2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Incremental Human Gait Prediction without Catastrophic Forgetting

1st Jonathan Jakob
Technical Faculty
Bielefeld University
Bielefeld, Germany
jjakob@techfak.uni-bielefeld.de

2nd Martina Hasenjäger
Learning and Personalization Group
Honda Research Institute Europe
Offenbach, Germany
martina.hasenjaeger@honda-ri.de

3rd Barbara Hammer
Technical Faculty
Bielefeld University
Bielefeld, Germany
bhammer@techfak.uni-bielefeld.de

***Abstract*—Human gait prediction is an important task in predictive exoskeleton control. However, if static models are used to facilitate this task, two problems arise. First, the models cannot adapt to new environments and terrains during deployment, and second, the models cannot be personalized to any given end user without costly involvement of a human expert. Incremental models can alleviate these shortcomings, but they usually are prone to catastrophic forgetting, which can be dangerous during live deployment. In this work, we introduce an incremental model, that can learn human gait from scratch without outside interference, but does not fall prey to catastrophic forgetting. We test and evaluate our model on a real world gait database and show, that it delivers competitive results with regard to other standard approaches.**

***Index Terms*—incremental, online, catastrophic forgetting, personalization, human gait, exoskeleton**

I. Introduction

Modern lower body exoskeletons are assistive devices that use electric motors to adjust the actuators which support the human end user during everyday gait movements [1]. Since, these motors need real physical time to adjust themselves, they are driven by predictive control algorithms, which rely on internal models, that forecast human gait patterns slightly into the future, thus enabling the control unit to always provide the best possible support at the correct time.

Human Motion Prediction is a research field in which algorithms and models can be sorted into different groups based on a number of criteria. First, one can make a distinction based on the objective of the prediction. This leads to two model classes: The first is used to predict movement trajectories or upcoming actions of other agents in shared environments, which aims to solve problems like effective navigation or collision risk minimization [2]. The second is used to predict upcoming movements of a specific agent in order to improve the support that this agent receives from associated systems [3].

Human Gait Prediction is a sub-field of Human Motion Prediction in which the models can be further divided based on the data they operate on. Here, the main classes are visual data obtained from cameras and motion capture systems, as well as sensor data obtained from IMU or EMG sensors. Models that work with visual data [4]–[6] are usually from the first class of objectives, because they can only work in closed environments that are equipped with motion capture systems or when a robot with visual capabilities observes its immediate surroundings. Models that are supposed to facilitate support functionality on the other hand are much more likely to use sensor data.

Support with sensor models can either be done by classifying the input time series into specific movement patterns and then executing a specific routine per class [7]–[10], or by predicting each upcoming time step through regression and then using this prediction directly as input for a predictive control scheme [11]–[13], or as a combination thereof [14], [15]. What all of these approaches have in common, is that they are *offline* trained models, which are static in nature and can pose a challenge from an operational viewpoint.

In Machine Learning, the term *offline* means, that a model is learned based on a specific set of training data and then applied during deployment without any subsequent changes to its parameters. This can lead to problems in situations where the distribution of the input data changes over time, a phenomenon commonly referred to as concept drift [16], which often has the effect of a reduction in model performance. In the context of predictive control for lower body exoskeletons, this can happen through a lack of personalization of the model to the end user or the appearance of walking patterns in the environment, that were not part of the training data (e.g. steep slopes). In fact, it has recently been shown, that personalized models in human movement classification perform much better than non-personalized ones [17].

Of course, static models can be personalized easily by including data from the specific end user into the training process, but in practice this is less than ideal because it requires human experts to obtain the data and create the model which is a costly and unwieldy approach for a real world deployment. Furthermore, even when these costs are accepted, there still is the problem, that a training set of limited size will most likely never contain all possibilities that can arise in a real world environment (e.g. different floor surfaces like gravel or other uneven terrain), which might lead to subpar predictions in these situations and therefore creates the necessity for other approaches that address these challenges.

A natural solution for these problems can be found in *incremental* or *online* learning [18], which is a paradigm that forms the counterpart to offline models. Incremental models do not have an initial, one time training phase, but rather are continuously updated with each new data instance that they obtain throughout their deployment. This enables such models to be easily and cheaply personalized to specific end users as well as to adapt themselves to unforeseen environmental conditions.

However, the extreme agility of incremental models also has a significant drawback, which is commonly known under the term *catastrophic forgetting* [19]. This means, that continuously adapting models are prone to forget previously learned concepts if they are not represented in the input data for a while, because they are overwritten with new concepts. This is a problem, especially when reoccurring concepts appear in a data stream, because in that case these concepts have to be relearned every time which leads to subpar predictions around the change points of the data stream. In the context of human gait prediction this can even be dangerous because an incremental model could for example forget the concept of going down the stairs which in the worst case might lead to avoidable accidents. Therefore, building incremental models that can safeguard against catastrophic forgetting seems to be a promising research direction for algorithms revolving around predictive exoskeleton control.

In this contribution, we present a Memory Management (MM) approach for an incremental kNN model, that can learn human gait predictions from scratch without human interference but does not fall prey to catastrophic forgetting. We evaluate our method on a large, real world human gait database and show that it outperforms static state of the art approaches as well as vanilla incremental models. Specifically, we show that large improvements in prediction quality can be made in regions of the data stream that are affected by catastrophic forgetting under normal circumstances.

