Comparison of Data Representations and Machine Learning Architectures for User Identification on Arbitrary Motion Motion Sequences

Christian Schell
Human-Computer Interaction (HCI) group
University of Würzburg, Germany

Andreas Hotho
Data Science Chair
University of Würzburg, Germany

Marc Erich Latoschik
Human-Computer Interaction (HCI) group
University of Würzburg, Germany

Abstract—Reliable and robust user identification and authentication are important and often necessary requirements for many digital services. It becomes paramount in social virtual reality (VR) to ensure trust, specifically in digital encounters with lifelike realistic-looking avatars as faithful replications of real persons. Recent research has shown great interest in providing new solutions that verify the identity of extended reality (XR) systems. This paper compares different machine learning approaches to identify users based on arbitrary sequences of head and hand movements, a data stream provided by the majority of today’s XR systems. We compare three different potential representations of the motion data from heads and hands (scene-relative, body-relative, and body-relative velocities), and by comparing the performances of five different machine learning architectures (random forest, multilayer perceptron, fully recurrent neural network, long-short term memory, gated recurrent unit). We use the publicly available dataset "Talking with Hands" and publish all our code to allow reproducibility and to provide baselines for future work. After hyperparameter optimization, the combination of a long-short term memory architecture and body-relative data outperformed competing combinations: the model correctly identifies any of the 34 subjects with an accuracy of 100% within 150 seconds. The code for models, training and evaluation is made publicly available. Altogether, our approach provides an effective foundation for behaviometrics-based identification and authentication to guide researchers and practitioners.

I. INTRODUCTION

The past decade has seen a significant boost in new developments and technologies for Virtual, Augmented, and Mixed Reality (VR, AR, MR: XR for short). A large variety of consumer-grade hardware devices and software solutions are now affordable, opening-up interesting and promising out-of-lab use-cases. Here, many of these use-cases benefit from social XR [1]. Social XR provides avatar-embodied encounters of users in an unlimited number of different virtual environments. Most prominently, social XR promises mediated human-to-human interaction over distances while supporting the richness of verbal and non-verbal interaction based on users’ avatars. Altogether, these benefits render social XR highly attractive, not least because of the Covid-19 pandemic and the necessary reduction or even avoidance of physical contacts.

Recently, technological advances in creating lifelike virtual humans have made considerable progress. The MetaHuman project [2] provides a variety of high-fidelity digital humans indistinguishable from real persons. In addition, photogrammetry-based methods [3]–[5] provide individualized near-photorealistic avatars that faithfully replicate the real physical visual appearance of users. Taken together, questions about the identity and authenticity of the avatars we interact with become crucial: How can we trust the digital alter egos of others and verify their real identity? A general question that becomes more and more important during times of fake news and deep fake approaches.

This article compares methods to leverage behaviometrics for user identification in XR contexts. Specifically, our solution uses movement data of head and hands and analyses these movements to detect specific characteristics of movement patterns attributable to individual real persons. Head and hand movements typically are provided by the majority of XR devices. They are used to capture essential user behaviors to generate accurate visual feedback and allow interactions. Hence, our approach is tailored for XR use-cases and specifically beneficial for social XR. Here, reliable identification of real users, i.e., the physical partners controlling potentially arbitrary avatars, becomes paramount to assure authenticity and trust. Note that in this work we assume that the recorded data reflects the actual movements of the user — this might not necessarily be the case in scenarios where movement data is deliberately altered, either benevolently (e.g., to improve interactions [6], [7]) or maliciously (e.g., to fool the system).

Previous work already shows that movement data captures individual user characteristics in principle and that it can be used for novel identification and authentication approaches. Researchers started to employ basic machine learning algorithms, and few works even employ deep learning models. However, several open questions remain about effective data representation and pre-processing, as well as about appropriate choices of classification architectures and hyperparameters. There is also very little work using publicly available datasets, which prevents reproducing results and verifying previous findings.

Our approach compares five different prominent machine learning techniques for the identification task, i.e., random forest, multi-layer perceptron, and three types of deep recurrent neural networks (fully recurrent neural network, long short-
term memory and gated recurring unit). We also highlight the importance of feature pre-processing of movement data: All three techniques are tested with scene-relative, body-relative, and body-relative velocity data from the publicly available "Talking With Hands" dataset [8]. The goal of this paper is to demonstrate the profound effects of these different approaches to guide future research and the development of practical applications: after an extensive hyperparameter optimization, deep learning approaches outperform the random forest and multilayer perceptron and achieve 100% identification accuracy within 150 seconds on test sequences on body-relative as well as body-relative velocity data. We also show that no machine learning models generalizes well using scene-relative data, which renders this data representation problematic for identification purposes. Altogether, our contribution provides an effective foundation for behaviometric-based identity checks based on deep learning of arbitrary motion data sequences for XR use-cases.

