security of the password paradigm [3].

This critical vulnerability is not inherent to biometric methodologies however. Biometrics is a class of methods by which individuals are identified by certain of their physical or behavioral traits [1]. Biometric attributes are advantageous in that they are difficult to forge and unlike passwords and tokens cannot be lost, stolen or forgotten. These desirable traits have likely driven the adoption of biometric authentication within both the public and private sectors [4]. There are, however, a number of conditions that must be satisfied in order for any biometric system to perform reliably [2]:

- *Universality*. The trait(s) must be attainable from the vast majority of people.
- *Invariance*. The trait(s) must remain as constant as possible for each subject over time, so they can be recognized far into the future. This issue is inherent to biometrics as all human attributes change over time, with certain modalities especially sensitive such as those based on subjects’ face, voice and motor-controls [2].
- *Collect-ability*. The trait(s) must be collectible within a reasonable amount of time and with minimal invasiveness to the subjects. The collection process must not be overly limited by any sort of practical constraints.
- *Performance*. The data collected must contain sufficiently distinguishing characteristics of the subjects.
- *Acceptability*. The trait(s) must be such that subjects have little to no discomfort or worry in submitting their data.
- *Circumvention-proof*. The trait(s) must be robust to mimicry by impostors.

Each of these conditions is necessary to any viable and scalable biometric authentication system, making biometrics clearly more complex to implement and more variable in reliability than knowledge or possession-based methods. The oldest and still most widely adopted biometric modality is fingerprint analysis [5], though the literature now shows a range of physical and behavioral biometric modalities in use [6].

Physical modalities include fingerprint, palmprint, vein pattern, ear shape, iris, retina and face images, all of which

S. Heiserman is with the Department of Systems Science and Industrial Engineering, Binghamton University, Vestal, NY, 13902 USA e-mail: (sheiser1@binghamton.edu).

K. Zaychik is with the Department of Mechanical Engineering, Binghamton University, Vestal, NY, 13902 USA e-mail: (kzaychik@binghamton.edu).

T. Miller was with the Department of Computer Science, Binghamton University, Vestal, NY, 13902 USA e-mail: (millerti@binghamton.edu).

Manuscript received April 15, 2020; revised May 26, 2020.

can be statically captured in a single snapshot. Behavioral modalities rely on sequences of data generated as subjects interact with their devices, they include keystrokes, voice, gait and signatures, all of which must be dynamically captured over time [7].

Dynamic behavioral biometrics offer several inherent advantages over their static counterparts. For one they can run passively in the background, monitoring subjects activity without needing their attention at all. For another they can potentially detect when a subject is acting out of the ordinary. For instance if a would-be driver or pilot going through authentication is intoxicated, static metrics like face image or fingerprint are less equipped to detect it than dynamic modalities that consider subjects control behaviors over time. Despite these strengths behavioral modalities are less common in stand-alone use than their static counterparts since they have not achieved the same broad performance reliability [1].

Reviews of leading behavioral biometrics, which tend to focused on gait, voice and keystroke dynamics point out several main limitations: First that the error rates reported have not met the European standard for commercial biometric technology, second that the authentication times have either been unreported or impractically long, and third that direct comparisons cannot be made between most studies due to the lack of common experimental methods and data sets [8]. In the present study we attempt to address these limitations, with a novel behavioral modality introduced, a novel data set gathered and a new HTM-based modeling approach applied using the open source NuPIC software [9].

The following subsection of the manuscript summarizes some of the most recent research efforts in the domain of biometric identification of individuals.

## II. RELATED WORK

Zaychik [10] proposed a methodology to characterize the control behaviors of given individuals by identifying person-specific parameter sets of the traditional model of the human operator, such as the Hess structural model [11], Hosman descriptive model [12] and the algorithmic Optimal Control Model [13]. Parameter identification is done via a genetic algorithm based technique called APID (Automatic Parameter Identification), which relies on matching the power spectrum density of the actual and simulated control behavior signals. The author was able to demonstrate some level of distinction between various operators, however the results were limited to one degree of freedom manual control compensatory task chosen for that study. That choice was forced by the fact that conventional models of the human operator were designed to work well a single degree of freedom control task. This is a serious obstacle, which limits the horizons for potential applications. The HTM-based approach is not limited in this way however, due to its capacity for multivariate inputs. The authors suspect this capacity to be crucial for performance in HTM and any multivariate method, given the 4-D structure of the data.

Though the behavioral biometric modalities have generally not proven as reliable as fingerprints and other physical modalities, several recent publications show the strong identifying

capacity of a new modality called Event-Related Potentials (ERP), calculated from electroencephalogram (EEG) sensor data. Laszlo [14] et al, identified individual subjects using their EEG ERP's generated in response to given streams of text. The authors achieved 82-97 percent classification accuracy with a pool of 45 subjects despite using the minimum possible three electrode sensors. In their subsequent work [15] the authors from the same lab proposed a protocol for ERP-based biometrics drawing from multiple brain systems, showing subjects pictures of phrases, faces and foods to elicit unique individual responses. This time they achieved 100 percent classification accuracy from a pool of 50 subjects. Finally, the same group of authors performed a follow-on work testing the system's reliability over time by rerunning the experiment with the same subjects and stimuli over a year later and still achieving 100 percent accuracy [16].

However reliable the ERP modality may be, it is still limited in its collect-ability due to the time and complications that come with conducting an EEG experiment. This type of biometric authentication may be considered invasive for the following reason: to obtain ERP data at least three electrode sensors must be connected to the subjects' heads in order for their neural responses to be recorded [15]. This modality is thereby limited in its collect-ability. The most practical and scalable biometric modalities are those with low cost, non-invasive data collection protocols. One such modality is subject signatures, which are both uniquely identifying and simple to record. ERP data can be thought of as a neural form of behavioral signature, though it is not simple to record.

