Hand Gesture Recognition for Human-Robot Interaction

Geir Paulsen
Department of Informatics
University of Oslo
Norway
geiryp@ifi.uio.no

Sebastian Cardenas
Department of Informatics
University of Oslo
Norway
juansc@ifi.uio.no

Rosa Alsgaard
Department of Informatics
University of Oslo
Norway
rnalsgaa@ifi.uio.no

***Abstract*—For a person with average knowledge about machines and robots, interacting with one may seem hard. The traditional ways to control a robot include using a hand-held controller, a computer, or even programming skills. By omitting traditional controllers and introducing simple hand gestures, a barrier between the user and a robot is lowered. This paper utilizes the robot *TIAGo*. We mean to show how machine learning (ML) can contribute to solving the interesting questions around human-robot interaction. The method we applied for reading the hand gestures is nonintrusive regarding privacy because it does not need to record video footage of the end user. The user only has to wear a wrist sensor that collects data on the wrist movements. Thus the user hopefully feels that the interaction with TIAGo is safe, simple, and privacy-preserving. We propose a CNN-LSTM machine learning model used to recognize a set of defined hand gestures in real time. First of all, we collect data from subjects executing these gestures. Furthermore, we manually segment the data sets and use the raw data to train a single ML model for all gestures. After testing a few ML models we then check the accuracy of each of the ML models using an 8-fold cross-validation test and proceed by utilizing the model achieving the highest score. The hand gesture recognition system can be adopted to make the interaction between robots and humans smoother. We aim to make a robust system, using methods that are appealing to other types of research and can be adapted to various tasks in the field of human activity recognition (HAR) and human-robot interaction.**

***Index Terms*—Machine learning, wearable sensors, gesture recognition, interaction with robot**

I. Introduction

Every single adult you know owns a sensor that they either carry around in their pockets, store in the car, or even wear on their bodies. Smartphones, AirPods, and smartwatches share the fact that they all contain numerous sensors like GPS, touch sensors, and optical sensors, and almost everyone constantly carries one of them glued to themselves. The popularity of wearable sensors as accessories and to some degree as health monitoring devices has rapidly increased in recent years. The growth is visible in everyday electronic stores and sports shops where varieties of smartwatches and activity bracelets are exhibited. Most smartphones do as well carry a range of sensors like thermostats, accelerometers, GPS, and light sensors. At the same time, ML technology is developing [2] at a great pace. Researchers combine already established ML models to find better ways of reaching their goals and implement variations while trying to optimize their solutions. The high market demand for wearable devices in combination with ongoing research on ML models casts a broad foundation for researching wearable sensor-based HAR using machine learning. This paper utilizes two common and easily accessible sensors (accelerometer and gyroscope) which are available in many devices. In addition, the device we use has a magnetometer. We also work with one of many publicly available deep learning models that are pre-written in Python.

A. Motivation

There are two main motivations for this project. Firstly this project aims to improve human and robot interactions by improving a robot's ability to interpret gesture commands with the help of a ML model. Secondly, the project aims to improve user privacy by utilizing an Internal Measurement Unit (IMU) to broadcast the commands rather than the usage of color sensors (RGB). An inertial measurement unit (IMU) is an electronic device that measures and reports a body's specific force, angular rate, and sometimes the orientation of the body, using a combination of accelerometers, gyroscopes, and sometimes magnetometers. When the magnetometer is included, IMUs can be referred to as IMMUs [8].

B. Related work

Reference [2] systematically categorizes and summarizes existing work about wearable-based HAR and shows that the most common body part to apply a wearable sensor is the wrist. Using the wrist is $\approx$3.5 times more common than the second most popular body part. The most common applications of wearable sensors are in fitness, lifestyle, and entertainment, covering 88% of wearable sensor usage [2]. These are all fields that can benefit from research done on hand gesture recognition for interaction with robots. When it comes to ML, we see that deep learning (DL) is widely applied to solve HAR tasks. For example by applying Neural

Structured Learning [1] and Convolutional Neural Network [4] [5]. Reference [3] and [5] utilize both traditional ML and DL and thereafter evaluate the results of their methods. Some of the traditional ML models used are SVM (with polynomial kernels, meaning that it represents the similarity of training samples in a feature space over polynomials of the original variables. This allows learning of non-linear models [9]), Naive Bayes, and Random Forests. For hand gesture recognition while the subjects are moving, for example running or walking, [5] proposes a novel segmentation algorithm to distinguish between the movements coming from transposing and movements coming from actual hand movements. The algorithm takes advantage of self-correlation - when the user is walking or jogging, the sensor readings are periodic and self-correlated at the frequency of walking or jogging. Once the user performs a gesture while walking or jogging, the sensor readings are neither periodic nor self-correlated.

Common validation tests to measure how well a machine learning model classifies data are the 5-fold cross-validation test and the leave-one-subject-out cross-validation test. One paper favors the leave-one-subject-out cross-validation test over the 5-fold cross-validation test to avoid overfitting [5].

II. Methods

Figure 1 below illustrates the proposed method for this project. Firstly, a wrist sensor is used to gather hand -and arm-gesture data from from 10 volunteers, alongside video-data of the volunteers performing the gestures. Once the data has been gathered it is manually segmented and labeled as it's corresponding gesture label.