II. RELATED WORK

We have summarized relevant related work in Table I to guide the following discussion. Several works started to exploit motion data as generated by typical VR setups [9]–[19]. Such setups often provide at least a basic three-point tracking: head tracking is essential to enable visual immersion, while hand tracking is required for user interaction. The used sensor technology will often vary in its principle physical measure (position, orientation, acceleration, or forces etc.) and characteristics, i.e., in precision, accuracy, sampling rate, latency, and drift. Often, the resulting motion data is highly interrelated independently of the exploited physical measure, and the characteristics often are tailored to support the same use-cases. Hence it seems safe to assume that most would also reliably incorporate essential individual movement characteristics of users. For example, Rogers et al. [20], Li et al. [21], and Shen et al. [22] show that biometric motion data from other sources, like the IMU sensors in Google Glasses, can be used for identification purposes as well.

In this paper we focus on identification. Note, that identification and authentication are not the same. Like Miller et al. [9] we also follow the definition by Rogers et al. [20]: Authentication focuses on independently identifying a user from any other potential user and is usually most concerned with avoiding false-positive classifications. Identification is the task of identifying a user from a known group of users and equally rates false-positive and false-negative classifications.

Previous work has mostly focused on identifying users who perform instructed tasks and correlating movement patterns. These range from fairly precise tasks, such as throwing a ball, to relatively loose instructions, such as troubleshooting a virtual robot [19]. However, in our approach we follow Miller et al. who could show that identification is also possible based on arbitrary movement patterns, e.g., by simply observing users interacting with an application not intended to trigger identifying behaviour [9]. This provides much broader applicability since it can be used to establish identification services running in the background based on the continuously produced data streams. However, it also poses increased challenges to successfully detect identifying patterns in arbitrary movement data streams.

The majority of prior work was using groups of 14 to 95 individuals to develop identification methods. The recent work by Miller et al. [9] is the first to provide convincing evidence that identification can be performed even on a large group of 511 individuals. The authors have trained a random forest model on raw motion data referenced in the coordinate system of the 3D scene (scene-relative). The model classified 95% of the withheld test sequences correctly. The same methods were evaluated by Moore et al. [19] on a different dataset and the authors concur that movement data can be identifying. Since scene-relative data makes it easy for machine learning models to adapt to features that capture session characteristics instead of user characteristics Miller et al. propose to use velocity features for identifying users in multiple sessions. Moore et al. seem to contradict this consideration by demonstrating that their models perform considerably worse when trained on velocity data. However, Moore et al. computed the velocities based on the scene-relative data, so the resulting features still encode session specific characteristics, since the orientation of the user remains captured within the data. We hypothesize that velocity data should be computed on coordinates that are referenced to some key point on the user (e.g., their head) to be viable for identification with machine learning models. Also, since patterns in velocity data might simply be too difficult to capture for basic machine learning models, we also follow the suggestion of Miller et al. and investigate deep learning approaches.

Altogether, we identified three common shortcomings of previous work and cover them in this paper: First, we follow Miller et al. [9] and Liebers et al. [18] in arguing that proper pre-processing of motion data features is important to help classification models learn user-specific instead of session-specific patterns. This might be a severe potential weakness of the — otherwise convincing — results of [9], yet little work has been done to systematically explore this specific matter. Second, there are only a few works exploring deep learning methods and their beneficial implicit feature processing and selection [14], [18], [25]. However, their approaches lack a convincing hyperparameter search and comparison to basic classifiers used by other authors, such as random forest. Consequently, it remains unclear how increased complexity and computational cost of deep learning architectures can be weighed against potentially better performances. Lastly, we think it is problematic that except of Liebers et al. [23] and Miller R. et al. [25] none of the previous work uses publicly available data which makes it impossible to compare different approaches. In addition, to our knowledge, only Miller R. et al. [25] have published their code for data processing and training, which compromises verifiability and reproducibility of all other previous publications.