The rest of this paper is structured in the following way: In the next section we briefly define the prediction problem mathematically and after that, Section III introduces our proposed model and explains it in detail. Then, an overview of the data that we use for evaluation is given in Section IV, which is followed by the description of our experiments in Section V. Afterwards, the results are presented in Section VI and then, the paper is completed by a short conclusion in the end.

II. Problem Setting

Throughout this paper, incremental regression is used to predict a data stream one instance after another. Hereby, a data stream $S = \{s_1, s_2, s_3, ..., s_t\}$ is defined as a potentially infinite set of data points $s_i \in \mathbb{R}^n$.

Furthermore, an incremental model is defined as an algorithm that receives a data stream instance after instance and generates a sequence of models $h_1, h_2, h_3, ..., h_t$ where $h_{i\ 1}(s_i) = s_{i+1}$ is a function that acts on the current instance and predicts the value of the subsequent instance of the data stream. After that, the true value s_{i+1} is revealed and a new model h_i is learned.

To evaluate such an incremental regression task, usually the *Interleaved-Train-Test-Error (ITTE)* is applied:

$$E(S) = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(h_{i\ 1}(s_i) - s_{i+1})^2}$$

This ITTE measures the *Root-Mean-Squared-Error (RMSE)* over every model h_i up to a given time point t.

III. Model

In a previous work [20], we compared a wide range of incremental algorithms and evaluated them with regard to their behaviour around sudden change points in data streams and their overall suitability for regression problems in predictive exoskeleton control. It turned out, that a simple kNN model [21] significantly outperforms all other, more sophisticated models, on all evaluation schemes. This is consistent with other research from the classification domain [22], where incremental kNN models also perform very competitively.

However, the problem with incremental kNN models is, that their memories simply accumulate more and more data over time, until the computation time at inference becomes so high that predictions do not arrive in time anymore, which renders the models useless for real world deployment. The common remedy for this problem is to implement the memory as a fixed length sliding window over the last n samples of the data stream. In this way, inference time will always be limited to a specific value that can be tuned to the available hardware through the number of samples n.

Nevertheless, this setup has another drawback, as it gives rise to the potential occurrence of the aforementioned problem of catastrophic forgetting, because any concept in the data, that is not expressed in the input stream for a time period that is longer than the size n of the sliding window, will automatically be forgotten. This is much less than ideal in the context of predictive control, because it is not possible to guarantee a certain minimum in prediction quality whenever long term, periodic change points are contained in the data stream.

A. Basic Goals and Considerations

Therefore, we want to create a memory management approach, that aims at preserving at least a few samples of any concept that has historically been observed in a data stream, in order to be able to guarantee a minimum in prediction quality, regardless of how long a historical concept has not been expressed.

However, we also want to retain the sliding window, because for any predictive task that runs over a time horizon, and is not in a region of sudden concept change, the most useful information is usually found in the direct vicinity in time of the input stream.

Since, there are no labels in incremental learning, we need to resort to clustering, in order to extract meaningful information about the existing concepts in the data stream, which is necessary to determine the specific data samples that are to be preserved in the memory. In the context of human gait prediction, this is not possible on a single data sample basis, because our tests have shown, that no naturally separable clusters exist when all samples of a data stream are examined individually. Therefore, we opt to using time series of discrete physical steps, which are defined as the set of data samples from the onset of the heel strike of one foot, until the onset of the subsequent heel strike of the other foot.

Depending on the number of terrains that an exoskeleton would finally be exposed to during deployment as well as the bodily characteristics of the end user, the number of expected clusters will change on a case by case basis. Therefore, the clustering algorithm needs to determine the number of clusters automatically. From the existing solutions in this realm, our test showed that the Affinity Propagation (AP) [23] algorithm works best.

Standard AP works only on vectorial input samples and not on multi-dimensional time series. Therefore, we compute meaningful pairwise distances between the time series representing the physical steps from the human gait data stream with the Dynamic Time Warping (DTW) [24] algorithm. While it is possible to perform clustering directly on these distances, our tests have shown, that using the distances to learn a low-dimensional embedding of the time series with UMAP [25], and performing clustering on that space, leads to better results in practice. Furthermore, comparing the clusters that are found through this process with ground truth obtained from labeled data sets, confirms that the clusters are in fact meaningful and correspond to different walking patterns in the data stream.

Since, the requirements for timely predictions in online learning still apply, it is important that the memory management can be run independently from prediction, as to ensure that the heavier computations of concept extraction do not interfere with punctual inference.

B. Memory Management Approach

A schematic overview of our Memory Management (MM) approach is visualized in Figure 1. It consists of three distinct paths, that a newly arriving input sample is traversing during its processing.

1) Memory Path: In the path that leads to the preservation of historical concepts, every new input sample is first fed into a step buffer, where all instances of the current physical step accumulate. After the physical step is finished, the time series representing it is given to a reservoir with a fixed maximum size where the time series are stored. If the maximum size of the reservoir is reached, the oldest time series is discarded for every new one that arrives. From there, the path only continues in specific predetermined time intervals. Whenever such an interval is over, all time series from the current memory are added to the reservoir, in order to ensure, that historical concepts that have already been preserved, but are not expressed in the reservoir at the current moment, are not forgotten. Then, pairwise distances of all time series in the reservoir are computed with DTW, a low-dimensional embedding is learned based on these distances with UMAP and clustering is performed on this embedding with AP. Finally, based on a predetermined number, a few time series from each cluster are extracted with a simple strategy that aims to cover as much of the data space as possible. These time series then form the new memory of historical concepts.