The labeled data will be split into 80% training data and 20% testing data. The training data is then split into 90% training data and 10% validation data. Using the training data we a train convolutional long short-term memory neural network (CNN-LSTM or LSTM-CNN.) The mentioned ML model is a deep learning model, which works better with larger amounts of training data. It is further explained in section *C. ML model.*

The window size, step size and epochs of the model will then be tuned and optimized before testing with the testing data.

A. Data collection

To collect the data used to train and test the ML model this project utilizes an "MbientLab MetaMotion S" sensor attached to the volunteer's wrist. The sensor transmits information regarding the wrist's movement with the help of the following sensors all of which will be used during this project.

- Accelerometer
- Gyroscope
- Magnetometer

The data obtained from the sensor gets marked with a time stamp of when it was collected. A video camera records the volunteers' gestures and is also marked with a time stamp on every frame of the video. The time stamp on the sensor and video data is then used to synchronize the two, this allows for accurate data segmentation as the video can be used to precisely determine when a volunteer starts and ends a gesture, by comparing the sensor and video timestamp. The volunteers are asked to perform the following seven hand/arm gestures using their right hand/arm.

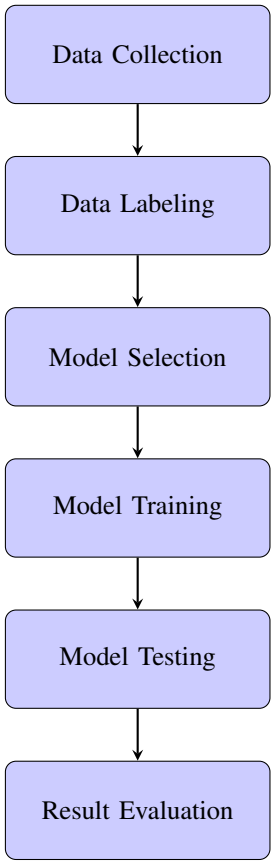

Fig. 1. Illustration of the proposed methodology

- Move forwards, palm facing the actor
- Move backward, palm facing away from the actor
- Move left-right
- Spin horizontally, like a lasso
- Spin vertically, like a magic wand
- Wave
- Go up-down, palm facing the ceiling

Each volunteer was supervised through performing the hand gestures by one of the paper authors and encouraged to perform them for as long as they found natural in light of communicating with a robot. The video data and sensor data from each volunteer were stored in two separate files. With 10 volunteers we got 10 video recordings and 10 csv-files including all seven gestures.

In figure 2 we have a box plot of the data's different features. The acceleration values vary less than the gyroscope values. In regards to the magnetometer values, we had all the participants face the same direction while doing the gestures. A gesture recognition model trained on this data could have trouble recognizing the gestures if the person performing the

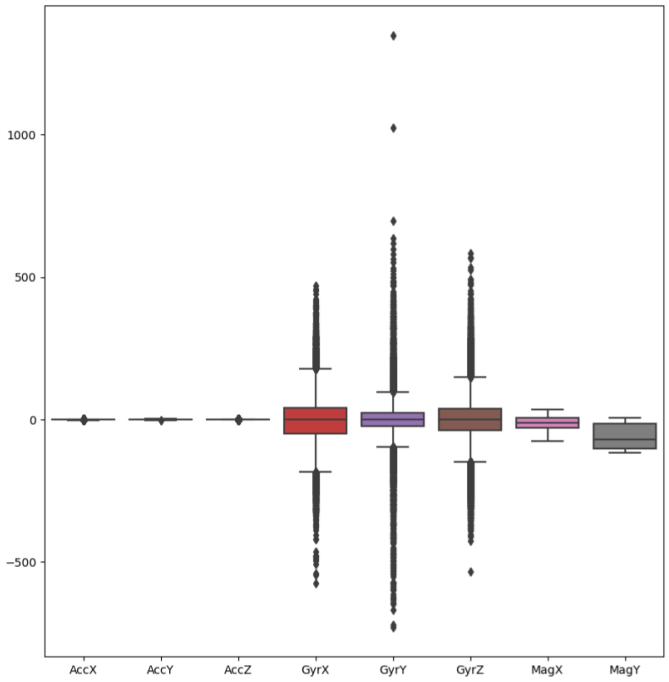

Fig. 2. Box plot of features.