Against this backdrop, we investigate the effects of three different motion data representations (i.e., scene-relative data,
| authors               | classifier            | task                                         | data representation                           | dataset  | device          |
|----------------------|-----------------------|----------------------------------------------|----------------------------------------------|----------|-----------------|
| Rogers et al. (2015) | random forest         | ident.: watching rapidly changing images    | acc. & orient. of head + eye blinking       | N=20; unpubl. | Google Glass   |
| Li et al. (2016)     | distance-based        | auth.: nodding to music                      | acceleration of head                         | N=95; unpubl. | Google Glass   |
| Mustafa et al. (2018)| logistic regression,  | auth.: walking                               | acc. & orient. of HMD                        | N=23; unpubl. | Google VR      |
| Kupin et al. (2019)  | nearest neighbor,     | auth.: ball throwing                         | SR of right controller                       | N=14; unpubl. | HTC Vive       |
| Pfeuffer et al. (2019)| random forest, svm    | ident.: point, grab, walk, type               | SR, SRV of HMD & controllers                 | N=22; unpubl. | HTC Vive       |
| Shen et al. (2019)   | distance-based        | auth.: walking a few steps                   | acc. & orient. of head                       | N=20; unpubl. | Google Glass   |
| Ajit et al. (2019)   | nearest neighbor,     | auth.: ball throwing                         | SR of HMD & contr.                           | N=33; unpubl. | HTC Vive       |
| Mathis et al. (2020) | fully conv. network   | auth.: interaction with a cube               | SR of controllers                            | N=23; unpubl. | HTC Vive       |
| Miller M. et al. (2020)| random forest, knn, gbm| ident.: watching 360° videos and answering questionnaire | SR of controllers and head | N=511; unpubl. | HTC Vive       |
| Miller M. et al. (2020)| distance-based      | auth.: ball throwing                         | SR, SRV, trigger position of controllers     | N=41 publ. | HTC Vive & Vive Cosmos, Oculus Quest |
| Olade et al. (2020)  | nearest neighbor      | auth. & ident.: grab, rotate, drop balls and cubes | SR of HMD & contr. + eye gaze               | N=15; unpubl.* | HTC Vive       |
| Miller M. et al. (2021)| siamese nn          | like [15]                                    | SR of HMD & contr.                           | like [15] | like [15]       |
| Liebers et al. (2021)| LSTM, MLP            | ident.: bowling, archery                     | BR of HMD and controllers                    | N=16; publ. | Oculus Quest    |
| Moore et al. (2021)  | random forest, knn, gbm| ident.: robot troubleshooting in VR          | SR and SRV of controllers and HMD            | N=60; unpubl. | HTC Vive       |
| Miller M. et al. (2022)| siamese nn          | like [15]                                    | SR of HMD & contr.                           | like [15] | like [15]       |
| Miller M. et al. (2022)| siamese nn          | like [15]                                    | SR of HMD & contr.                           | like [15] & [13] | like [15] |
| this paper           | random forest, MLP, FRNN, LSTM, GRU | ident.: conversation                    | SR, BR, BRV of head and hands               | N=34; publ. | 3-point tracking from full body mocap |

*authors indicated a publication of their data, but there has been none so far.

**TABLE I**

**RELEVANT WORK TARGETING IDENTIFICATION OR AUTHENTICATION BASED ON MACHINE LEARNING OF MOVEMENT DATA; N DESCRIBES THE NUMBER OF INDIVIDUAL USER; SR = SCENE RELATIVE, SRV = SCENE RELATIVE VELOCITY, BR = BODY RELATIVE; HMD = HEAD MOUNTED DISPLAY.**

III. METHODS

A. Posture and Movement data representation

3D engines, such as Unity or the Unreal Engine, usually provide access to a steady stream of positions and rotations of tracked devices or joints from the 3D motion sensors. This data is always specified with respect to a coordinate frame of reference. In general, such reference frames are free to choose, and they can be mapped from one to the other using appropriate 3D-transformations. However, often the engine’s

*Authors indicated a publication of their data, but there has been none so far.*

Body-relative and body-relative velocity data combined with five prominent machine learning architectures (random forest, multi-layer perceptron and three different types of recurrent neural networks). We propose a structured approach to train capable deep learning models for behaviometric-based user identification. To allow replication of our work we use the publicly available "Talking With Hands" dataset from [8] and publish the code we have developed for the data preprocessing and machine learning steps under [to be added after review].
scene root serves as the default reference. We therefore refer to this type of data as scene-relative (SR) data. This choice of the frame of reference corresponds to the approaches of Miller et al. [9], Ajit et al. [13], Moore et al. [19] and Mathis et al. [14]. SR data frames consists of 21 features: (pos-x, pos-y, pos-z, rot-x, rot-y, rot-z, rot-w) × (head, wrist-left, wrist-right) all given with respect to the scene root as frame of reference, the rotations are represented as quaternions.

However, SR-encoded data does not only incorporate user-specific characteristics directly correlating to the user’s identity. For example, the user’s absolute position and orientation in the VR scene is, at least partially, arbitrary: It can change for the same person because of new calibrations and frames of reference, or due to specific interactions and navigation (e.g., teleportation). Unfortunately, it is easy for a model to optimize towards these characteristics that are not user-specific instead of learning more complex movement patterns. To remove any information about the whereabouts of the user, we transform the coordinate system from scene-relative to the body-relative (BR) frame of reference: position and orientation of the wrists are defined with respect to the head, which results in the wrists being positioned and orientated independently of the user’s original scene position and orientation. The head’s positional features and the rotation around the up axis become obsolete and are therefore removed, which only leaves one quaternion encoding the head’s rotation around the horizontal axes. This has the effect that the data yields the same values for the same movement (e.g., waving), even if the user changes position or orientation within the scene in between. BR data consists of 18 features: (pos-x, pos-y, pos-z, rot-x, rot-y, rot-z, rot-w) × (wrist-left, wrist-right) + (rot-x', rot-y', rot-z', rot-w') of the head, all given with respect to the users head as frame of reference.