2) Sliding Window Path: The path that is responsible to provide accurate information about the local concepts in time is a very simple sliding window with a fixed, predetermined size. Whenever a new input sample arrives, it is added to the sliding window and in return, the oldest sample from that window is discarded.

3) Prediction Path: The final path is the one that provides the predictions of the approach. Here, every incoming sample is forecast based on the kNN rule, where the memory of the model is a simple union of the current contents of the historical memory and the sliding window. Note, that predictions can be made as soon as more than k samples have accumulated in the sliding window. In this regard, the Prediction Path is entirely independent from the Memory Path, which means that they can be run in parallel and the heavier computations of memory extraction can be distributed over the whole time interval in which new memories are computed.

IV. Data

To evaluate our approach, we use data from the 'Multi-Modal Gait Database of Natural Every-Day Walk in an Urban Environment' (MMGD) [26]. While there is a number of other gait databases that are publicly available (see [26] for a comprehensive list), the MMDG is the only one that provides long, continuous data streams that are suitable to simulate a real world setting for human gait prediction in a regression context. The MMDG contains data from 20 people, who completed three different walking courses with different terrain characteristics. In this work, we concentrate on 'CourseB' for evaluation, which is visualized in Fig. 2 and contains eight segments of alternating walk modes from the set {'Level Ground Walk', 'Slope Down', 'Slope Up' and 'Stairs Up'}. All participants completed three rounds of the course, yielding data streams with an average duration of about 10 minutes. While the database provides a wide range of data (eye-tracking, full-body Xsens IMU and foot insoles pressure data), we only utilize data from the six lower body IMU sensors located at the ankles, shins and thighs as well as the data from the pressure insoles, which is used to segment the data stream into physical steps. Note, that this can also be accomplished by tracking the acceleration of the lowest IMU in forward direction, but the pressure data is more accurate. In addition to the evaluation on 'CourseB' we also utilize data

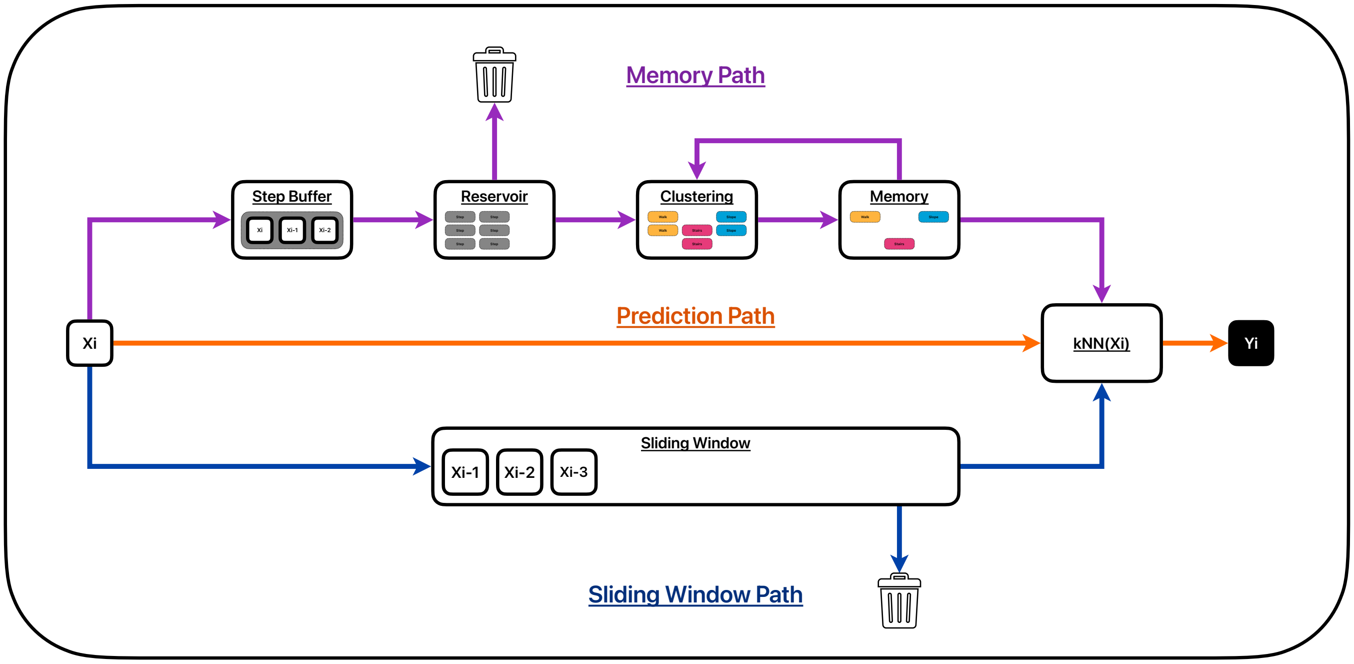

Fig. 1. Overview of the Memory Management (MM) approach. The Memory Path is shown in purple, the Sliding Window Path is shown in blue, and the Prediction Path is shown in orange. Note, that the Prediction Path is completely independent from the Memory Path and both can be run in parallel. For a detailed description see the text in section III.

from 'CourseA' to train the offline models that are part of the experimental setup.

V. Experiments

In our experiments, we investigate two things. First, we want to see, how well state of the art offline approaches perform on the data and then compare them to a vanilla model and our new approach from the online domain. Afterwards, we want to test, how and where improvements in prediction quality can be achieved by comparing our MM approach to the vanilla online model.

A. Models

Judging from the literature [11]–[15], the state of the art in Human Gait Prediction consists of deep neural architectures of various flavours that are trained in an offline manner. As a proxy of all of these approaches, we opt for a Gated Recurrent Unit (GRU) [27]. In a real world scenario it will never happen, that the training data of a predictive control model comes from the same environment as the one observed during later deployment. We simulate this by training the offline models only on 'CourseA' of the database, while evaluation is done on 'CourseB'.

Based on the findings in [20] we settle on the kNN as the best performing representative for online learners, while the MM approach described above is our attempt at improving this vanilla model by solving the problem of catastrophic forgetting.

For evaluation, we create a specific model from each model class for every participant in the MMGD database. All model classes are listed in Table I and are described in the following:

TABLE I
All model classes that are compared in the experiments. Experiments are conducted with one independent model from each model class for every participant in the MMGD database. Offline models are trained on the data sets from 'CourseA'. Personalized means that the training set for the Offline model of any participant included data from that participant.

Model Classes used in the Experiments		
Name	**Type**	**Description**
GRU-NP	Offline	Non-Personalized Gated Recurrent Unit
GRU-P	Offline	Personalized Gated Recurrent Unit
kNN	Online	k-Nearest-Neighbours Model
MM	Online	Memory Management Approach

1) GRU-NP: The Non-Personalized version (NP) of the GRU is trained on data from 'CourseA' of all participants, but excluding the person that the current model is created for. The models have a hidden state of 128 units and are fed time series of 60 data points as input. Hereby, the size of the input series was chosen to be roughly equivalent to two physical steps (one left and one right) in the data sets, which is the maximum number that has any meaning for the forecasting problem, since a different walk pattern can always appear in the environment.

2) GRU-P: The Personalized version (P) of the GRU is trained on data from 'CourseA' of all participants, including the person that the current model is created for. Otherwise,

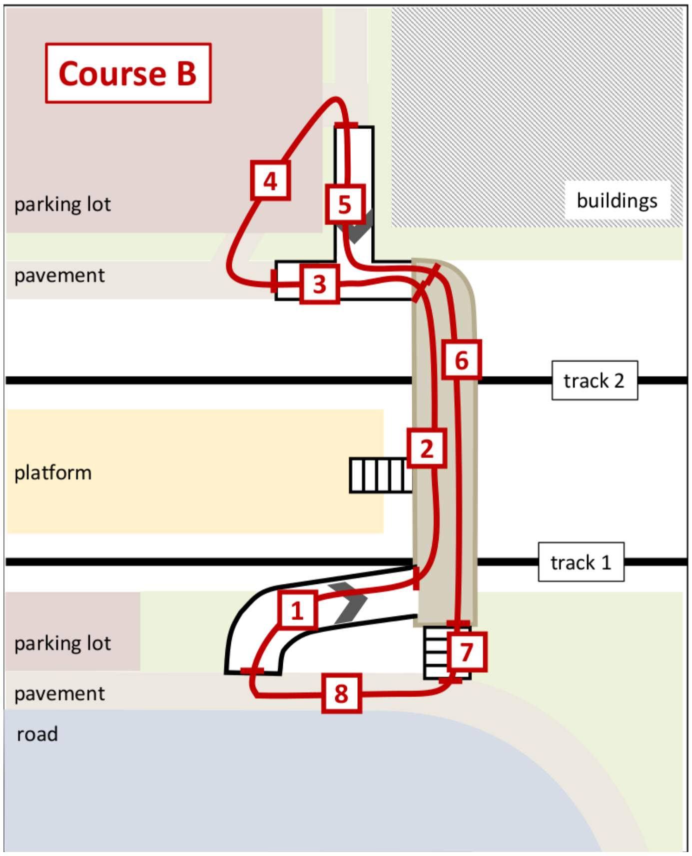

Fig. 2. Outline of 'CourseB' from the Multi-Modal Gait Database. The course starts with a 'Slope Down' segment (1), which is followed by 'Walk' (2), 'Slope Up' (3), 'Walk' (4), 'Slope Down' (5), 'Walk' (6), 'Stairs Up' (7) and finishes with a final 'Walk' (8) segment. The data stream from CourseB for each participant contains three consecutive rounds of the course. Image taken from [26].

the models are identical to GRU-NP.

3) kNN: The vanilla kNN model only sees data from 'CourseB' of the person that it is created for. The memory size is 5000 time steps and was chosen somewhat arbitrarily as the maximum size that would still allow computations to be fast enough, so that the whole data stream of a person could be processed in the time the person needed to complete the course, on the machine that was used during the experiments.

4) MM: The MM approach also only sees data from 'CourseB' of the person that is it created for. To make the comparison with the vanilla kNN fair, the total memory size is also capped at 5000 but divided into a 4000 time step sliding window and a 1000 time step historical memory. The size of these models also allowed for fast enough computations to handle all participants in the necessary time frame.

B. Online vs Offline

The first experiment evaluates the GRU-NP, GRU-P, kNN and MM models on all participants using the data from 'CourseB'. The aim is to find out, how personalization affects the average-person models of the offline domain, how online and offline models stack up against each other, and what kind of impact training in a different environment than deploying can have on performance.

Evaluation is performed by comparing the RMSE of each model over the whole data set from 'CourseB' for every participant.