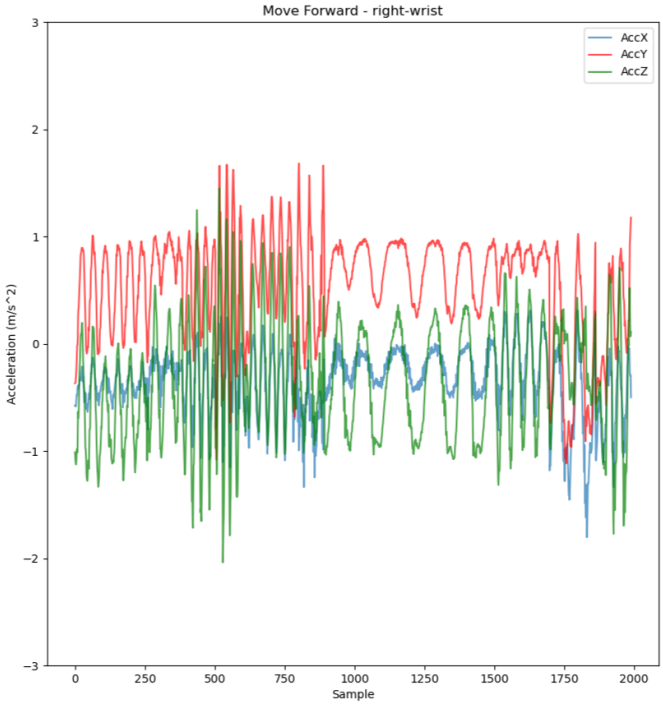

Fig. 3. Acceleration for gesture ”Move forward”.

gestures faces another direction. If this is the case will have to be investigated. With more training data, the ML model could solve this problem.

Figure 3 and 4 show the collected data of the acceleration from different volunteers. The acceleration is measured as units of gravities (g), where one $g = 9.81m/s^2$. The acceleration values are scaled by this value. This is a reason the acceleration data has relatively low values compared to the data from the gyroscope and magnetometer (see figure 2). Figure 5 and 6 show angular velocity. Figure 7 and 8 show the angle compared to the earth's magnetic field. The data is for the gestures ”Come forwards” and ”Move backward”. Some of the volunteers performed the gestures fast or at a normal pace, while some volunteers performed the gestures slowly. This is reflected in figure 3 and 4. Looking at the figures, one can see differences in the acceleration. For the gesture ”Move forward” the y-acceleration is separate from the x and y-acceleration. For the gesture ”Move backward” the y-acceleration is more similar to the x and z-acceleration. This indicates that a machine learning model could learn to distinguish between the two gestures.

B. Labeling

The data from the gestures is labeled using the aforementioned segmentation method, utilizing the time stamp obtained in the video recording and the wrist sensor sampling. We checked for asynchronization between the time stamp in the wrist sensor and the video, but found that the difference is neglectable. We encountered a difficulty after finishing the data collection. An error in our python code wrongfully translated the time stamp into the csv file of the wrist sensor data, which

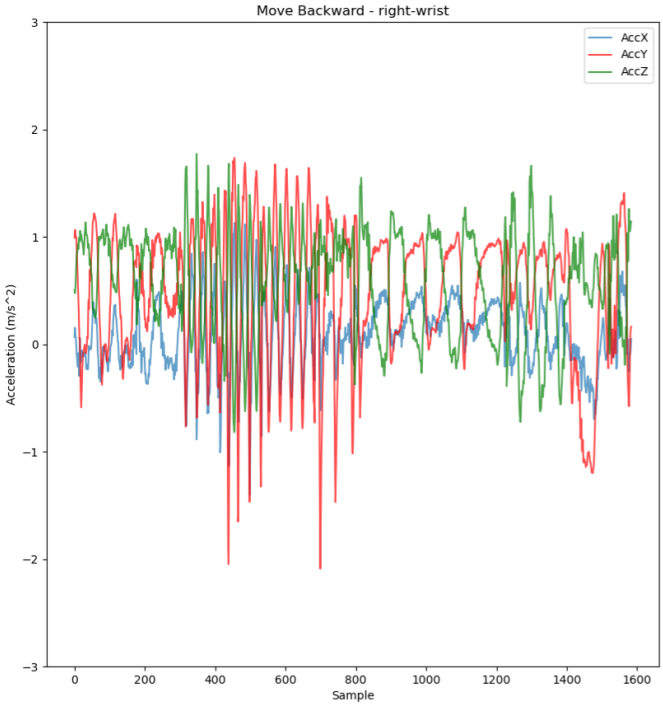

Fig. 4. Acceleration for gesture ”Move backward”.

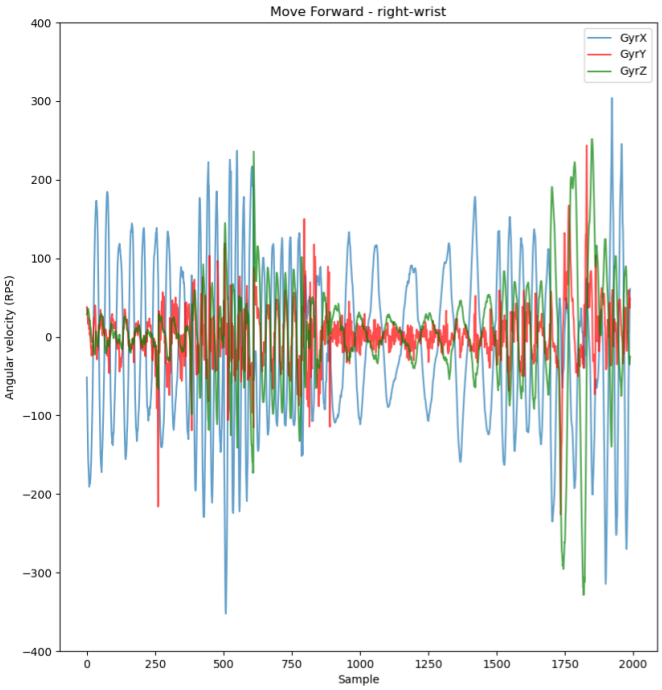

Fig. 5. Angular velocity for gesture ”Move forward”.

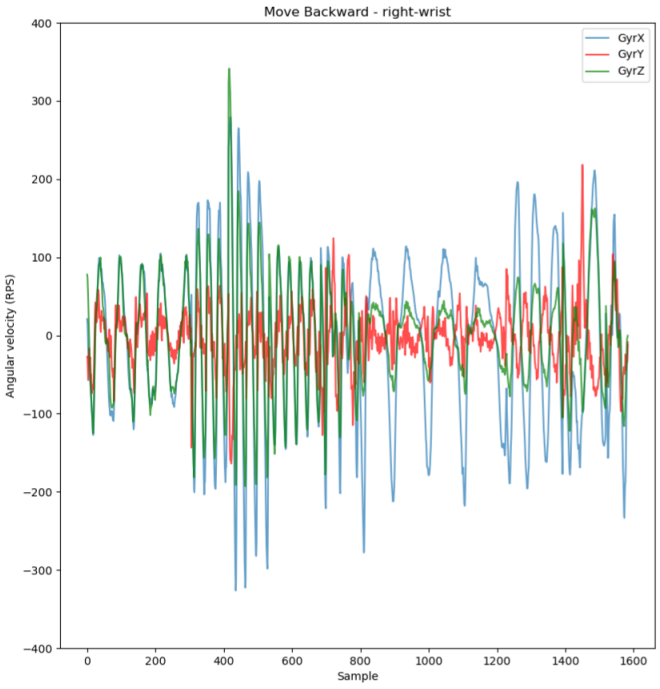

Fig. 6. Angular velocity for gesture ”Move backward”.

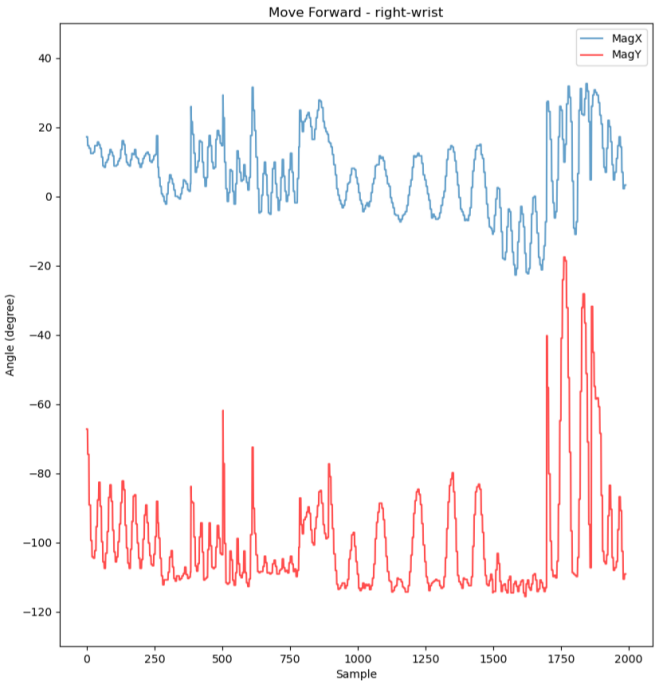

Fig. 7. Angle for gesture ”Move forward”.

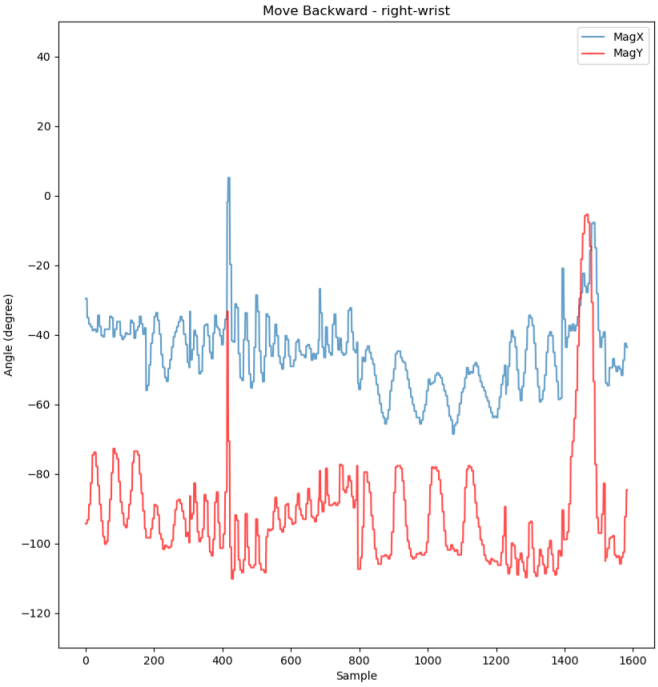

Fig. 8. Angle for gesture ”Move backward”.

made the centiseconds (cs) wrong between the first cs and the 10th cs of each second. This affected around 5 data points per second, or 5 data points per 50 data points. The difficulty was not grave enough to recollect the data, and we managed to correctly label the data by estimating which time value was supposed to be instead.