Even though BR data contains less information than SR data, it may still contain information that is irrelevant or even misleading for identifying the originator. For example, the data encodes the exact distance and angle of the hands to the head at each point in time: if a user repeats the same motion, the joints might follow the same trajectory, but start and end at different positions w.r.t. the body. In theory, machine learning models can learn to deal with such variances, but require an appropriate amount of training data. It might therefore be beneficial to remove even more information from the data by computing body-relative velocity (BRV) data: for positions we subtract the value of each feature with the value of the corresponding value of the previous frame, for rotations we rotate each quaternion by the corresponding inverted rotation of the previous frame. This way the data only contains a signal, when the user is actually moving. Pfeuffer et al. [12], Miller et al. [15] and Moore et al. [19] also use velocity data, but these are computed directly from SR data, which results in scene-relative velocity data that still encodes session characteristics such as the user’s orientation. The BRV data also consists of 18 features, just like the BR data: (pos-x’, pos-y’, pos-z’, rot-x’, rot-y’, rot-z’, rot-w’) × (wrist-left, wrist-right) + (rot-x’, rot-y’, rot-z’, rot-w’) of the head.

It might be worthwhile to further investigate acceleration or even jerk data as well. However, adding further data representations would increase the hyperparameter search space later described, so to keep the scope of this work within reasonable limits, we focus on the three presented data representations.

### B. Machine Learning Architectures

We compare the performances of the following five different machine learning architectures for the identification task:

1. Random Forest (RF)
2. Multilayer Perceptron (MLP)
3. Fully Recurrent Neural Network (FRNN)
4. Long-Short Term Memory (LSTM)
5. Gated Recurrent Unit (GRU)

Each architecture provides several parameters that configure either the architecture itself or the training process. These so called hyperparameters have profound impact on the capabilities of the resulting models and are individual to each problem and dataset. We provide an overview of the selected hyperparameters and the search space we explore in Table II.

### C. Dataset

We use the “Talking with Hands” dataset from [8], since it is to our knowledge the largest publicly available dataset containing human motion capture data. It contains approximately 20 hours of footage recorded with a full body motion capture system from 37 individual subjects recorded over 32 sessions. In each session, two subjects have been recorded performing several conversational tasks. A session consists of multiple

#### Table I

| Hyperparameter | Search Space |
|----------------|--------------|
| random forest  | n_estimators: 50 - 1,000, min_samples_leaf: 1 - 1,000 |
| MLP            | number of layers: 1 - 6, size per layer: 10 - 300 neurons, learning rate: 0.00001 - 0.01 |
| FRNN, LSTM, GRU| hidden size: 20 - 200 neurons, number of layers: 1 - 8, dropout: 0 - 0.6, learning rate: 0.0001 - 0.01 |

#### Table II

| Relevant Parameters Selected for the Hyperparameter Search - The Architectural Parameters Explored During Hyperparameter Search Stage 1. |
|-----------------------------|------------------|
| Hyperparameter | Search Space |
|-----------------|--------------|
| bins            | frames per bin: 10 - 1,350 frames |
| windows         | frames per second: 10, 30, 60, 90 fps, window size: 10, 100, 300 frames |

#### Table III

| Relevant Parameters Selected for the Hyperparameter Search - The Data Parameters Explored During Hyperparameter Search Stage 2: The Original Frame Rate is 90 FPS. |
|-----------------------------|------------------|
| Hyperparameter | Search Space |
|-----------------|--------------|
| frames per second: 10, 30, 60, 90 fps, window size: 10, 100, 300 frames |
takes, each containing either recordings of conversations of the two subjects, or of system calibration tasks performed by one subject.

The recordings contain tracking data for the full body. To emulate three-point tracking we chose three bones from the tracked skeleton that are closest to the sensors of a typical XR setup: we select the bone "b_head" to represent the HMD and the bones "b_l_wrist_twist" and "b_r_wrist_twist" for the respective controllers.

The authors of the dataset used a standardized skeleton to project each subject’s movements on to, which removes any explicit information about body proportions. This is different to typical XR setups using HMDs and controllers, where the tracking data gets projected directly onto virtual heads and hands and therefore reflects, for example, a person’s height. Consequently, in the data used in this paper there is no explicit information about individual body heights or arm lengths. However, body proportions incorporate significant information attributable to the identity of users. Hence it is safe to assume that the developed solutions could even be improved when this standardization by an enforced retargeting would not be applied.

We also considered evaluating our approaches with the data from Liebers et al. [23] and Miller R. et al. [25]. However, there is less than 4 minutes of recording per user and feature set in each of the datasets. This made it difficult for us to find a legitimate way to split the data into training, validation and test sets for our methods to allow a meaningful comparison.

We have also contacted the authors of Miller M. et al. [9] and Moore et al. [19] for access to their datasets. Unfortunately, due to missing user agreements neither authors were able to share their data.