C. kNN vs MM

In the second experiment, we want to examine, whether our MM approach can outperform the vanilla kNN and if the problem of catastrophic forgetting can be avoided.

Here, evaluation is performed in a more precise way. Since, the MM approach is designed to only be beneficial in those few regions of the data stream, where catastrophic forgetting would be an issue, we specifically evaluate the performance of both models in the two regions where catastrophic forgetting is most prevalent. In the 'CourseB' data streams, these regions are the change points from Segment 6 to Segment 7 (see Figure 2) of the second and third iteration of the course. Segment 7 is the only segment of the course, where participants walked up some stairs. Since, a complete traversal of the course is much longer than the memory sizes of our online models, there will not be any data from Segment 7 of the previous round be left in the sliding windows, whenever this segment is encountered again. Thus, this change point marks the best spot to evaluate whether our MM approach can really deliver on what it was designed to do. To measure the performance we take the RMSE over the first two (one left and one right) and the first six (three left and three right) steps of Segment 7 in the second and third round of the course.

VI. Results

The outcomes of experiments one and two are reported in Table II and Table III respectively, as well as visualized in Figures 3, 4 and 5. A detailed description and interpretation of the results is given below.

A. Online vs Offline

The results in Table II show, that the online models outperform the offline domain by a definite margin. The same information is visualized in Figures 3 and 4 (since the kNN and MM perform so similar, we plotted them separately with a different scale).

On the online side, the MM model beats the vanilla kNN on every participant, but as expected, only with a very small margin. This is due to the fact, that the internal adjustments of the MM approach are only beneficial in those regions of the data set that revolve around change points, and these regions constitute only a small portion of the stream as a whole.

In the offline domain, some clear differences between Personalized and Non-Personalized models can be observed, with Personalization taking the clear lead on every participant. This is in line with the findings in [17] where the same behaviour is observed in the context of Human Movement Classification.

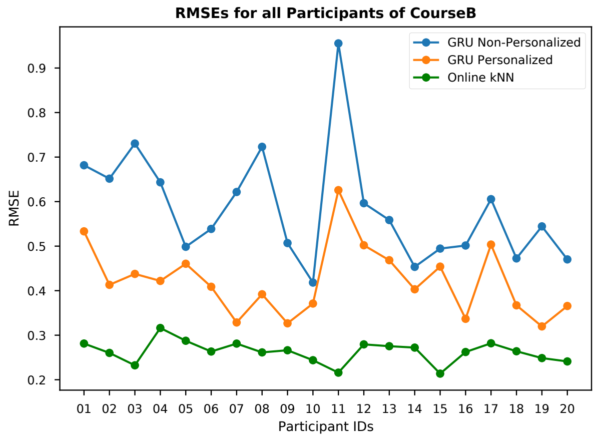

Fig. 3. Root-Mean-Square-Errors (RMSE) of two offline and one online model class for all participants on 'CourseB'. The Non-Personalized Offline models were trained on the data from 'CourseA' of all participants, excluding the one being evaluated, and are shown in blue. The Personalized Offline models were also trained on the data from 'CourseA' of all participants, but including the one being evaluated, and are shown in orange. Lastly, the Online models only saw the data of each participant being evaluated and are shown in green.

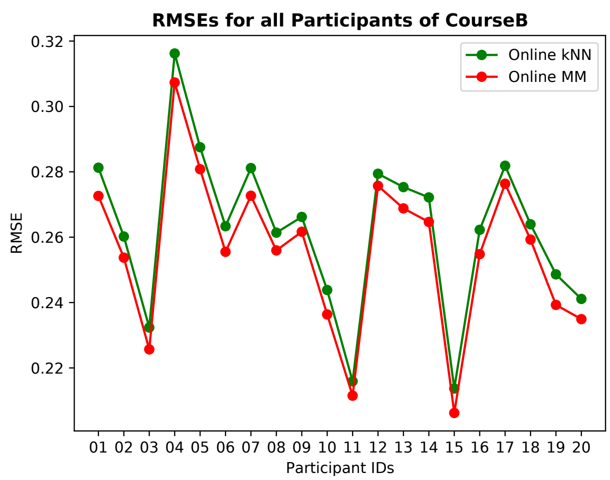

Fig. 4. Root-Mean-Square-Errors (RMSE) of an Online kNN model and the Online Memory Management (MM) approach. The Online kNN is shown in blue and the Online Memory Managemnt is shown in orange.

Another thing that strikes out, is the much higher variance that the offline models exhibit in comparison with the online ones. This finding is most pronounced for the Non-Personalized models which could indicate that average-person models can run into trouble, if the characteristics of the person for whom they are deployed differs significantly from people that were part of the training set (e.g. in height, weight or specific bodily impairments). However, since the Personalized models also exhibit this phenomenon (albeit to a lesser degree), the fact that offline models can never be trained in the exact same environments that they are deployed in, might also play a role.

TABLE II

ROOT-MEAN-SQUARE-ERROR (RMSE) VALUES OF TWO OFFLINE AND TWO ONLINE MODELS FOR ALL PARTICIPANTS OF THE MMGD DATABASE. EVALUATION IS PERFORMED ON THE FULL DATA SETS FROM 'COURSEB' (SEE FIG. 2). THE OFFLINE MODELS COMPRISE A PERSONALIZED (GRU-P) AND A NON-PERSONALIZED (GRU-NP) VERSION AND ARE TRAINED ON THE FULL DATA SETS FROM 'COURSEA'. ALL VALUES ARE AVERAGED OVER 10 INDEPENDENT RUNS. THE LAST TWO ROWS SHOW THE MEAN AND VARIANCE OVER ALL PARTICIPANTS.

	RMSEs for all Models on 'CourseB'			
	Offline Models		**Online Models**	
Persons	**GRU-NP**	**GRU-P**	**kNN**	**MM**
1	0.681	0.533	0.281	**0.272**
2	0.651	0.413	0.260	**0.253**
3	0.730	0.437	0.232	**0.225**
4	0.643	0.421	0.316	**0.307**
5	0.498	0.460	0.287	**0.280**
6	0.538	0.408	0.263	**0.255**
7	0.621	0.328	0.281	**0.272**
8	0.723	0.391	0.261	**0.255**
9	0.506	0.326	0.266	**0.261**
10	0.418	0.371	0.243	**0.236**
11	0.955	0.625	0.215	**0.211**
12	0.596	0.502	0.279	**0.275**
13	0.558	0.468	0.275	**0.268**
14	0.453	0.403	0.272	**0.264**
15	0.494	0.454	0.213	**0.206**
16	0.501	0.336	0.262	**0.254**
17	0.605	0.503	0.281	**0.276**
18	0.472	0.367	0.263	**0.259**
19	0.544	0.319	0.248	**0.239**
20	0.470	0.365	0.241	**0.234**
Mean	0.583	0.421	0.262	**0.255**
Var	0.0150	0.0057	0.00058	**0.00056**

B. kNN vs MM

Table III contains the outcome of the second experiment. These results show, that the MM approach delivers significantly better results than the vanilla kNN in those regions of the data streams, which are most susceptible to catastrophic forgetting. More detailed, the MM model exhibits an average decrease in RMSE of 24.25% on the first two steps of Segment 7 compared to the kNN. After the first six steps, the average decrease is still 18.68%. This number continues to go down, because with each new step taken, there is more new information about the current concept being added to the sliding windows.

The higher performance of MM in this region of the data stream derives from its refined internal memory structure, of which a snapshot from the time point directly before the start of the second iteration of Segment 7 is visualized in Figure 5. The plots show the content of both, the vanilla kNN sliding window, and the union of the MM sliding window and the historical memory. Since, the kNN memory only contains data from Segments 6 ('walk') and 5 ('slope down'), which directly precede the current time step, it can not adequately react to the upcoming Segment 7 ('stairs up').

The bulk of the data in the MM memory is also from Segments 6 ('walk') and 5 ('slope down'). However, in addition to that, the approach has managed to preserve a small amount of data, representing the concepts 'stairs up' and 'slope up',

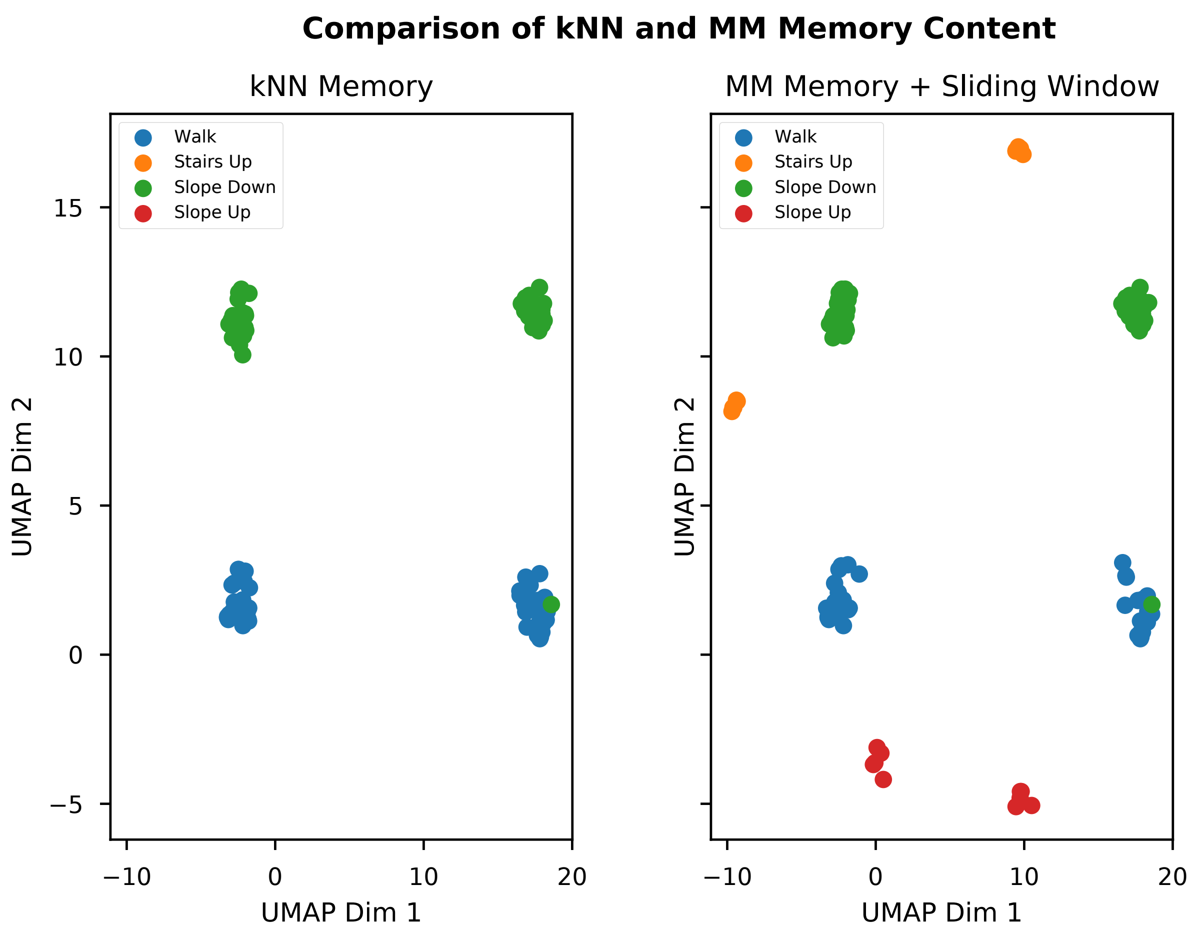

Fig. 5. Content comparison of the kNN and MM memories directly before the second iteration of Segment 7 in 'CourseB' (see Figure 2) for Person 1. Both plots contain the same number of points. Each point is a 2-dimensional UMAP embedding of a time series representing a full physical step. Points are colored according to the movement patterns they belong to (The green point in the blue cluster is most likely labeled wrongly). Note, that every movement pattern has two clusters, one for each foot. The plot on the left shows the memory of the kNN sliding window. It only contains points from the segments 6 ('walk') and 5 ('slope down'), which come immediately before the snapshot. The plot on the right shows the union of the MM sliding window and the historical memory. It holds less points from Segment 5 because the sliding window is shorter, but the historical memory adds points from other previous segments, so that data from the patterns 'slope up' and 'stairs up' is also available for prediction.

from the first iteration of the data stream, thus enabling it to react to the upcoming Segment 7 ('stairs up') in a much better way.

VII. Conclusion

In this paper, we used the recently published MMGD Human Gait Database, to investigate the problem of Human Gait Prediction in the context of predictive exoskeleton control. In particular, we examined how the state of the art, consisting of neural architectures that are trained in a offline way, holds up against more agile models from the online learning domain.

We saw, that within the offline realm, Personalization to the end user plays a large factor with regard to prediction quality, a finding that is in line with previous research on Human Motion Classification [17].

Furthermore, we found that overall, the simple incremental kNN models performed much better than the much more sophisticated GRUs from the static model classes. We attribute this behaviour at least in part to the circumstance that the offline models were trained on data from a slightly different environment, which is in line with most real world deployment scenarios.

Since, incremental approaches were shown to be beneficial to the problem at hand, we set out to mediate their most problematic drawback, by proposing a Memory Management (MM) model for an incremental kNN regressor, that parlays a dual-memory architecture into a remedy for the catastrophic forgetting problem. In a second round of experiments we showed, that this architecture is in fact able to improve the predictive performance in the regions of the data streams that are most susceptible to catastrophic forgetting by a significant amount.

Overall, based on the findings in this work, we think that incremental approaches should be considered for at least some applications of Human Gait Predictions. However, when ex-

TABLE III
ROOT-MEAN-SQUARE-ERROR (RMSE) VALUES OF AN INCREMENTAL kNN AND OUR MEMORY MANAGEMENT (MM) APPROACH FOR ALL PARTICIPANTS OF THE MMGD DATABASE. EVALUATION IS PERFORMED IN TWO WAYS: ON THE FIRST 2 (1 LEFT AND 1 RIGHT) AND THE FIRST 6 (3 LEFT AND 3 RIGHT) STEPS OF SEGMENT 7 IN 'COURSEB' (SEE FIG. 2).THE THIRD COLUMN OF EACH EVALUATION SHOWS THE DECREASE OF THE RMSE FROM kNN TO MM IN PERCENT (%). ALL VALUES ARE AVERAGED OVER 10 INDEPENDENT RUNS AND OVER THE SECOND AND THIRD REPETITION OF SEGMENT 7 IN THE DATA SETS. THE LAST ROW SHOWS THE MEAN VALUES OVER ALL PARTICIPANTS.

	RMSEs for Beginning of 'CourseB' Segment 7					
	First 2 Steps			**First 6 Steps**		
Persons	**kNN**	**MM**	**%**	**kNN**	**MM**	**%**
1	0.555	0.462	**16.78**	0.501	0.394	**21.37**
2	0.447	0.296	**33.77**	0.378	0.285	**24.45**
3	0.338	0.198	**41.40**	0.333	0.234	**29.65**
4	0.459	0.320	**30.28**	0.354	0.269	**23.93**
5	0.593	0.485	**18.02**	0.481	0.403	**16.29**
6	0.590	0.465	**21.18**	0.462	0.365	**21.10**
7	0.384	0.272	**29.06**	0.320	0.266	**16.86**
8	0.384	0.292	**24.07**	0.307	0.269	**12.24**
9	0.370	0.334	**9.751**	0.336	0.307	**8.627**
10	0.461	0.354	**23.12**	0.430	0.354	**17.76**
11	0.404	0.341	**15.41**	0.355	0.312	**12.29**
12	0.474	0.373	**21.34**	0.407	0.323	**20.69**
13	0.499	0.339	**32.05**	0.425	0.346	**18.65**
14	0.401	0.308	**23.28**	0.327	0.282	**13.99**
15	0.444	0.345	**22.41**	0.442	0.362	**18.16**
16	0.559	0.457	**18.23**	0.545	0.472	**13.43**
17	0.563	0.416	**26.04**	0.421	0.328	**22.06**
18	0.352	0.294	**16.57**	0.287	0.255	**11.12**
19	0.557	0.354	**36.41**	0.446	0.317	**29.01**
20	0.505	0.375	**25.68**	0.441	0.345	**21.93**
Mean	0.467	0.354	**24.25**	0.400	0.324	**18.68**

oskeletons are supposed to be utilized in certain rehabilitative scenarios for example, a purely incremental approach, that learns entirely from scratch without any prior knowledge, is also not a feasible proposition. Therefore, some effort could be directed into investigating how online and offline models can be made to work together, in order to build models that are versatile in a wide range of real world environments. The investigation of these topics remains the subject of future work.