We manually labeled each data point from hand gesture 1 to 7 respectively. By looking at the videos we could see which hand gesture was performed in the corresponding data in the csv-files. Label 0 is used for all other arm movements. We also include sub-gestures in our labeling process, in case we need a finer-grained classification. Each hand gesture is divided into the two - or more - sub-gestures shown below.

- Move forwards: 11 retract arm, 12 extend
- Move backwards: 21 retract, 22 extend
- Move left-right: 31 left, 32 right
- Spin horizontally: 41 first spin, 42 second spin ...
- Spin vertically: 51 first spin, 52 second spin ...
- Wave: 61 left, 62 right
- Go up-down: 71 up, 72 down

Right or left means right or left for the volunteer. Once the data from all 10 volunteers has been gathered and labeled the resulting data packages is used to train the LSTM-CNN.

C. ML model

In order to achieve the best gesture recognition three models were tested and the best model will be selected for further tuning.

- Support Vector Machine (SVM) with a polynomial kernel
- Long Short Term Memory (LSTM)
- Convolutional Neural Network (CNN) + LSTM [10]

Model Accuracy	
Model	Validation Accuracy
SVM	79.38 %
LSTM	91.51 %
CNN+LSTM	94.73 %

The CNN+LSTM model achieved the highest validation accuracy and will therefore be selected as the main model for this project. The CNN+LSTM model will be trained using 8-fold cross validation (i.e. leave one out cross validation) training the model with the data from seven subjects and validating with the data from the remaining subject, this results in eight different final weights. When the model is initiated the weights are randomized this leads to some randomization in the final weights values. In order to determine the best final weight the model is trained ten times and the average accuracy of each weight is calculated.

Once the CNN+LSTM model has been tuned it will be tested using the testing data. The testing data consist of the complete data from 2 subjects. Three test will be conducted one for a subject that had more mechanical gestures, one for a subject with more organic gestures and finally a combination of both subjects. This is done in order to determine how the model has any bias towards mechanical or organic moving users/subjects.

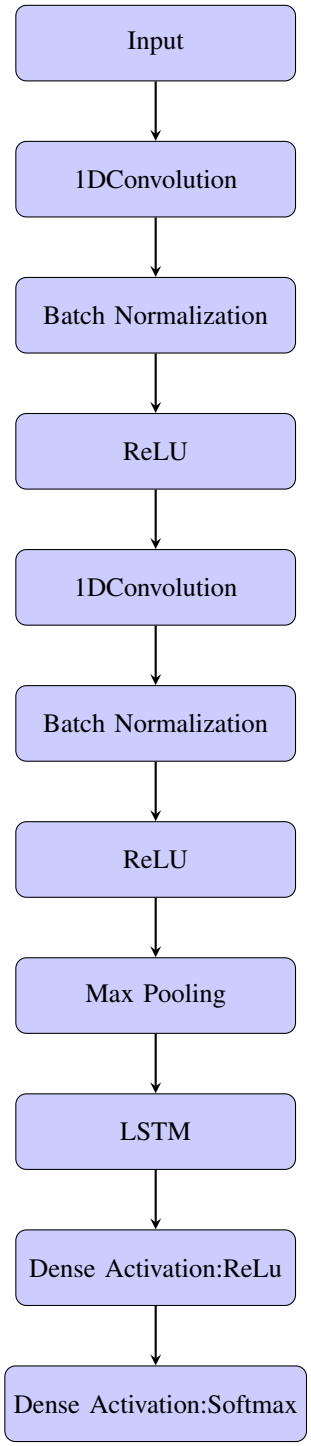

Fig. 9. Illustration of the CNN-LSTM layers

D. Tuning

In order to optimize the accuracy of the model 3 parameters will be adjusted, these being.

- Window size
- Step size
- Epochs

Window and Step size correspond to the size of each input

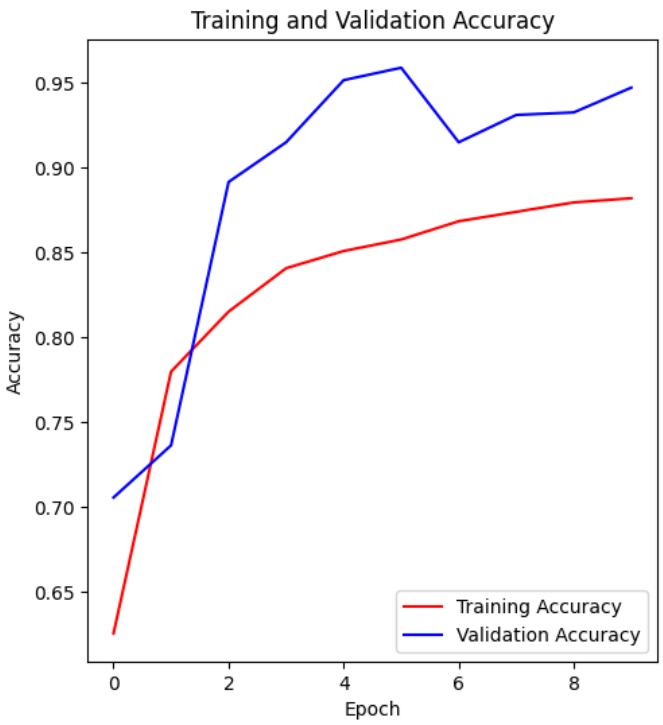

Fig. 10. Epochs vs Accuracy.