D. Data Filtering and Splitting

Takes used for training and evaluating models should contain only recordings of actual conversations and exclude any calibration takes. Since the takes are not labeled, we have conversed with the authors of the dataset and followed their recommendation: only takes are selected that 1) are from scenes with two subjects, 2) have a minimum length of five minutes and 3) show a minimum indication of movement. The last point is necessary, because we encountered corrupted takes containing frozen armatures that do not move at all. The authors have been made aware of this issue and may update the dataset in the future.

We divide the dataset into three subsets: The training set is used for training the models. Due to overfitting, models may perform well on training data, but may not generalize well for unseen data. We therefore use a validation set to evaluate the performance during training and to compare performances of different configurations during hyperparameter search. Finally, we use the test set to report and discuss the results of the final models.

Consequently, we require a minimum of three takes per subject, one for each subset. For the split, we sort the takes of each session by their length and use the shortest for testing and the second shortest for validation. This way we end up with one take per subject for each validation and test set with a minimum length of 5 minutes. Five subjects appear in more than one session, for these we select the longest session and ignore the remaining sessions. From the 37 subjects, 34 remain after filtering. The length of footage per subject differs because the authors of the dataset did not enforce any constraints regarding the number or length of takes per session.

E. Data Sampling

Depending on the machine learning architecture the data has to be sampled and transformed into a format the algorithms can work with. In this section we describe two sampling approaches, one for RF and MLP, the other for the RNNs. Additionally, we define hyperparameters for each approach that control how many frames the machine learning algorithms can look into the past (Table III). In both cases features are scaled to zero mean and unit standard deviation computed from the training data set, which helps the machine learning training process.

1) Binned Samples for Random Forest and MLP: To transform the time series data into one-dimensional samples, we adopt the methodology of [9]: we bin the frames of each recording into chunks of a fixed time period and then reduce each chunk to a selection of five statistics per feature: minimum, maximum, mean, median and standard deviation. Consequently, a binned sample either has 105 features for SR data or 90 features for BR and BRV data.

2) Windowed Samples for Recurrent Neural Networks: The RNNs can work directly with time series data. The only pre-processing required is to organize the frames into windowed samples. For windowed samples we investigate two hyperparameters. Firstly, there is the window size. While in theory RNNs would work with arbitrarily long sequences, in practice there has to be a fixed window of frames to make training technically feasible. Secondly, there is the number of frames per second. The recordings of the Talking With Hands dataset are sampled at 90 frames per second (fps). To measure the effects of lower temporal resolutions we also downsample the data to lower frequencies. Consequently, the time period of a windowed sample depends on both hyperparameters. For example, a window size of 30 frames and a frequency of 10 fps results in a period of $\frac{30 \text{frames}}{10 \text{fps}} = 3$ seconds per sample.

F. Implementation and Training

All models are trained on a computing cluster of the computer science department of our university. We use PyTorch Lightning [26] to implement MLP, FRNN, LSTM and GRU in Python. For optimization we use Adam with the categorical cross-entropy as loss criterion. We train each model for a maximum of 300 epochs, as preliminary runs did hardly improve beyond that. Additionally, we stop training early when the train loss drastically deteriorates after an initial grace period. During training we save a snapshot of each model at its validation highpoint for later evaluation since performance can decline towards the end. Each PyTorch training job runs with
We separate the hyperparameter search for these two categories. We therefore switched to the minimum accuracy (Table II) for training. During this stage, the data hyperparameters (Table II) as well as 2) the data hyperparameters (Table III). We separate the hyperparameter search for these two categories into two stages to keep the search space manageable.

In the first stage, we use Optuna’s default sampler to propose configurations from the search space (as defined in Table II) for training. During this stage, the data hyperparameters remain fixed to a duration of one second (following [9]): binned samples are set to 90 frames per bin, windowed samples are set to a window size of 30 frames and a resolution of 30 frames per second.

In the second stage, we select for each combination the configuration that scored the best weighted accuracy on the validation data and continue with that configuration for the data hyperparameter search. In this stage we perform one training for each combination of the data hyperparameter search space.

H. Evaluation Metric

During initial trial runs we used the mean accuracy averaged over all subjects to pick winner models, which is in line with previous work that is concerned with identification tasks [12], [17]: The accuracy of one subject $s$ is given by the ratio of true positive classifications $TP$ to the number of all subject samples $n_s$: $Acc_s = \frac{TP_s}{n_s}$. The mean accuracy is the mean of the accuracy of all subjects $S$: $MeanAcc = \frac{1}{S} \sum_{s=1}^{S} Acc_s$.

However, we noticed that this metric is not suitable to select reliable models, since classes with high accuracy compensate for classes that the model produces mainly false negatives for. We therefore switched to the minimum accuracy as metric. The minimum accuracy is the smallest of all $Acc_s$: $MinAcc = \min(Acc_1, Acc_2, \ldots, Acc_S)$. This way we prefer models that work decently for all subjects over models that detect most, but not all, subjects.

