Combining Variational Autoencoder and Recurrent Neural Networks for Generic Human Motion Prediction

Jonas Hansert

Karlsruhe University of Applied Sciences
jonas.hansert@h-ka.de

Abstract. Real-time motion prediction in a three dimensional environment is required for many application from autonomous cars to human robot collaboration to free-fall sorting machines. The most widely distributed sensors for the detection of three-dimensional environments like time of flight cameras, lidar sensors, stereo cameras or radar devices delivers point clouds or other formats that can easily converted to point clouds. The high dimensionality of point clouds and even voxel grids is a major challenge for real-time motion prediction. Most approaches use a skeleton tracking algorithm for dimensionality reduction, which itself is very error-prone. We investigated an approach consisting of a combination of two separately trained neural networks. We used a variational autoencoder for dimension reduction combined with a long short-term memory or a gated recurrent units network for time series prediction in latent space. We were able to show that it is possible to make reliable motion predictions up to one second into the future, depending on the motion.

Keywords: Human Motion Prediction, Recurrent Neural Network, Variational Autoencoder, Machine Learning, Time Series Prediction, 3d Computer Vision

1 Introduction

Predicting human motion is crucial for human-machine interaction. Humans are adept at predicting human motion, allowing us to collaborate with others. Navigate through crowds without causing accidents, solving tasks together, or to defend ourselves against attacking players in sports games are some examples. But even simple tasks such as shaking hands require a reliable prediction of human movement.

For intelligent machines, predicting human motion is important for avoiding collisions and enabling successful human-robot collaboration. It has already been applied to autonomous driving, intelligent robots and human-robot collaboration. Most 3D human motion prediction methods are based on skeletons [1–4]. Skeletons have the advantage that their dimensions are small, for example 25 joint positions in the widely used Microsoft Kinect V2 skeleton tracker [5]. However, estimating the position of skeletal joints is itself a challenging task with a non-negligible error. This error becomes much larger when there are objects close to the person or when some parts are in shadow of other objects.

In our approach, we work with voxel grids and reduce their dimension with a Variational Autoencoder (VAE). In the low-dimensional latent space, we use time series prediction models, which have been used with good results in many other applications [6–8]. This has several advantages over skeleton tracking. We have information about the environment that can influence the human's motion, and we can also predict changes in that environment.

2 Hardware and software setup

For data collection, we used an assembly station with the task of building a Fischertechnik model. The step-by-step instructions were projected on the worktop and the participants could move a wooden cube to see the next step. A detailed description of the setup and the task can be found in [9]. The data is captured using a RealSense D435 depth camera mounted centrally at a height of 2.5 meters at the rear end of the worktop. In figure 1 the assembly workplace used in this work is shown.

Fig. 1: The assembly workplace used for data collection.

The complete workflow of our 3d human motion prediction algorithm is illustrated in figure 2.

The depth camera (A) is mounted centrally at a height of 2.5 meter at the rear end of the worktop, so that the worker is imaged from the front-up.

In the next step we use the GPU-Voxels library [10] (B) to convert the point cloud to a voxel grid of size 64x64x64 voxels with the origin at the left rear corner of the worktop and with the worktop as a ground plane. In a calibration phase, we record the empty workspace and mark all the detected voxels as background and remove them from the voxel grid. To speed up the following calculations we unite a cube of 8 voxels to a single voxel if a minimum of 4 voxels are occupied in this cube. In this representation each voxel has a length of 4cm and we observe a quadratic space of 1,28 meter length in each axis. Our tests showed, that this is large enough for our use case.

Next, the voxel grid is used as input for the 3d convolutional VAE (C) where the dimension is reduced to a vector of size 32. The architecture of our VAE and the reason for the latent vector size are discussed in section 4.