One other form of control behavior which is both universal and simple to record is mouse movements (from traditional PC control devices), the focus of a subfield known as Mouse Dynamics Authentication (MDA). With no invasive sensors or other specialized hardware needed MDA offers the advantages of low cost and low intrusion data collection, making it a viable modality within the modern Cloud Computing paradigm. The mouse data is generated continuously through the sessions, enabling its use for continuous authentication in real-time [17]. These strong attributes have likely driven the increase in MDA research interest over the past ten years [18]. The MDA modality is constrained, however, by the common limitations to behavioral modalities mentioned earlier:

- reported error rates have not met the European standard for commercial biometric technology
- authentication times have either been unreported or impractically long
- direct comparisons can't be made between most studies due to the lack of common experimental methods and data sets [17].

In spite of these drawbacks each survey of the MDA field concludes that mouse dynamics as a behavioral modality has shown clear value in distinguishing between human subjects, and strong promise for future research [19].

## III. METHODOLOGY

In order to test identification performance of the proposed HTM-based system and compared distance measures a set

of experiments were devised, to collect the data generated by human operators on a particular manual control task. There were 38 individuals who participated in this study. Each subject conducted 15 trial runs during which data were collected. HTM models were trained for each subject using 12 of their 15 runs, and then tested for their similarity to the other 3. Cross-validation was applied by switching which 3 runs are tested on, generating new sets of profiles and Rank Scores for each fold. The proposed methodology is illustrated in Figure 1.

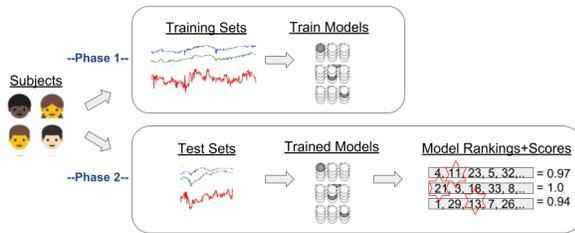


Fig. 1: Proposed framework of the Motion-Print methodology

During Phase 1, twelve (12) out of fifteen (15) recorded runs per subject were concatenated into a single training sequence. The remaining three (3) runs were concatenated into single test-sequence and held aside to be used during Phase 2 of the process.

Under the HTM method operators were profiled by models trained on their respective training sets. Under the compared distance-based methods operators were profiled by their training sets themselves, with no modeling since they calculate distance directly between raw data sets.

In order to increase the statistical reliability of the analysis, cross validation was performed, whereby each subject's 15-run data set was assembled into 5 non-overlapping folds. Each fold's training set was composed of 12 runs concatenated to one, with with test set composed of the remaining 3 runs concatenated. Then the process of model learning (in HTM), testing and ranking was repeated with each fold independently. Such data handling algorithm resulted in 38 Rank Scores per fold and 190 total.

Each subject's profile (either HTM model or raw training set) was ranked from 1 to 38 based on how similar it was to that subject's test set, relative to all other subjects' profiles. For the compared time series distance metrics this ranking was maintained. The difference was that similarity for these measures was not model-based, rather calculated directly as the distance between each train and test set. Results of the cross-validation study are presented and discussed in the Results section of this manuscript.

Finally Phase 2 was conducted in both static and real-time data settings. In the static setting the concatenated test sets were presented to the algorithms all at once, in a single batch. In the real-time setting this data was presented incrementally, one point at a time. This required the algorithms to calculate a distance at each time step, and use the average for the total train-test distance. Since the HTM method only processes data incrementally it was the same for both settings.

The following paragraphs and subsections detail the key elements of the proposed Motion-Print framework and HTM methodology.

### A. Control Task

Traditionally, a manual control task can be described as follows: a human operator controls a dynamic system (the 'plant') by adjusting his/her control behavior in response to some feedback information received from the plant. The objective for control is set by some reference signal. The manual control task is illustrated in Figure 2.

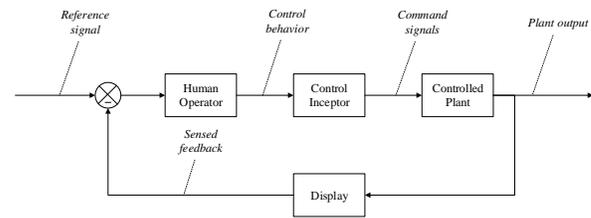


Fig. 2: Closed loop manual control diagram

In this particular study the manual control task involves maneuvering a remotely operated robotic arm to dock with a satellite orbiting the planet. Human operators exercise control over the robotic arm via a joystick controller. Plant position (robotic arm end effector) with respect to the target (satellite) is fed back to the human operator via regular LCD display.

### B. Hardware and software

The control inceptor used in this study was the Logitech Force 3D Pro joystick controller [20] as shown in Figure 3



Fig. 3: Joystick Control Device

The joystick offers a simple and intuitive user interface to perform multi-channel control of the simulated plant. Relationships between joystick movements and controlled parameters are summarized in Table I

Joystick Input	Plant Response
roll	x-coordinate
pitch	y-coordinate
trigger	on/off translation along z-axis
throttle	speed, rate of translation along all axes

TABLE I: Control Inceptor Commands

The controlled plant (robotic arm) has simple  $\frac{1}{s}$  dynamics in all three translational degrees of freedom.

Data collected at each run includes pitch and roll stick deflections, trigger setting and throttle position, z-coordinate of the plant and distance-to-target. Cartesian distance between plant and target (“distance” hereafter) is a computed parameter, according to the formula  $\sqrt{x^2 + y^2 + z^2}$ .