IV. RESULTS

A. Hyperparameter Search

We performed the two hyperparameter stages and selected the winner models based on their minimum accuracy on the validation data. A table of the finally selected architectures and sample configurations can be found in the appendix. During the first stage of the hyperparameter search we retrained each of the nine architecture+data representation combinations with at least 100 different hyperparameter configurations, totaling over 1700 training runs with a combined computation time of about 45 weeks. For each configuration we select the model with the highest minimum accuracy for the second stage. In the second stage we retrained each model with every sample configuration listed in Table III, the results are reported in Figure 1 and Figure 2.

The exact configurations and results of all trainings of both stages are provided as CSV tables in the supplemental material.

![Stage 2 Results - Bin Samples](image1)

![Stage 2 Results - Window Samples](image2)

Fig. 1. Results of the data hyperparameter search for binned samples (see Table III), indicating the achieved minimum accuracy of each configuration on the validation data in percent.

Fig. 2. Results of the data hyperparameter search for windowed samples, sample length = \( \text{window size} \times \text{fps} \), indicating the achieved minimum accuracy of each configuration on the validation data in percent.

B. Evaluation

The trained models are evaluated with the 34 test takes (one take for each subject with a minimum length of 5 minutes). We report the resulting accuracy scores for each model in Table IV.
At least in theory deep learning models are able to learn user-specific information across all subjects; e.g., the FRNN+BR V model achieves 90% average classification accuracy of all 50 second clips, and classifies 100% of all 160 second clips. However, it remains unclear if the models learned user-specific characteristics (i.e., body height) in XR settings where body proportions are not normalized.

Table IV illustrates the accuracy of each model on the test data.

| Model     | Mean Accuracy | Minimum Accuracy |
|-----------|---------------|------------------|
| RF        | 82% (20%)     | 14% (0%)         |
| MLP       | 91% (3%)      | 41% (0%)         |
| FRNN      | 90% (5%)      | 42% (0%)         |
| LSTM      | 91% (4%)      | 49% (0%)         |
| GRU       | 91% (8%)      | 44% (0%)         |
| LSTM+BR   | 100%          | 100%             |
| GRU+BR V  | 100%          | 100%             |
| RF+BR     | 100%          | 100%             |
| MLP+BR    | 100%          | 100%             |
| RF+BR V   | 100%          | 100%             |
| LSTM+BR V | 100%          | 100%             |
| FRNN+BR V | 100%          | 100%             |

Fig. 3. Clip classification accuracy on the test data for each model averaged over all subjects; e.g., the FRNN+BR V model achieves 90% average classification accuracy of all 50 second clips, and classifies 100% of all 160 second clips.

High accuracies and little misclassifications suggest that SR data generally outperforms the other data representations. However, it remains unclear if the models learned user-specific or session-specific characteristics. Since the test data originates from the same session, we cannot eliminate either possibility. At least in theory deep learning models are able to learn user-specific characteristics and disregard the prominent session-specific characteristics (i.e., position and orientation of the subject in the scene). However, the hyperparameter search resulted in very short sample lengths for all SR models, which may be an indication that the models are not learning actual movements, but rather memorize positions and orientations of the subjects. To evaluate this, we created a second version of the SR test, which we call SR offset: we shift each subject along both horizontal axes by adding 0.5 meters to the X- and Z- positions. After re-evaluating the models on these SR offset data, the results show that SR models are completely thrown off (Table IV), when the subjects are positioned differently than during training — models trained on BR and BR V data remain unaffected, since the shifting has no effect on their features.

We assume that this constraint makes SR data of little interest for most scenarios and therefore focus on BR and BR V data in the further analysis. Here, the GRU+BR V model achieves with 86% the highest mean accuracy on the test data. All RNN models perform generally better than the RF or MLP models, with the exception of the RF+BR model that ranks third overall.

For reliable user identification it is important that not only most, but all users are identified correctly, which is why we also report the minimum accuracy in Table IV. Almost all model combinations can identify all of the 34 subjects in the test takes correctly when majority voting is applied, the only exception being the RF+SR, RNN+BR and MLP+BR models, which misclassify at least one take. However, we’ve noticed that reliable identifications can already be made on shorter sequences than the full length of each test take. Therefore, to see how the models perform on shorter intervals, we re-evaluate the models on shorter sequences from the test takes: We select all sequences of a given length from each test take and determine the subject by majority voting and repeat this process for sequence lengths ranging from the number of frames required for one sample up to 27,000 frames (i.e., 5 minutes). The results are illustrated in Figure 3. To rank the models we compare the required sequence lengths to achieve mean accuracies of 90% and 100%. In general, most RNNs perform well: LSTM+BR, GRU+BR V, LSTM+BR V, FRNN+BR V all reach 100% accuracy within 5 minutes. The LSTM+BR and GRU+BR V require with 2.5 minutes the shortest period to reach 100% accuracy. Interestingly, the FRNN+BR V is not much worse and achieves 100% accuracy in 160 seconds. From the non-deep learning models only the RF+BR model achieves 100% accuracy (after 240 seconds).