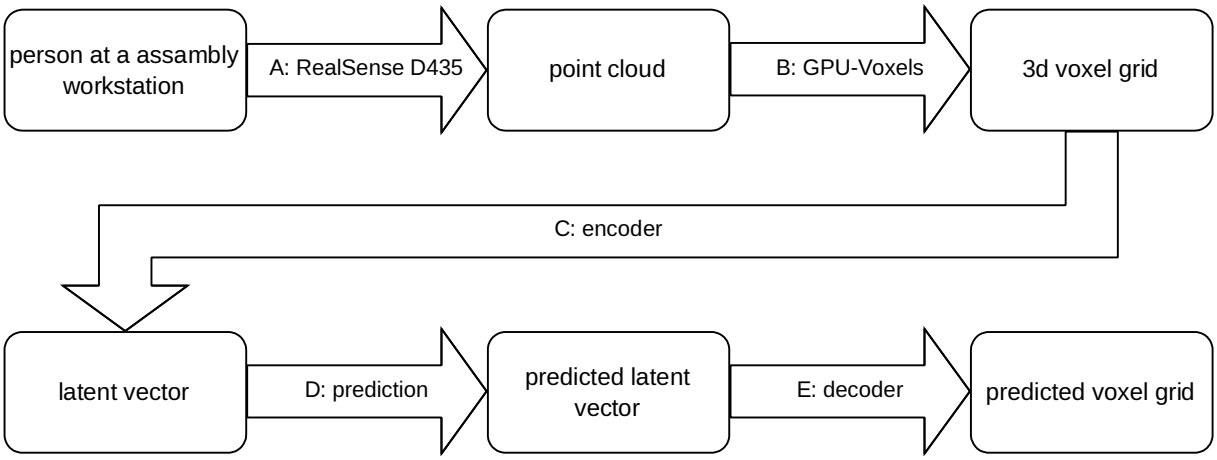

Fig. 2: The pipeline of our system for motion prediction on assembly workstations

We use this latent vector to predict the latent vector of the next time step. We use different models, like Multi-Layer Perceptron (MLP), Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN), in this step (D). We train them and evaluate their accuracy. They are described in section 5.

To get the predicted voxel grid, we decode the predicted latent vector with the decoder of our VAE (E). If we want to predict more than one step, we have to calculate this for every step.

For data collection, training and evaluation, we used a workstation with a 16-core Intel I7 CPU, 64GB RAM and a NVIDIA GeForce RTX2080 Ti GPU.

3 Datasets

Because motion prediction does not require any labels in the dataset every dataset containing time series of point clouds could be used. Nevertheless the most of the widely used data sets for 3d human motion prediction only contain skeleton points or angles but not the pointcloud data or depth data we need for our algorithm. In this section we will introduce some datasets for motion prediction.

Human3.6M [13] is a large dataset with 3.6 million human poses and 17 scenarios. For motion prediction only the scenarios are from interest. The dataset contains 3d joint positions and angels and time-of-flight range data and some additional data not relevant for human motion prediction. This is the most widely used dataset for human motion prediction. Unfortunately, our account was not activated, so we did not get access to this data set.

The CMU Panoptic Dataset [14] contains, aside to a large range of multi-person scenarios, just 23 short recordings of single-person motions and each of them in different scenarios. This makes it hardly suitable to train or evaluate our neural networks.

3DPW [15] is another large dataset containing many scenes in the wild. It contains 60 video sequences and more than 51000 indoor and outdoor poses. Because it do not contain point clouds or depth data it is not suitable for our purpose.

G3D [16] is a large dataset of actions in computer games. It contains skeleton data as well as depth data. Each of the 24 action is recorded 30 times with 10 different actors. This dataset is suitable for our application but not used for the evaluation of any other

motion prediction tasks. This led to our decision, that there is no advantage in using the G3D dataset compared to our own dataset.

AMASS [17] dataset is a large dataset widely used for 3d motion prediction tasks. Unfortunately this dataset only contains skeleton and surface shape data. Caused by the missing point cloud data we can not use this dataset for our application.

The NTU RGB+D [18] Dataset contains 120 action recognition classes, most of them are single person daily actions. It contains 114,480 video samples of an average length of 1.9 seconds. If we want to train our network to predict a time step 1 second in future, this means, that we only have 0.9 seconds training data with no historical data we usually need for prediction with Long Short-Term Memory (LSTM) oder Gated Recurrent Unit (GRU) networks. This makes the NTU RGB+D Dataset not suitable for our application.

WBHM [19] dataset contains a total length of 7.68 hours. The long sequences would be perfect for human motion prediction. However, the dataset unfortunately contains neither point cloud nor depth data.

Because there is no dataset which fits well to our requirements, we collected two own datasets. In the first dataset we collected many different human poses in order to train the VAE with this data. Because balanced training data leads to better results [20], it is important that the number of frames for each human pose is uniform distributed. We recorded two persons for 15 minutes, grasping one part after the other, without building the model.

The second dataset contains the recordings of the building process of the Fischertechnik models. It combines the model building parts with short sequences of picking up new parts from the pick-by-light boxes. We recorded eight test subjects with durations between 16 and 48 minutes. In total we recorded 263 minutes.