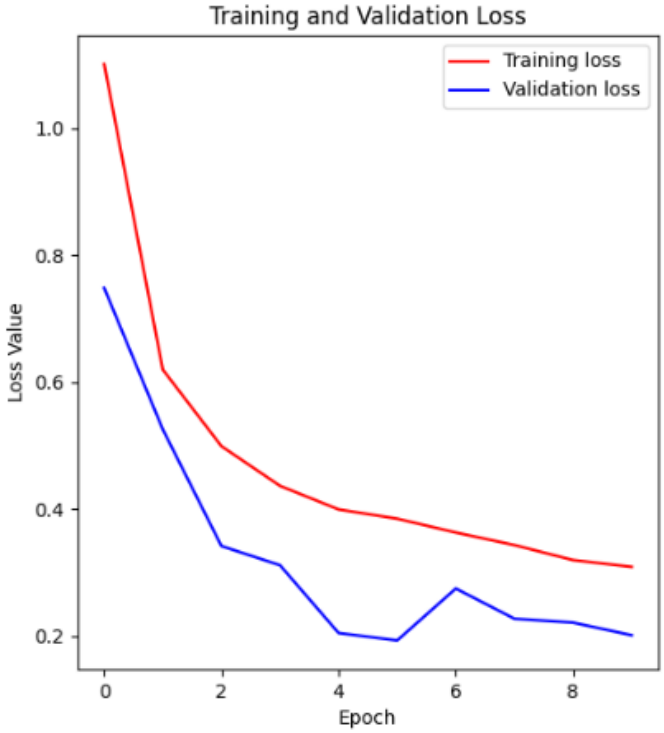

Fig. 11. Epochs vs Loss.

gesture. For example if a window had the size 5 and a step of 3 it would take five continues sample of the data-set , determine the corresponding gesture, move down three samples and repeat. Should there be an overlap between gestures the input will be set as the dominant gesture for the window.

Due to the limited amount of data collected for this project we were restricted whilst selecting a window size that could accommodate the sample size of each gesture, as some gestures had over 200 sample and others had less than 70. After some trials it was determined that the best values for these two parameters were a window size of 11 with a step size of 5. Notice that the window size was specifically chosen to be an odd number in order to avoid issues when selecting a dominant gesture for the window when there were equal amounts of samples for two gestures, for example five gesture-0 samples and five gesture-2 samples. It was also determined after trial and error that 10 was the optimal amount of epochs to avoid under and over-training. This gives us the following figures for loss fig. 11 and accuracy fig. 10.

III. Ethics

This project has some ethical requirements that need to be taken into account. Because the ML models will be applied to human data. We will be collecting data from our fellow students as volunteers to have data to train our ML model. There will be no physical injury, or financial, social, or legal harm posed to the participants. The participants may leave the study at any time they want. As the project involves human data collection we will comply with corresponding national guidelines to ensure the privacy of research subjects and the protection of data [6]. Potential ethical aspects of the proposed research formed by the use and transfer of data from humans will be addressed by

1) The project has been notified to the Norwegian Center for Research Data (NSD).
2) Guaranteeing confidentiality and anonymous processing of data.
3) Acting by the Guidelines for Research Ethics (NENT, 2008 [7])
4) Complying with the research guidelines for adolescents (over 15 years)
5) Obtaining approval from humans involved in the experiments through the use of consent forms.
6) Deletion of all video recordings and other non-essential sensitive data once the project is over

The storage and processing of sensitive data will be carried out inside the university's GitHub infrastructure in a private repository. Seeing as how the project mainly aims to improve human-robot interactions there is no apparent way that the results of this project could be implemented with any malicious intent. Additionally, this project does not discriminate, exclude, or otherwise negatively impact people. Furthermore, the VIROS project data management plan will guide the research in this project. The research activities will ensure continuing compliance and will take into account relevant revisions to the mentioned legislation and directives.

Regarding reproducibility, we used the following packages which need to be preinstalled:

- Python3 version 3.10.12
- Pip version 23.3.1
- MetaWear SDK-Python
- PyWarble
- OpenCV version 4.8.0

Furthermore, the raw data used to run the ML models are structured into 10 csv-files - one per participant. A CSV file is structured by 11 columns: One for time stamp (hour, minute, seconds, and centiseconds), three for accelerometer data, three for gyroscope data, and two for magnetometer data. The remaining two columns are used for gesture and sub-gesture labels. A CSV file has around 2000-4000 data points, corresponding to one row per data point. The first row of a CSV file is the header, and the data starts from the second row. The data set we collected for this paper is made public at *insert link/explanation of how to gain access here*.

IV. Results

After the best weight is selected using eight-fold cross-validations (i.e., leave-one-subject-out) the model is tested using the two test subjects given the following results.

TABLE I
GESTURE, PRECISION, RECALL

Gestures	Precision	Recall
nothing	93 %	98 %
move forward	88 %	95 %
move backward	93 %	98 %
move left-right	92 %	94 %
circle horizontal	87 %	92 %
circle vertical	97%	54 %
wave	81 %	81 %
up down	96%	88%

Precision and recall are given by the equations below. A true positive is a test result that correctly indicates the presence of a gesture. A false positive is a test that incorrectly indicates the presence of a gesture. A false negative classifies a gesture as a different gesture.

$$Precision = \frac{t_p}{t_p + f_p} \tag{1}$$

$$Recall = \frac{t_p}{t_p + f_n} \tag{2}$$

This gives us a test accuracy of 91.68% with a test loss of ca. 0.305 as well as the following confusion matrix fig 12.

Testing each test subject individually resulted in a test accuracy of 93.5% for the subject with organic movements and an accuracy of 90.6% for the subject with more mechanical movements. This suggests that the CNN+LSTM model has a bias towards more organic movement as opposed to mechanical movement.

By taking a look at their corresponding gesture, precision, and recall data as well as their corresponding confusion matrices below fig 13 & 14

It is clear by looking at the recall of the vertical circle for the mechanical movements that a lot of them are getting miss

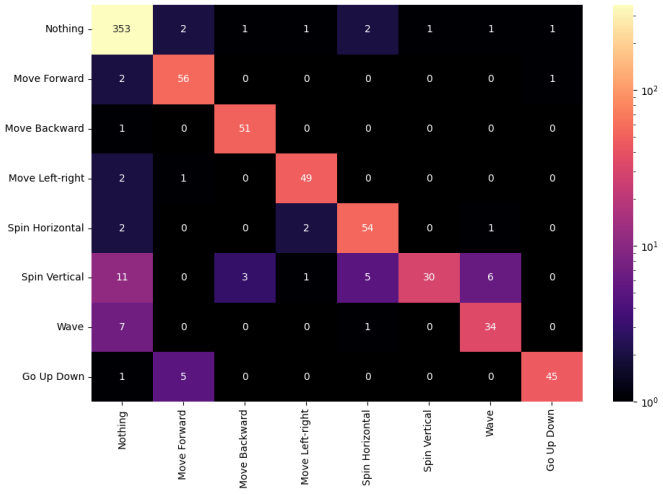

Fig. 12. Confusion Matrix of both test subjects.

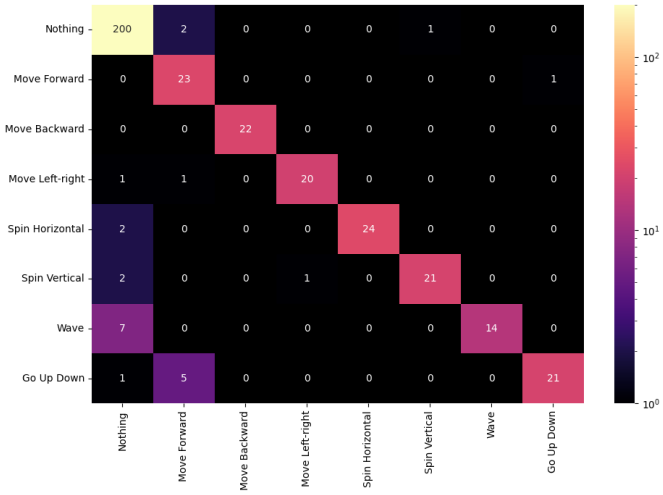

Fig. 13. Confusion Matrix of Organic-movement test subject.

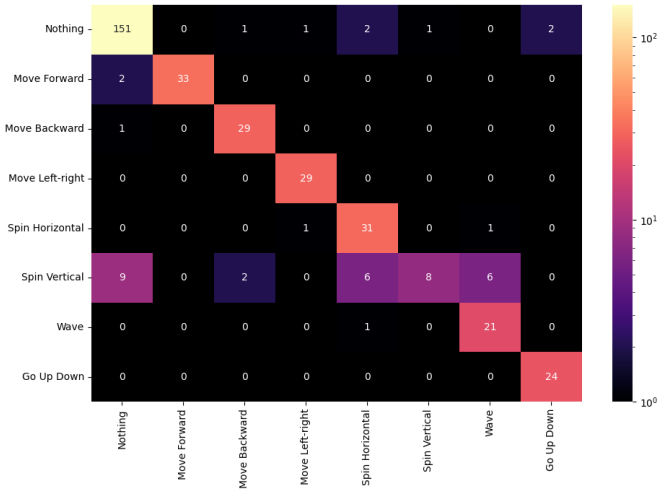

Fig. 14. Confusion Matrix of Mechanical-movement test subject.

TABLE II
ORGANIC: GESTURE, PRECISION, RECALL

Gestures	Precision	Recall
nothing	94 %	99 %
move forward	74 %	96 %
move backward	100 %	100 %
move left-right	95 %	91 %
circle horizontal	100 %	92 %
circle vertical	95%	88%
wave	100 %	67 %
up down	90%	78 %

TABLE III
MECHANICAL GESTURE, PRECISION, RECALL

Gestures	Precision	Recall
nothing	93 %	96 %
move forward	100 %	94 %
move backward	91 %	97 %
move left-right	94 %	100 %
circle horizontal	78 %	94 %
circle vertical	89%	**26**%
wave	75 %	95 %
up down	92%	100 %

misclassified as other gestures this is made even more apparent when looking at the corresponding confusion matrix.

After reviewing the subject's video data it was determined that this erroneous classification was caused by the subject rotating their arm counterclockwise and therefore not an indication of mechanical vs organic-movement bias.

Comparing the confusion matrix of the SVM model figure. 15 and the CNN+LSTM model fig. 12 it is aparent that the CNN+LSTM model greatly outproforms the SVM model. This is due to the fact that this project focuses on the classification of time-sequential data a factor that is overlooked in the SVM as it only classifies individual samples rather than a sequence of samples.

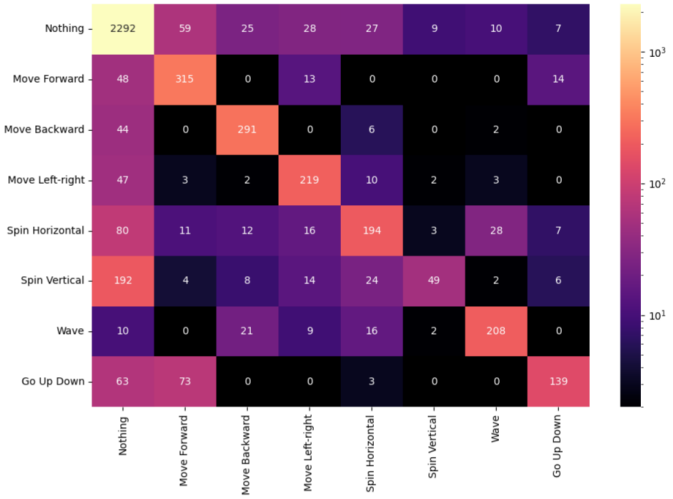

Fig. 15. Confusion Matrix SVM Model.

V. CONCLUSION

We have used an IMU to collect measurements from 10 volunteers. In addition, we recorded a video which we used to label the data with gesture data. Of the machine learning models we tried, we found the cnn+lstm model achieved the best result on the validation set. The cnn+lstm machine learning model got a 94.73% accuracy score. Using the cnn+lstm model on our test set, we get good results as illustrated in figure 12 and table I.

The test set contains the measurements from 2 volunteers. In figure 13 and 14 (which corresponds to the results in table II and III) we have separated the test set, one for each volunteer. We see that cnn+lstm performs well, except for predicting the gesture "Spin vertical". We believe this is due to the direction (clockwise vs. counterclockwise) the volunteer spins his arm. In the training set, some volunteers spin their arms clockwise and some spin their arms counterclockwise. We have therefore less consistent data for this gesture, which could explain why the model is bad at recognizing this gesture for this volunteer.

Additionally, the model performed exceptionally well when distinguishing between the come forwards and go backward gestures. These two gestures were identical apart from the orientation of the subject's hand. Indicating the possible implementation of gestures that are executed similarly.

A. Future work

Though this project got positive results it is still possible to improve them by redoing some of the data collection as well as further testing of the current data sets. Regarding further testing of the current data set, it became apparent that shallow learning models are not adequate for this project. Therefore we recommend that any future model testing of the collected data be performed using DL models. While collecting the data used for this project each volunteer performed each gesture once for approximately 3 seconds which proved to yield a very small sample size. Should this data collection be redone we recommend having the volunteers perform each gesture multiple times. Another possible improvement could be to have the volunteers perform the gestures while facing in different orientations, to get more robust and reliable magnetometer data.

REFERENCES

[1] Z. Uddin and A. Soylu, "Human activity recognition using wearable sensors, discriminant analysis, and long short-term memory-based neural structured learning," Sci Rep 11, 16455. Oslo, August 2021.

[2] S. Zhang et al., "Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances," Sensors, vol. 22, no. 4, pp. 1476, February 2022.

[3] M. Nguyen, L. Fan, and C. Shahabi, "Activity Recognition Using Wrist-Worn Sensors for Human Performance Evaluation," 2015 IEEE International Conference on Data Mining Workshop, pp. 164-169 Atlantic City, NJ, USA, 2015.

[4] C. A. Ronao and S-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," Expert Systems with Applications, Volume 59, pp. 235-244. 2016.

[5] H. Zhao, Y. Ma, S. Wang, A. Watson, and G. Zhou, "MobiGesture: Mobility-aware hand gesture recognition for healthcare," Smart Health, Volumes 9–10, pp. 129-143. 2018.

[6] Datatilsynet, Personal Data Act. As of 14 April 2000 No. 31 relating to the processing of personal data. 2000
[7] The National Committee for Research Ethics in Science and Technology (NENT), Guidelines for Research Ethics in Science and Technology. 2008
[8] B. Fang, F. Sun, H. Liu and C. Liu, “3D human gesture capturing and recognition by the IMMU-based data glove”, Neurocomputing, Volume 277, p. 198-207, 2018.
[9] Wikipedia page, polynomial kernel. https://en.wikipedia.org/wiki/Polynomial_kernel
[10] ParthJain02, Ankit Sharma, ”Mobile Health Human Behavior Analysis”. https://www.kaggle.com/code/parthjain02/cnn-lstm-95