The display used in the experiments was a generic 24 inch LCD panel. The simulated plant position relative to the target as seen on the display is shown in Figure 4

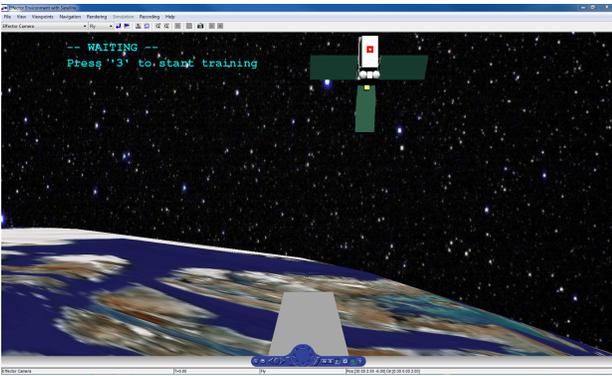


Fig. 4: Screenshot of the simulation as seen by the operator

The entire simulation of the control task was implemented in Matlab by a group of students under the NASA/New York Space Grant Consortium National Space College and Fellowship Program, sponsored by NASA Goddard Space Flight Center in 2016.

### C. Experimental Setup

Experiments were conducted at the Man-Machine Systems Laboratory at the State University of New York at Binghamton. The pool of 38 subjects was primarily composed of engineering graduate students and faculty from the Watson School of Engineering and Applied Science. Participants ranged in age from 23 to 42 years old, with a gender distribution of 84% male vs 16% female individuals. No other participant meta-data were collected. Each subject completed several practice runs in order to familiarize with the controls and task objectives. Task performance was evaluated for each subject by time used to complete docking with the satellite. Once the training asymptote was achieved, each subject proceeded with 15 runs. Experimental data collected during each run included the control inceptor commands as well as the simulated plant output parameters such as spatial coordinates of the robotic arm end effector.

Data were collected at a sampling rate of 100 Hz. Since the natural frequency of manual control behavior hardly exceeds

3 Hz, identification performance was suspected to improve at lower sampling rates. The data were thus preprocessed for the static data setting, by aggregating the signals over every 1, 2, 3, 5, 10, 20, 30, 50, 100 and 200 time steps, which correspond to 100, 50, 33.33, 20, 10, 5, 3.03, 2, 1 and 0.5 Hz granularity levels respectively. This approach, equivalent to Piece-wise aggregate approximation (PAA), served to smooth the time series and reduce the amount of data to be ingested by the proposed HTM method and compared measures [21]. Other preprocessing methods were also implemented, including moving average smoothing and differencing, although the results were excluded from consideration since they were not nearly as good.

### D. Compared Measures

The following algorithms were used to perform identification, measuring the familiarity of each unknown test sequence to each subject profile. The EDR and DTW distance measures compared to HTM are noted as a leading time series distance metric in the literature [30]:

- Hierarchical Temporal Memory (HTM)
- Dynamic Time Warping (DTW)
- Edit Distance on Real Sequences (EDR)

**Hierarchical Temporal Memory (HTM)** is a neural network which forms predictive models incrementally from data streams. This means that it makes predictions and updates its prediction model with each new input. It thereby does not need a fixed time series to learn from or calculate anomaly score distance from. This makes it more fit for live streaming scenarios than methods which operate all at once on fixed series. Unlike in leading Artificial Neural Network models the internal mechanics of HTM are derived entirely from neuroscience, specifically how the neocortex learns time-based patterns [24].

**Edit Distance on Real Sequences** is a measure of distance between two vectors based on the minimum number of edit operations (insertions, deletions and substitutions) called for to transform one sequence into the other. It is specifically made to measure similarity between trajectories. In the setting of numerical data the distance between points in the time series is reduced to 0 or 1 [27].

**Dynamic Time Warping** is a measure of distance between two vectors of equal or different sizes based on finding the optimal alignment between them. It does so by searching for the minimal path in a distance matrix that defines a mapping between them, and is known for its robustness to certain transformations between the series such as local shifting and warping [28].

### E. HTM Modeling Approach

Hierarchical Temporal Memory is a theory on the mechanisms of sequence learning and prediction as they occur in the mammalian neocortex. It is believed that the neocortex is heavily involved in the process of learning, sometimes known as the ‘seat of intelligence’ in mammals [22]. NuPIC, an open source machine learning project developed by Numenta

inc. [9], implements the HTM algorithms, which are used in this study to learn separate models from the motions of each subject. The main hypotheses explored in this study is that human subjects each have unique control patterns which emerge over time, and that the subjects can be identified by recognizing these patterns.

The HTM learning algorithms implemented in NuPIC were designed specifically for temporal data, to form models in an online fashion, learning incrementally one data point at a time. In this case the models were constantly taking in sequences of past control behaviors and plant states in 1 to 4 dimensions and outputting predictions and anomaly scores. The following paragraphs describe in detail the functionality of the critical steps of the HTM algorithm as they are shown in Figure 5.

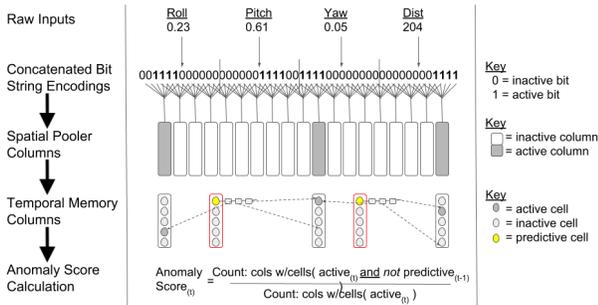


Fig. 5: HTM algorithm description

### Encoding Step

First, all input values are converted into binary bit strings. However, the raw data inputs are not literally converted to the binary language, but to bit strings meant to capture their semantic traits. When two inputs have semantic similarity they have similar encoding vectors, with overlapping ON-bits (those equal to 1). For scalar numbers this is simple, the closer two numbers are the more semantically similar they are. For instance the bit string encoding for the 'pitch' metric has its four consecutive ON-bits all the way to the left (111100000000) because its value of 0.05 is in the lowest range. A pitch value of say 0.10 may generate a similar yet non-identical encoding with shared ON-bits, like (011110000000). The value 0.10 is similar to 0.05 in that it is also low, but different enough to produce a slightly different encoding. The total encoding is formed when the bit strings for all inputs are concatenated into one [23].

### Spatial Pooling Step

Once the encoding of all input signals is complete, it is then passed into the HTM Spatial Pooler (SP), which activates a small subset of columns to represent the spatial traits of the encoding. This is illustrated in phase 2 of Figure 5. In NuPIC the default number of SP-columns used across many applications is 2048 [24], of which only 40 (~2 percent) are activated at each time step. Each column is connected to a subset of the encoding known as its 'receptive field', and the more of a columns' receptive field encoding bits are ON the better chance it has to activate. This ON-bit proportion is calculated as the 'overlap score' for each SP-column, and the top 2 percent of columns in overlap score activate and inhibit all others [25]. Since the connections from the SP-columns to

the encoding bits are random the actual receptive fields would not look so ordered as they do in the figure, which was done for visual clarity. The SP-columns activated at the current time step are shown in gray.

### Temporal Memory Step

The set of activated columns from the SP is passed into the HTM Temporal Memory (TM), which activates certain cell(s) within each column and sets other cells to 'predictive' based on these current activations. This is illustrated in phase 3 of Figure 5. While the SP-columns each represent a semantic spatial quality of the input (like 'small negative pitch value' or 'high positive roll value'), the TM-cells each represent the current temporal contexts of these traits [26]. For instance small negative deflections of the control inceptor in pitch may indicate a different pattern if preceded by three high positive deflections in the same channel, than by three other small negative ones. The exact same set of columns representing 'small negative pitch value' will activate in both cases, but different TM-cells within the columns will activate and yield completely different predictions. With 40 active columns and 32 cells per column, the TM can potentially represent the same small negative pitch value or high positive roll value in  $32^{40}$  unique ways [24]. This allows identical input sequences to generate vastly different predictions from the same model, all depending on the given context.

TM-cells learn to predict their own activity by connecting to other cells that activated just prior. This is also illustrated in phase 4 of Figure 5. For instance a cell that represents the letter 'C' may become active after seeing the letters 'A, B' because it has followed them in the past and thus formed connections to those 'A, B' cells. This means that when the TM activates a set of cells it likely also puts another set of cells into the 'predictive state', each anticipating their own activations [23]. The TM-cells shown in gray are active at the current time and those in yellow are predictive, expecting to activate at the next time step. The yellow are predictive because a high number of the cells they connect to are currently active, like 'C' getting ready after seeing 'A, B'. This set of yellow cells constitutes the HTM's prediction for the next time step.

### Anomaly Score Step

When the next data point is received, the proportion of SP-columns that activate and contain no predictive cells is the anomaly score. If all activated SP-columns contain predictive cell(s) from the prior time step then the anomaly score is 0, no surprise at all. If, however, none of these SP-columns contain predictive cells then the anomaly score is 1.0, and this new data point has surprised the system entirely. The anomaly score always falls between 0 and 1, with some of the activated SP-columns expecting it and the others not [24]. The NuPIC Anomaly Score (NAS) is calculated as shown in Equation 1.

$$\text{NAS} = \frac{\text{number of active columns with no predictive cells}}{\text{number of active columns}} \quad (1)$$

NAS represents the unpredicted proportion of the observed input at each time step. More precisely it is the number of SP-columns that activated and contain no predictive cells from the prior time step. What determines how anomalous an input

is to a model - is the set of connections formed between its TM-cells within their distal dendrite segments. Different subjects generate different sequences, which leads to different connections formed and different predictions made. In this way, a sequence may be completely anomalous to one model while being completely unsurprising to another. The lower the anomaly scores produced, the better the match between the sequence and the model is.

Another core metric used in this research is referred to as Mean NuPIC Anomaly Score (MNAS). It is calculated as shown in Equation 2.

$$MNAS = \frac{\text{length of data set}}{\sum_{n=1}^{\text{length of data set}} NAS(n)} \quad (2)$$

MNAS score is instrumental to calculating the Rank Score metric, which is described in the following subsection of the paper. As it will be shown in later sections of the manuscript, MNAS is also critical to facilitating the real-time applications of the proposed HTM-based methodology.

#### F. Metrics

The essence of the Rank Score (RS) based metric used in this research, is to measure how correctly the matching subjects' models were ranked out of the total pool of subjects. RS quantifies how highly the system ranks a matching subject's training data set (the HTM model built from it) to its corresponding test set. RS score is computed according to Equation 3:

$$RS = 1 - \frac{MRP - 1}{N} \quad (3)$$

where  $N$  is the total number of subjects participated in the experiments, and  $MRP$  is the Matching Rank Position. The latter is the measure of how highly the matching subject's model was ranked out of total of  $N$  models. In short, the  $MRP$  score is inversely proportional to the corresponding  $MNAS$  value. The higher the  $MNAS$  value of a given model for a particular subject test-data, the lower the model's  $MRP$ .