V. DISCUSSION

The results confirm that classification with SR, BR and BR V data sampled from arbitrary motion data is possible. However, mere SR data is problematic, since it encodes a lot of straightforward session specific data (i.e., position and orientation of a user within the scene). Models achieve high accuracy scores on the validation data that originate from the same session, but fail if user location within the scene changes. While this is arguably an expectable result, we think this is an issue worth highlighting: if position or orientation of subjects is not guaranteed to match the training data during inference, it becomes inevitable to train models on BR or BR V data. That being said, some SR features, like the positional up-axis, remain interesting, since they can encode important personal characteristics (i.e., body height) in XR settings where body proportions are not normalized.

The machine learning models were able to also identify subjects with BR and BR V data. The highest accuracies on the
test data were reached with the BRV data. The reason may be that models trained on BRV data are less prone to overfitting on predominant body postures of subjects in the training data. Neural networks produced better results compared to random forest models on this dataset. RF and RNNs work comparatively well on SR and BR data. However, RNNs outperformed RF and MLP on the BRV data considerably. We theorize that patterns in BRV data are more difficult to learn and RNNs seem to profit from working directly on the raw sequence data. In general, we believe that BRV data provides a promising basis for robust user identification especially for multi-session scenarios and is an interesting topic for future work. A majority voting approach on sequences with more than one sample improved the reliability of all models. Here, BR and BRV data yield similar results. The deep learning approaches performed generally well and profit from samples that span over a longer period of time. We think that all three RNN architectures are viable models for the identification tasks, differences in the achieved accuracies might be eliminated with a broader hyperparameter search. Random Forest and MLP architectures were all outperformed by at least one RNN architecture, even though it has to be noted that the RF+BR model achieved considerable results.

The goal of this paper is to provide viable insights for future research in the field of user identification with motion data in XR contexts. However, the used dataset originates from a full body motion capture system, and not from typical XR hardware, so are our findings even valid for XR setups? Since we specifically only selected features also available in XR setups and other characteristics are comparable (e.g., frame rate, reference, etc.), we see only two potential objections: First, sensor specific characteristics are different between a motion-capture system and a HTC Vive. However, sensor specific characteristics are also different between an HTC Vive and an Oculus Rift, so in either case models would have to be retrained on data from the respective hardware for optimal performance, and Miller R. et al. [16] have shown that machine learning models can even work with multiple systems at the same time. Second, user specific characteristics could be more or less dominant in an XR scenario where the user wears HMD and controllers than in a full-body motion tracking scenario. We think that this is much more an issue of the target application scenario than of the tracking hardware used: there will be inherently much more movement, and therefore more potentially identifying behaviour, in more active settings where users need to walk around than in more passive settings where users can sit down and may need to interact only occasionally. Altogether, our findings about architecture and data representation are not tied to the hardware it was recorded with and are therefore relevant for future research in the XR context. In fact, the results presented can serve as a baseline for future research, which is an urgent desideratum in this field, since hardly any of the previous work published neither data nor code.

VI. LIMITATIONS AND FUTURE WORK

The used dataset is comparatively large in terms of the overall coverage of movement data and the length of the recordings. However, it only consists of movement data of 34 subjects, ranking it somewhere in the average of the reviewed approaches.

The normalization to a uniform skeleton of the movement data can currently be seen not as a limitation but as a benefit of our solution. Since the models already solve the identification task with high accuracy on this — in terms of the individual body proportions — somewhat degraded dataset, it is safe to assume our solution would at least work as reliable on a dataset including information on body proportions that would originate from an actual XR setup. Chances are, it would even show an improved performance.

Overall, still the main limitations can be identified to be the choice, format, and size (i.e., amount of individual subjects) of the used dataset. The public availability of comprehensive and reliable datasets is a well-known problem in the field of machine learning, to be able to replicate, compare, and ultimately improve possible solutions. To address these issues, future work is needed in this direction. After the successful results on user identification reported here, and some other related research at our lab, we have already started to consolidate the various XR-related recordings of movement data performed in our lab to generate such datasets.

VII. CONCLUSION

This article compared five machine learning architectures on three selections of movement data for user and avatar identification. Results on body-relative and velocity data promise to enable identification of subjects across multiple sessions. After a hyperparameter search LSTM+BR and GRU+BRV outperform competing models and can classify all subjects with an accuracy of 100% within 150 seconds.

Altogether, our approach provides an effective foundation for behaviometric-based identification solutions, based on deep learning of arbitrary motion data sequences in XR contexts. We believe that such solutions will be increasingly important specifically for embodied encounters in social XR, where the look and appearance of avatars becomes more and more indistinguishable from the appearance of real human individuals. Here, risks of identity theft and the misuse of our own avatars arise, which seems specifically sensitive for prominent avatars of influential people or of people of power and authority, e.g., in the political system or in companies. We need reliable mechanisms helping us to decide if we can still trust our virtual encounters. In summary, this work a) provides a solution to find effective models for behaviometric-based user identification and b) motivates efforts to create more expansive biometric datasets to further explore this subject.