4 Variational autoencoder architecture

The VAE is line symmetric. Due to better local dependency and fewer weights, we decided to use 3d convolution layers instead of fully connected layers in a vanilla VAE. We followed the architecture used by Brock et al. [11] but made some changes. We changed the input layer to 50x50x50, the size of our voxel grid and adapted the size of the other convolution layers to 16x16x16, 16x16x16 and 8x8x8 in the encoder part of our model. The decoder is designed symmetrical to the encoder. We added a dropout layer after each convolution layer for better generalisation changed the number of latent neurons to the desired dimension of the latent space.

To find the best dimension of the latent space, we trained the 3d convolutional VAE with latent dimensions of 16, 32 and 64. A latent dimension of 32 gave the best results with an accuracy of 0.974.

5 Time series prediction

We decided to train and evaluate the time series prediction models separated from the 3d convolutional VAE because the training is faster and because the the training data recorded for the prediction is less suited for training the VAE. Therefore, we give our hole training data set to the VAE and use the sequences of latent vectors for the training of our prediction models. We tested several different architectures and some of them with different hyper-parameters.

Multi-Layer Perceptron The architecture of MLP is poorly suited for time series prediction. Nevertheless, we did not want to ignore them because of their simple structure and the completeness of our comparison. We decided to use the past 10 time steps as input, which corresponds to 320 neurons in the input layer. We evaluated two different architectures. The first consists of one fully connected hidden layer with 500 neurons (Fig. 3 left) and the second of two fully connected hidden layers with 500 and 320 neurons . We tested both variants for the prediction of the next time step and the prediction of the next ten time steps (Fig. 3 right).

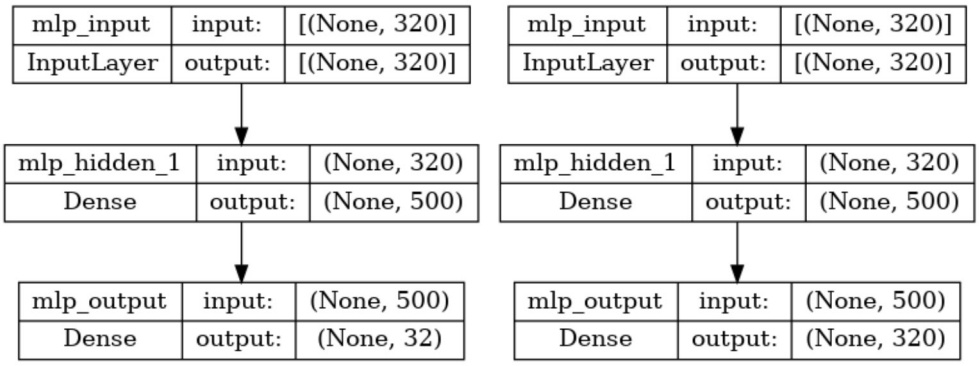

Fig. 3: MLP with 1 hidden layer for the prediction of one time step (left) and MLP with one hidden layer for the prediction of 10 time steps (right)

The mean squared error of the single layer network is with mse = 0.03 better than the two-layer network (mse = 0.04) for the prediction of one step and they are equal for the prediction of ten steps with mse = 0.07.

Convolutional Neural Network CNNs are well suited for time series prediction. With convolution over time, they are good at learning temporal relationships due to their structure. In our architecture a single 1d convolutional layer with 64 kernels of size 4. It is followed by a max pooling layer with kernel size 4 and after a flatten layer a fully connected layer (Fig. 4). For the one step prediction the output layer consists of 32 neurons and for the prediction of 10 time steps of 320 neurons.

Single Layer Long Short-Term Memory Network LSTM has been the first type of RNN to be used successfully in a wide range of applications, because they include mechanisms to avoid vanishing and exploding gradients. Our LSTM network consists of a single LSTM layer with the input and output layers of our MLP network (Fig. 5). For multi step prediction we designed the LSTM like the MLP with a larger output layer for 10 time steps.

The results of our single layer LSTM is better than the MLP but with mse = 0.025 in the case of one step prediction and mse = 0.065 for predicting the next ten steps it is still not very reliable.

Single Layer Gated Recurrent Unit Network GRUs are faster in the training process. and in the application often with better results, compared to LSTM. We decided to train a single layer GRU network and to compare it to the single layer LSTM. The architecture is similar to the architecture of the LSTM network.