To better illustrate the relationship between  $MRP$  and RS, consider the following: if HTM model 16 was ranked 5-th on the subject-16 test set, the corresponding RS score is 0.895; if it were ranked 2nd the RS would be 0.974, and only if ranked 1st would the RS be 1.0. If this identification task were done with random guessing, the average RS would be 0.5 overall.

#### IV. RESULTS AND ANALYSIS

In this section identification performance as measured by Rank Score (RS) is summarized for all compared measures: HTM, DTW and EDR. Each method was evaluated in both static and real-time data settings, with one subsection dedicated to each.

The main finding is that HTM outperforms the competition by far in the real-time setting, with the data provided in an incremental manner. In the static setting, with the data provided all at once in a batch, the competition is close between HTM and DTW.

#### V. STATIC DATA

In the static data setting, all three algorithms are tested across a range of data granularity levels, and with all eligible field-sets.

Performance is shown across the three algorithms on static data for 10 and 20 HZ in Figures 6 and 7, and in Tables II and III. It's clear that in the static setting DTW is competitive with HTM. Each algorithm is shown with its most favorable field-set, as found from Figures 11 through 13. Please note that the choice of the aggregation levels shown in this comparison is dictated by the fact that the natural frequency of human manual control behavior does not exceed 1.5 Hz in this control scenario.

DTW	EDR	HTM
mean = 0.88	mean = 0.77	mean = 0.93
median = 0.97	median = 0.89	median = 0.96

TABLE II: Rank Scores by Algorithm (10 HZ) - Static Data

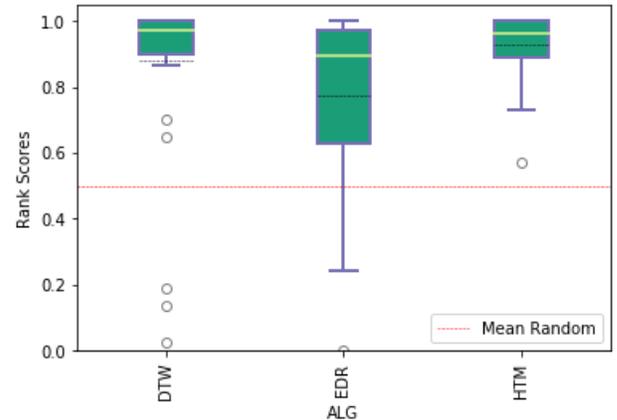


Fig. 6: Rank Scores by Algorithm (10 HZ) - Static Data

DTW	EDR	HTM
mean = 0.88	mean = 0.78	mean = 0.94
median = 0.97	median = 0.91	median = 0.97

TABLE III: Rank Scores by Algorithm (20 HZ) - Static Data

However, for the sake of completeness of the analysis, each algorithm was also evaluated across all granularity levels as shown in Figures 8 through 10.

DTW reaches its peak performance at 10 HZ, with EDR virtually tied between 33.33, 50 and 100 HZ. Neither are as high or as stable across the board as HTM, which peaks at 1 HZ while performing comparably up to 100 HZ with a slight dip at 10 HZ.

Each algorithm was also evaluated across field-sets, shown in Figures 11 through 13. The granularity level was set to 10 HZ, the most favorable to HTM's main competitor DTW.

It's clear that both HTM and DTW do best with the set comprised of all four fields ('roll', 'pitch', 'z' and 'dist'), while EDR does best with 'z'. Since EDR was only designed for 1-dimensional vectors it was only tested on the four 1-dimensional field-sets.

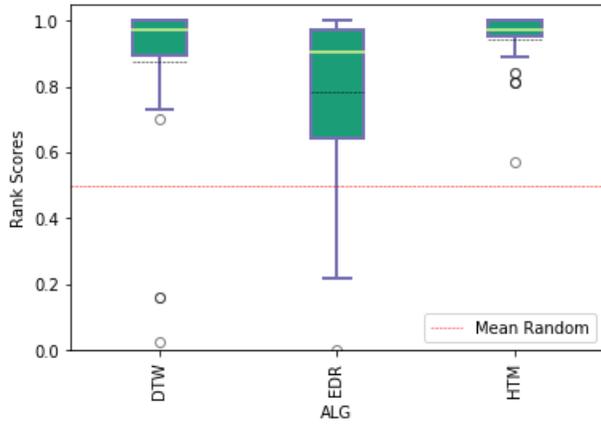


Fig. 7: Rank Scores by Algorithm (20 HZ) – Static Data

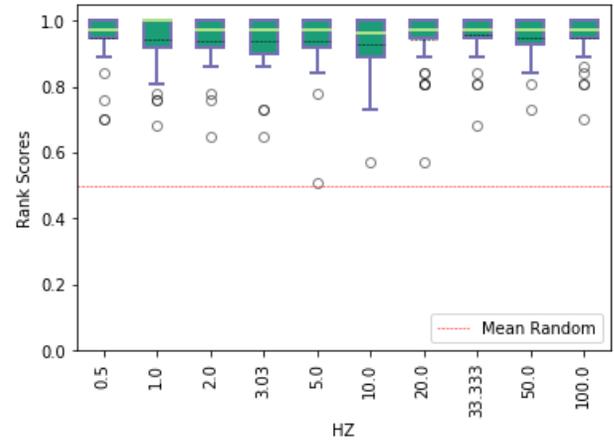


Fig. 10: HTM – Rank Scores by HZ

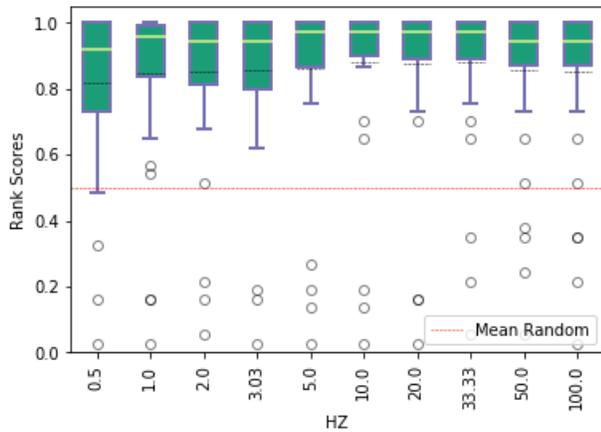


Fig. 8: DTW – Rank Scores by HZ – Static Data

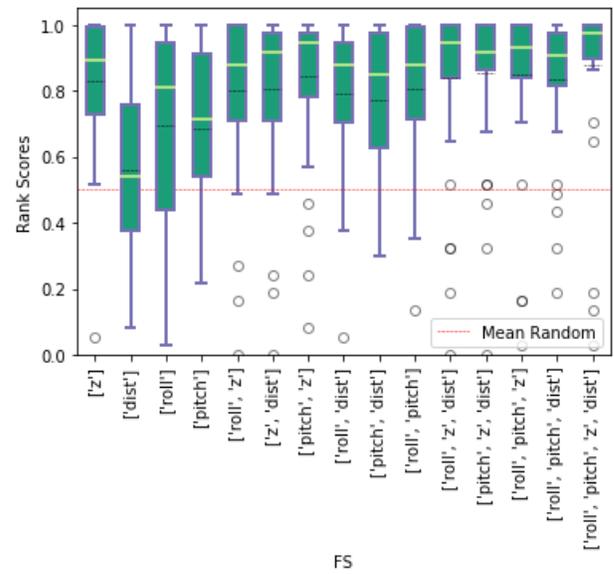


Fig. 11: DTW – Rank Scores by Field-Set – Static Data

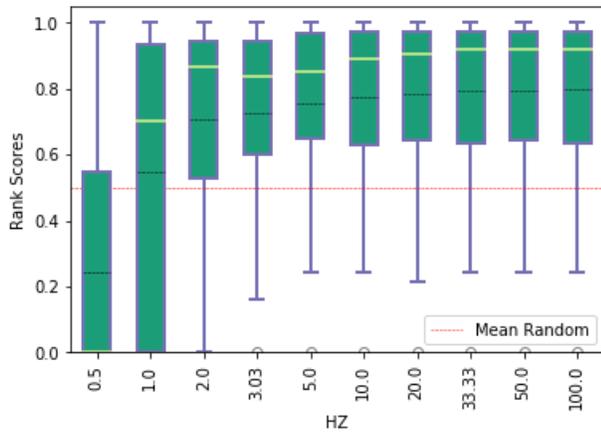


Fig. 9: EDR – Rank Scores by HZ – Static Data

VI. REAL-TIME DATA

In the real-time data setting the DTW and EDR algorithms are tested across a range of window sizes, at only the 100 HZ granularity level. Since the raw data was sampled at 100 HZ the data is not pre-processed at all, as in a real-time setting. Each algorithm is again shown with its most favorable field-set, as found from Figures 11 through 13.

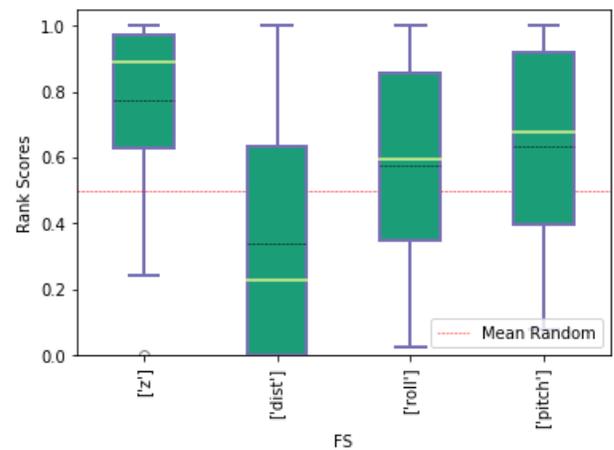


Fig. 12: EDR – Rank Scores by Field-Set – Static Data

In order to robustly test DTW and EDR in real-time settings, a number of different sliding window sizes were tried. At a

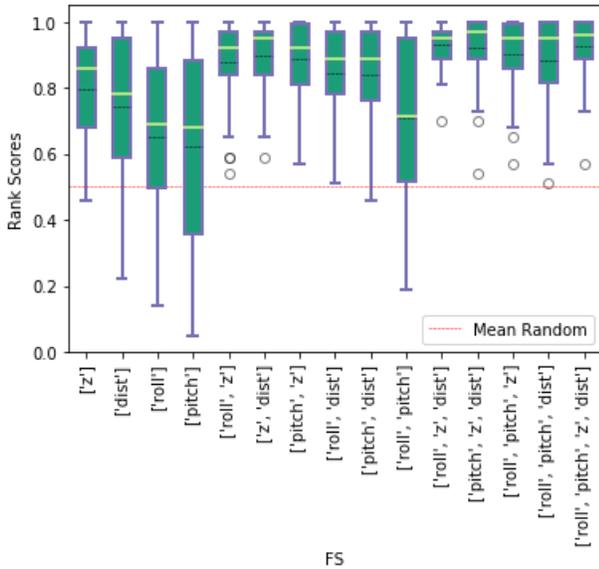


Fig. 13: HTM – Rank Scores by Field-Set

window size of  $n$ , the test set at each time step consisted of only the last  $n$  data points. The most favorable window sizes for DTW and EDR were both found to be 1000, which corresponds to 10 seconds or runtime.