REFERENCES

[1] D. Pamungkas, W. Caesarendra, H. Soebakti, R. Analia, and S. Susanto, "Overview: Types of lower limb exoskeletons," *Electronics*, vol. 8, p. 1283, 11 2019.

[2] R. Andrey, P. Luigi, H. Michael, K. Kris M, G. Dariu M, and A. Kai O, "Human motion trajectory prediction: a survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[3] K. Lyu, H. Chen, Z. Liu, B. Zhang, and R. Wang, "3d human motion prediction: A survey," 2022.

[4] M. S. Yasar and T. Iqbal, "Improving human motion prediction through continual learning," 2021.

[5] Y. Wang, Z. Li, Y. Chen, W. Liao, and A. Wang, "Human gait prediction for lower limb rehabilitation exoskeleton using gated recurrent units," pp. 128–135, 2021.

[6] L. Gril, P. Wedenig, C. Torkar, and U. Kleb, "A tensor-based regression approach for human motion prediction," *Quality and Reliability Engineering International*, vol. 39, no. 2, pp. 481–499, 2023.

[7] N. Thakur and C. Han, "Exoskeleton-based multimodal action and movement recognition: Identifying and developing the optimal boosted learning approach," *Advances in Artificial Intelligence and Machine Learning*, vol. 01, pp. 48–65, 01 2021.

[8] C.-y. Zhao, X.-g. Zhang, and Q. Guo, "The application of machine-learning on lower limb motion analysis in human exoskeleton system," vol. 7621, 10 2012, pp. 600–611.

[9] A. Nagaswathi, A. Turlapaty, and B. Gokaraju, "A machine learning system for classification of emg signals to assist exoskeleton performance," 10 2018, pp. 1–4.

[10] C. Wang, X. Wu, Y. Ma, G. Wu, and Y. Luo, "A flexible lower extremity exoskeleton robot with deep locomotion mode identification," *Complexity*, vol. 2018, pp. 1–9, 2018.

[11] B. Ren, Z. Zhang, C. Zhang, and S. Chen, "Motion trajectories prediction of lower limb exoskeleton based on long short-term memory (lstm) networks," *Actuators*, vol. 11, no. 3, 2022.

[12] X. Jin, J. Guo, Z. Li, and R. Wang, "Motion prediction of human wearing powered exoskeleton," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–8, 12 2020.

[13] J.-L. Ren, Y.-H. Chien, E.-Y. Chia, L.-C. Fu, and J.-S. Lai, "Deep learning based motion prediction for exoskeleton robot control in upper limb rehabilitation," 05 2019, pp. 5076–5082.

[14] B. Fang, Q. Zhou, F. Sun, J. Shan, M. Wang, C. Xiang, and Q. Zhang, "Gait neural network for human-exoskeleton interaction," *Frontiers in Neurorobotics*, vol. 14, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnbot.2020.00058

[15] T. Qin, Y. Yang, B. Wen, Z. Chen, B. Zhong, H. Dong, K. Dou, and C. Yang, "Research on human gait prediction and recognition algorithm of lower limb-assisted exoskeleton robot," *Intelligent Service Robotics*, vol. 14, 2021.

[16] I. Zliobaite, M. Pechenizkiy, and J. Gama, *An Overview of Concept Drift Applications*, ser. Studies in Big Data. Germany: Springer, 2016, vol. 16, pp. 91–114.

[17] V. Losing, M. Hasenjäger, and T. Yoshikawa, "Personalized online learning with pseudo-ground truth," 01 2021.

[18] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal, and J. Gama, "Machine learning for streaming data: state of the art, challenges, and opportunities," *ACM SIGKDD Explorations Newsletter*, vol. 21, pp. 6–22, 11 2019.

[19] P. Kaushik, A. Gain, A. Kortylewski, and A. Yuille, "Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping," 2021.

[20] J. Jakob, M. Hasenjäger, and B. Hammer, "On the suitability of incremental learning for regression tasks in exoskeleton control," 12 2021, pp. 1–8.

[21] E. Fix and J. L. Hodges, "Discriminatory analysis - nonparametric discrimination: Consistency properties," *International Statistical Review*, vol. 57, p. 238, 1989.

[22] V. Losing, B. Hammer, and H. Wersing, "Knn classifier with self adjusting memory for heterogeneous concept drift," 12 2016.

[23] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972 – 976, 2007.

[24] P. Senin, "Dynamic time warping algorithm review," 01 2009.

[25] L. McInnes and J. Healy, "Umap: Uniform manifold approximation and projection for dimension reduction," *ArXiv*, vol. abs/1802.03426, 2018.

[26] V. Losing and M. Hasenjäger, "A multi-modal gait database of natural everyday-walk in an urban environment," *Scientific Data*, vol. 9, 08 2022.

[27] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.