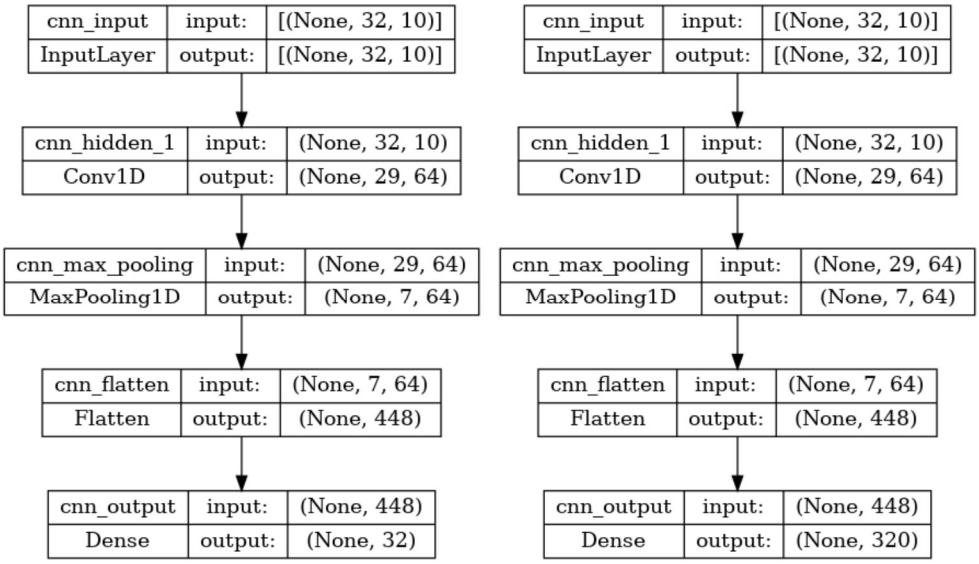

Fig. 4: CNN for single step prediction (left) and CNN for multi step prediction (right)

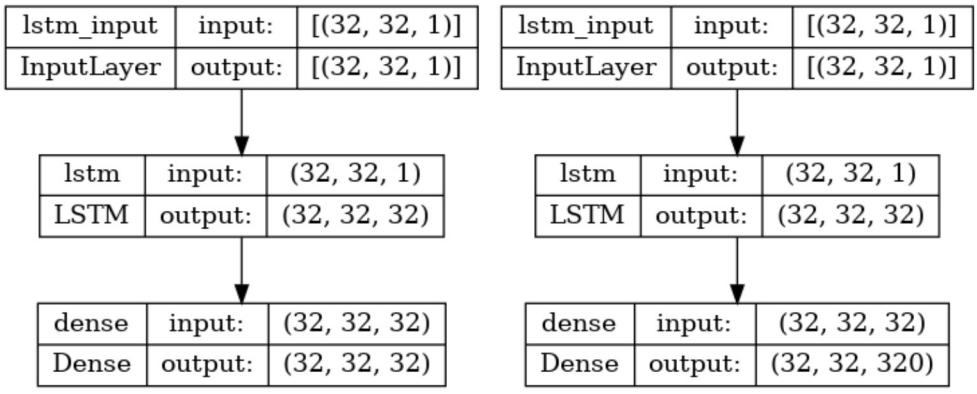

Fig. 5: LSTM with 1 hidden layer (left) and LSTM for multi step prediction (right)

The single layer GRU network has mse = 0.024 for single step prediction and mse = 0.063 for multi step prediction. This is very similar to the single layer LSTM network. Because it is faster then the LSTM network in both, training and run time, we decided to use this network for our following tests.

Multi-Layer Gated Recurrent Unit Network Due to the best performance of the single layer GRU network, we tried to give more complexity to the model by adding a second GRU layer (Fig. 6). This architecture outperformed all other tested architectures with mse = 0.0235 for single step prediction and mse = 0.060 for 10-step prediction.

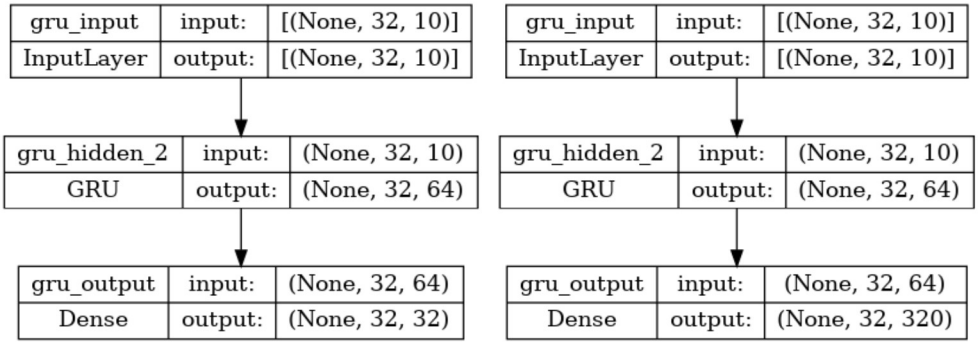

Fig. 6: GRU network with two GRU layers.