The performance across window sizes is shown for DTW and EDR in Figures 14 and 15. The "NA" window represents the static data setting (no window applied) for comparison.

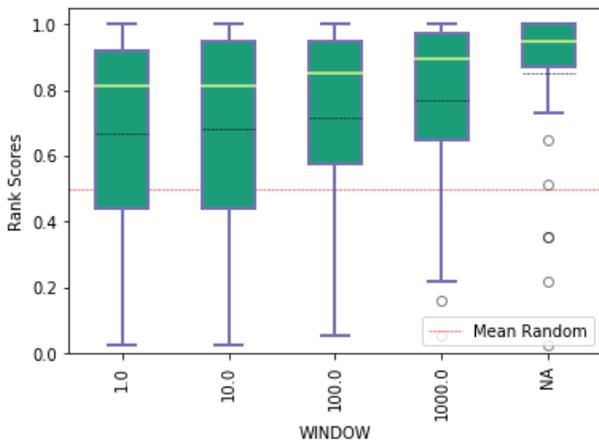


Fig. 14: DTW – Rank Scores by Window (100 HZ)

Though DTW and EDR still perform well above random and improve as window size grows, they don't do as well with real-time data as they do with static. The drop is especially pronounced for DTW. This stands to reason as they were designed to operate on static time series, not deal with incremental change. HTM however is designed to ingest data in an incremental fashion, so there is no static setting.

Performance is shown across the three algorithms in Figure 16 and Table IV, using their most favorable window sizes, as found in Figures 14 and 15. HTM has a fixed window size of 1, since it is incremental by nature and thus always sees only

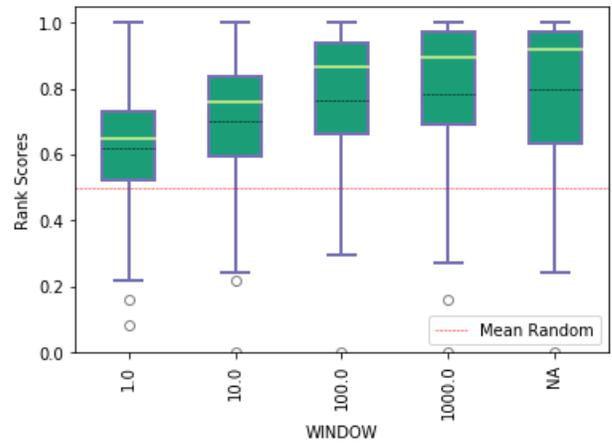


Fig. 15: EDR – Rank Scores by Window (100 HZ)

the current data point. This comparison makes it clear how HTM pulls away from the competition in the real-time data setting.

DTW	EDR	HTM
mean = 0.77	mean = 0.78	mean = 0.95
median = 0.89	median = 0.89	median = 0.97

TABLE IV: Rank Scores by Algorithm - Real-Time Data

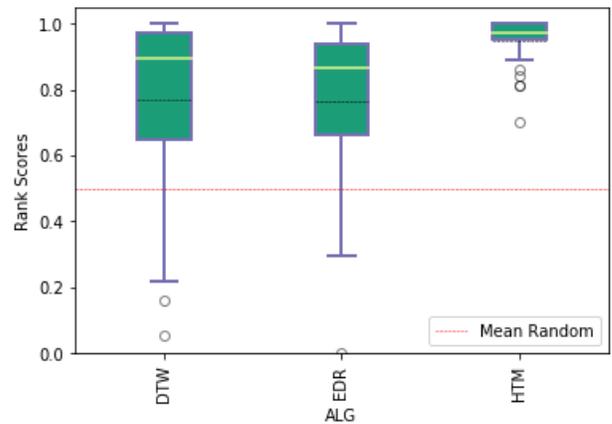


Fig. 16: Rank Scores by Algorithm (100 HZ) – Real-Time Data

## VII. CONCLUSION

The findings of this study suggest the viability of the proposed Motion-Print modality and HTM-based identification method. The HTM method clearly outperformed DTW and EDR in the real-time setting, and tied with DTW for best in the static setting.

Furthermore HTM's continuously accrued memory structure makes it highly adaptive to fast-changing patterns and thus broadly applicable across problem settings, including prediction and anomaly detection from streaming data.

In the setting of human-operator monitoring, HTM is naturally equipped both to detect when anomalous control motions

set in and to predict what the given subject would normally do in that context, having been trained on many sequences of their prior actions. This means that if subjects' behavior changes from normal to abnormal HTM models can detect when it happens in real-time, with a unique concept of normality learned for each subject. Given the preliminary nature of this study, further investigation of this aspect of the HTM-based identification algorithm is required.

While Artificial Neural Networks are often based on the Multilayer Perceptron (MLP) model, HTM networks derive their internal mechanics entirely from neuroscience - specifically how pyramidal cells in the neocortex learn time-based patterns [24]. The neocortex has been found to be remarkably modular and uniform across all mammals, implying that evolution has landed on these core mechanisms as robust and efficient for real-time learning and prediction within the natural world [22]. It seems fitting for such naturalistic mechanisms to guide artificial systems meant to model human controls.

The Motion-Print biometric modality and HTM method proposed in this study show intellectual merit, achieving promising results despite the small size of data generated, the largely homogeneous group of subjects and the generic nature of the control task. Given its capacity for canonical real-time learning and prediction from multi-dimensional data streams and the results shown here, HTM is believed to have strong potential for broader impact in human control modeling at large. As a task-agnostic methodology, it can be seamlessly applied to other objectives with sequential data – such as workload assessment and monitoring for the onset of sickness and distraction in human operators.

## REFERENCES

- [1] J.A. Unar and W.C. Seng and A. Abbasi, *A review of biometric technology along with trends and prospects*. Pattern Recognition, vol 47, pp. 2673-2688, 2014.
- [2] S.K. Sahoo and T. Choubisa and S.R. Prasanna, *Multimodal biometric person authentication: A review*. IETE Technical Review, vol 29, pp. 54-57, 2012.
- [3] S. Turner, *2019 Data Breaches; The Biggest Breaches of the Year*. [Blog] Data Breach and Technology, [Accessed 10 Dec. 2020].
- [4] *Behavioral Biometrics Market Value Will more than Double by 2020*. [Blog] Technavio, [Accessed 15 Dec. 2020].
- [5] D. Thakkar, *Top Five Biometrics: Face, Fingerprint, Iris, Palm and Voice*. [Blog] Bayometric, [Accessed 20 Dec. 2020].
- [6] T.J. Neal, and D.L. Woodard, *Surveying biometric authentication for mobile device security*. Journal of Pattern Recognition Research, 1(74-110), p.4, 2016.
- [7] A. Lumini and L. Nanni, *Overview of the combination of biometric matchers*. Information Fusion, vol 33, pp.71-85, 2017.
- [8] A. Mahfouz and T.M. Mahmoud and A.S. Eldin, *A survey on behavioral biometric authentication on smartphones*. Journal of information security and applications, vol 37, pp.28-37, 2017.
- [9] M. Taylor and S. Purdy and S. Ahmad and M. Lewis and C. Surpur (2018). *NuPIC* Numenta.
- [10] K. Zaychik, *Intelligent systems approach for automated identification of individual control behavior of a human operator*. Dissertation, State University of New York at Binghamton, 2009.
- [11] R.A. Hessn, *A motion-based theory for analyzing human control behavior*. Advances in Man-Machine Systems Research, vol 2, pp. 129-175, 1985.
- [12] R. Hosman, *Pilot's perception and control of aircraft motions*. Dissertation, Delft Technical University, 1996.
- [13] D.L. Kleinman and S. Baron and W.H. Levison, *An optimal control model of human response. Part I: Theory and Validation*. Automatics, vol 6, pp. 357-359, 1970.
- [14] B. Armstrong and M. Ruiz-Blondet and N. Khalifian and K. Kurtz and Z. Jin, and S. Laszlo, *Brainprint: Assessing the uniqueness, collectability, and permanence of a novel method for ERP biometrics*. Neurocomputing, vol 166, pp.59-67, 2015.
- [15] M. Ruiz-Blondet and Z. Jin and S. Laszlo, *CEREBRE: A Novel Method for Very High Accuracy Event-Related Potential Biometric Identification*. IEEE Transactions on Information Forensics and Security, vol 11(7), pp.1618-1629, 2016.
- [16] M. Ruiz-Blondet and Z. Jin and S. Laszlo, *Permanence of the CEREBRE brain biometric protocol*. Pattern Recognition Letters, vol 95, pp.37-43, 2017.
- [17] C. Shen and Z. Cai and X. Guan and Y. Du and R. Maxion, *User Authentication Through Mouse Dynamics*. IEEE Transactions on Information Forensics and Security, vol 8(1), pp.16-30, 2013.
- [18] A. Motwani and R. Jain and J. Sondhi, *A Multimodal Behavioral Biometric Technique for User Identification using Mouse and Keystroke Dynamics*. International Journal of Computer Applications, vol 111(8), pp.15-20, 2015.
- [19] T. Katerina and P. Nicolaos, *Mouse behavioral patterns and keystroke dynamics in End-User Development: What can they tell us about users' behavioral attributes?*. Computers in Human Behavior, vol 83, pp.288-305, 2018.
- [20] Logitechg.com. (2019). *Extreme 3D Pro Joystick - Logitech Gaming*. [online] [Accessed 12 Oct. 2019].
- [21] R. Angryk and P. Riley and S. Naduvil-Vadukootu, *Evaluating pre-processing strategies for time series prediction using deep learning architectures*. In: In The Thirtieth International Flairs Conference., 2017.
- [22] J. Hawkins and S. Blakeslee, *On intelligence*. New York: Macmillan, 2007.
- [23] J. Wu and W. Zeng and F. Yan, *Hierarchical Temporal Memory method for time-series-based anomaly detection*. Neurocomputing, vol 273, pp.535-546, 2018.
- [24] J. Hawkins and S. Ahmad and S. Purdy and A. Lavin, *Biological and machine intelligence*. Redwood City, 2016.
- [25] Y. Cui and S. Ahmad, and J. Hawkins, *The HTM Spatial Pooler—A Neocortical Algorithm for Online Sparse Distributed Coding*. Frontiers in Computational Neuroscience, vol 11, 2017.
- [26] Y. Cui and S. Ahmad, and J. Hawkins, *Continuous Online Sequence Learning with an Unsupervised Neural Network Model*. Neural Computation, 28(11), pp.2474-2504, 2016.
- [27] L. Chen and MT. Ozsu, and V. Oria, *Robust and Fast Similarity Search for Moving Object Trajectories*. Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, 2005.
- [28] D.J. Berndt and J. Clifford, *Using dynamic time warping to find patterns in time series*. KDD workshop, vol 10(16), pp.359-370, 2017.
- [29] J. Serra and J. Arcos, *An Empirical Evaluation of Similarity Measures for Time Series Classification*. Knowledge Based Systems, vol 67(1), pp.305-314, 2014.
- [30] J. Lozano and U. Mori and A. Mendiburu, *Similarity Measure Selection for Clustering Time Series Databases*. IEEE Transactions on Knowledge and Data Engineering, vol 8(1), pp.181-195, 2016.