REFERENCES

[1] M. E. Latoschik, F. Kern, J.-P. Stauffert, A. Bartl, M. Botsch, and J.-L. Lugrin, "Not Alone Here?! Scalability and User Experience of Embodied Ambient Crowds in Distributed Social Virtual Reality," IEEE
T. Mustafa, R. Matovu, A. Serwadda, and N. Muirhead, "Fast generation of realistic virtual humans," Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST, vol. Part F1319, 2017.

S. Wenninger, J. Achenbach, A. Bartl, M. E. Latoschik, and M. Botsch, "Realistic Virtual Humans from Smartphone Videos," in 26th ACM Symposium on Virtual Reality Software and Technology, (New York, NY, USA), pp. 1–11, ACM, 11 2020.

M. E. Latoschik, D. Roth, D. Gall, J. Achenbach, T. Waltemate, and M. Botsch, "The effect of avatar realism in immersive social virtual realities," in 23rd ACM Symposium on Virtual Reality Software and Technology (VRST), pp. 39:1–39:10, 2017.

S. W. Cook, H. S. Friedman, K. A. Duggan, J. Cui, and V. Popescu, "Hand Gesture and Mathematics Learning: Lessons From an Avatar," Cognitive Science, vol. 41, no. 2, pp. 518–535, 2017.

K. Lilija, S. Kyllingsbaek, and K. Hornbaek, "Correction of avatar hand movements supports learning of a motor skill," Proceedings - 2021 IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2021, pp. 455–462, 2021.

G. Lee, Z. Deng, S. Ma, T. Shiratori, S. Srinivasa, and Y. Sheikh, "Talking With Hands 16.2M: A Large-Scale Dataset of Synchronized Body-Finger Motion and Audio for Conversational Motion Analysis and Synthesis," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 763–772, IEEE, 10 2019.

M. R. Miller, F. Herrera, H. Jun, J. A. Landay, and J. N. Bailenson, "Personal identifiability of user tracking data during observation of 360-degree VR video," Scientific Reports, vol. 10, no. 1, pp. 1–10, 2020.

T. Mustafa, R. Matou, A. Serwadda, and N. Muirhead, "Unsure how to authenticate on your VR headset? Come on, use your head!," IWSPA 2018 - Proceedings of the 4th ACM International Workshop on Security and Privacy Analytics, Co-located with CODASPY 2018, vol. 2018-Janua, pp. 23–30, 2018.

A. Kupin, B. Moeller, Y. Jiang, N. K. Banerjee, and S. Banerjee, "Task-Driven Biometric Authentication of Users in Virtual Reality (VR) Environments," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11295 LNCS, pp. 55–67, 2019.

K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt, "Behavioural Biometrics in VR: Identifying People from Body Motion and Relations in Virtual Reality," Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI ’19, vol. 12, pp. 1–12, 2019.

A. Aji, N. K. Banerjee, and S. Banerjee, "Combining pairwise feature matches from device trajectories for biometric authentication in virtual reality environments," Proceedings - 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2019, pp. 9–16, 2019.

F. Mathis, H. I. Fawaz, and M. Khamis, "Knowledge-driven biometric authentication in virtual reality," Conference on Human Factors in Computing Systems - Proceedings, 2020.

R. Miller, N. K. Banerjee, and S. Banerjee, "Within-System and Cross-System Behavior-Based Biometric Authentication in Virtual Reality," Proceedings - 2020 IEEE Conference on Virtual Reality and 3D User Interfaces, VRW 2020, pp. 311–316, 2020.

R. Miller, N. K. Banerjee, and S. Banerjee, "Using Siamese Neural Networks to Perform Cross-System Behavioral Authentication in Virtual Reality," in 2021 IEEE Virtual Reality and 3D User Interfaces (VR), pp. 140–149, IEEE, 3 2021.

I. Olade, C. Fleming, and H. N. Liang, "BioMove: Biometric user identification from human kinesiological movements for virtual reality systems," Sensors (Switzerland), vol. 20, no. 10, pp. 1–19, 2020.

J. Liebers and S. Schneegass, "Gaze-based Authentication in Virtual Reality," Eye Tracking Research and Applications Symposium (ETRA), pp. 2019–2020, 2020.

A. G. Moore, R. P. McMahan, H. Dong, and N. Ruozzi, "Personal Identifiability and Obfuscation of User Tracking Data From VR Training Sessions," 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 221–228, 2021.

C. E. Rogers, A. W. Witt, A. D. Solomon, and K. K. Venkatasubramanian, "An approach for user identification for head-mounted displays," ISWC 2015 - Proceedings of the 2015 ACM International Symposium on Wearable Computers, pp. 143–146, 2015.