6 Evaluation

In this chapter we take the best performing network architecture, the GRU network with two layers of 32 and 64 cells and evaluate it in different settings. We use the recording of one person for the evaluation. This net has a mse = 0.024 on the test data which is the same we had on the validation data (Fig. 7). For the prediction of 10 time steps in future it performs worse with mse = 0.075, but still better than all the other architectures we tested.

To have an idea how good the network architecture performs for the prediction of longer terms, we trained it for the prediction of one time step, 0.33, 0.66 and 1 second in future. The results are shown in Table 1.

predicted time	mean squared error
1 step (0.03 second)	0.024
10 step (0.33 second)	0.06
20 step (0.66 second)	0.09
30 step (1.00 second)	0.11

Table 1: Mean squared error for different prediction periods

Until now, we only looked at the predicted time series of latent vectors. Due to the properties of the VAE, this correlates to the accuracy of the predicted voxel grids. However, this needs to not be a linear relationship. For our hole algorithm, we have an

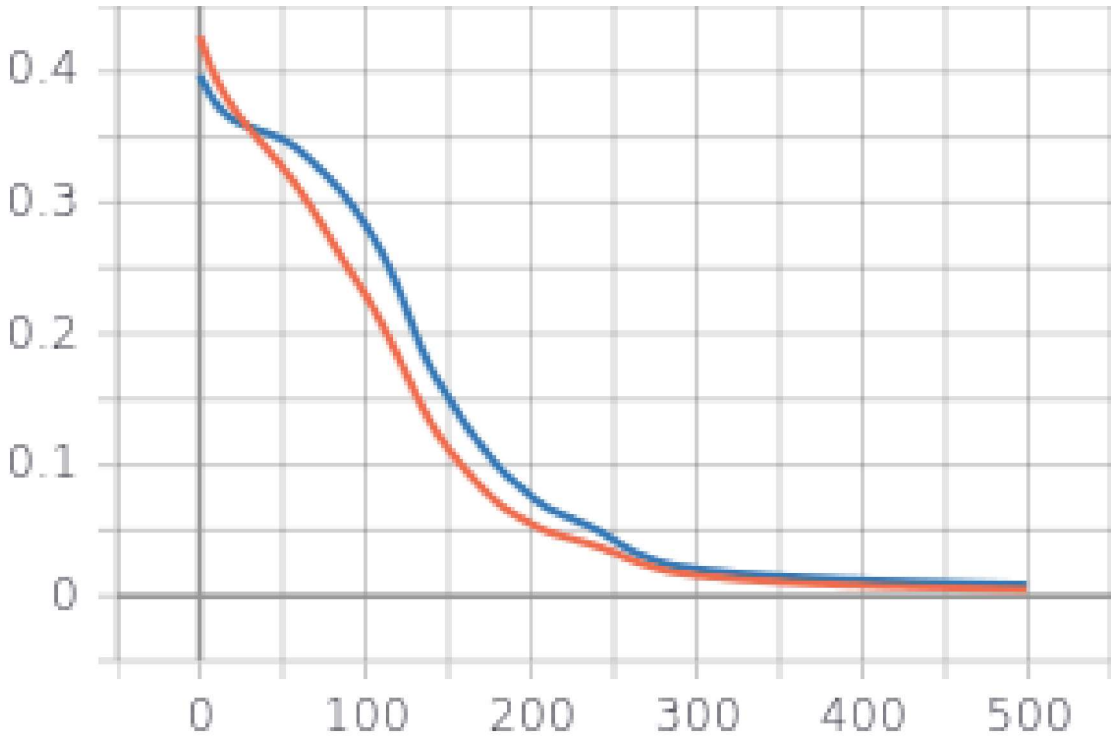

Fig. 7: The mean squared error over the training epochs on the training data (blue) and the validation date (red).

accuracy of 6.624 for the prediction of a single step 0.33 seconds in future from voxel map to voxel map.

7 Discussion

Bütepage et al. [2] [1] introduces a system using an autoencoder to predict motions based on the angels of 24 joints of the human skeleton. They compared three methods to input the timesteps to the autoencoder. In the first, the autoencoder is symmetrical, this means the decoder can be seen as the approximate inverse of the encoder. In this model the input of the autoencoder is only the actual time step and the output is one predicted timestep.

In the second autoencoder some 1d convolutional layers are added in front of the autoencoder input layer. Their kernels cover all the 24 joint angles and only fold over time. The third autoencoder uses a graph neural network to model the connections between the joints and to shrink the dimension of the input data before combining the different time steps. They got the best results with the third model. It has a mean angle error on the joint angles of 0.27 for prediction of 1 second and 0.15 for prediction of 0.08 seconds.

This result can not be compared to our result because we used voxel grids and Bütepage used joint angles. We see, that the accuracy does not decrease as fast as in our approach for long term prediction. However, Bütepage does not take into account the error in the skeleton tracking algorithm.

Martinez et al. [21] introduced a sequence to sequence model to predict 57 joint angles of the human skeleton. He used a single GRU layer with 1024 units and input and output

layers of 57 neurons. They introduced a zero velocity model as baseline. They reached a mean angle error of 1.15 for the prediction of 0.4 seconds and 0.36 for the prediction of 0.08 seconds. This result can not be compared to our result as already described above.

8 Conclusion

In summary, our algorithm is able to predict human motion 0.33 seconds into the future. For long term prediction, our algorithm, like all time series prediction algorithms, has some disadvantages compared to motion primitives.

The advantage of our algorithm is the omission of the skeleton tracking algorithm and the generation of a voxel grid that can be used by state-of-the-art robot motion planning algorithms without further conversion.

Due to the different data structures, it is not possible to compare the performance of our algorithm with state-of-the-art algorithms. Butepage et al. [2] and [1] use an autoencoder and Martinez et al. [21] use a GRU for short-term human motion prediction. As the dataset they use contains only skeleton points, it is not possible to compare our results with theirs. Generating voxel grids from the skeleton data and comparing the results on this basis could be a solution for further work.

Acknowledgment

This work was partially founded by the German Federal Ministry of Education and Research within the project KIIWI under grant number 16DHBKI060.

References

1. Butepage, J., Black, M.J., Kragic, D., Kjellstrom, H.: Deep representation learning for human motion prediction and classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE 1591–1599
2. Butepage, J., Kjellstrom, H., Kragic, D.: Anticipating many futures: Online human motion prediction and generation for human-robot interaction. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE 1–9
3. Kulić, D., Ott, C., Lee, D., Ishikawa, J., Nakamura, Y.: Incremental learning of full body motion primitives and their sequencing through human motion observation. **31**(3) 330–345 Publisher: SAGE Publications Ltd STM.
4. Luo, R., Hayne, R., Berenson, D.: Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces. **42**(3) 631–648
5. Moon, S., Park, Y., Ko, D.W., Suh, I.H.: Multiple kinect sensor fusion for human skeleton tracking using kalman filtering. International Journal of Advanced Robotic Systems **13**(2) (2016) 65
6. Cao, J., Li, Z., Li, J.: Financial time series forecasting model based on ceemdan and lstm. Physica A: Statistical mechanics and its applications **519** (2019) 127–139
7. Yamak, P.T., Yujian, L., Gadosey, P.K.: A comparison between arima, lstm, and gru for time series forecasting. In: Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence. (2019) 49–55
8. Livieris, I.E., Pintelas, E., Pintelas, P.: A cnn–lstm model for gold price time-series forecasting. Neural Computing and Applications **32**(23) (Dec 2020) 17351–17360
9. Swenja, S., Maximilian, P., Thomas, S.: Evolution of pick-by-light concepts for assembly workstations to improve the efficiency in industry 4.0. **204** 37–44
10. Hermann, A., Drews, F., Bauer, J., Klemm, S., Roennau, A., Dillmann, R.: Unified gpu voxel collision detection for mobile manipulation planning. 4154–4160
11. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks
12. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context
13. : preprint.dvi
14. : Carnegie mellon university - cmu graphics lab - motion capture library (9/27/2023)
15. von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: European Conference on Computer Vision (ECCV). (2018)
16. : Gaming datasets (7/28/2016)
17. Naureen Mahmood, Nima Ghorbani, Michael J. Black, Gerard Pons-Moll, Nikolaus F. Troje: Amass: Archive of motion capture as surface shapes
18. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(10) (2020) 2684–2701
19. : Mandery2015apdf
20. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. **6**(1) (jun 2004) 20–29
21. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks