MINISTRY OF EDUCATION AND TRAINING

HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY OF MECHANICAL ENGINEERING

🙞🙜✪🙞🙜

PROJECT OF ROBOTICS AND AI

RECOGNIZING THE MOTION PHASES OF REHABILITATION BICYCLES USING AI MODELS

Supervisor:	**Dr. Cái Việt Anh Dũng**
Students:	**Lê Đức Pháp Anh**
ID:	**21134001**
Class:	**21134NT**

Ho Chi Minh City, December 2024

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM
KHOA CƠ KHÍ CHẾ TẠO MÁY

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do – Hạnh phúc

NHẬN XÉT ĐỒ ÁN CỦA GIẢNG VIÊN HƯỚNG DẪN

Họ và tên sinh viên: Lê Đức Pháp Anh MSSV: 21134001

Tên đề tài: Nhận dạng các pha chuyển động của xe đạp phục hồi chức năng

Ngành đào tạo: Robot và Trí tuệ nhân tạo

Họ và tên GV hướng dẫn: TS. Cái Việt Anh Dũng

NỘI DUNG NHẬN XÉT

Điểm: ……… **Bằng chữ: ………**

Ý kiến bổ sung:

……………………………………………………………………………………………
……………………………………………………………………………………………

TP.HCM, ngày tháng năm 20

Giảng viên hướng dẫn

(*(Ký, ghi rõ họ tên)*

COMMITMENT

- Project: RECOGNIZING THE MOTION PHASES OF REHABILITATION BICYCLES USING AI MODELS
- Supervisor: Dr. Cai Viet Anh Dung
- Students: Le Duc Phap Anh
- ID: 21134001
- Phone number: 039 260 4589
- Project report submission date: 12/2024
- Commitment: "I hereby affirm that this graduation thesis is the result of my own research and work. I have not copied anything from published sources without due attribution. In the event of any infringement, I accept complete responsibility for my acts."

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to Ho Chi Minh City University of Technology and Education, particularly the distinguished members of the Mechanical Engineering Faculty. Their steadfast support and invaluable guidance have been essential throughout my academic journey, significantly contributing to the completion of an important part of my training program. The faculty's exceptional teaching has provided me with not only in-depth technical knowledge but also a strong foundation for academic and professional growth.

I am especially grateful to Dr. Cai Viet Anh Dung, whose expertise and mentorship have been pivotal in addressing the challenges of this project. His thoughtful guidance and insightful feedback have greatly enhanced the scope and quality of my research.

As with any academic work, errors are inevitable, and I welcome any constructive feedback and suggestions from the academic community to further improve the comprehensiveness of this report. Your input will be invaluable in refining my findings.

Sincerely,

Le Duc Phap Anh

ABSTRACT

Health is one of each individual's most valuable assets and serves as the foundation for society's long-term development. However, the world today faces numerous health challenges, ranging from infectious to noncommunicable diseases. Modern medicine has made significant advances in disease prevention and treatment, from common diseases to dangerous diseases that affect long-term health or extremely serious conditions, saving the lives of many patients. The advancement of science and technology is one of the factors driving and supporting this development. New technologies and advanced methods are constantly being researched and applied in medicine to improve the ability to treat, support patients' treatment, increase recovery ability, or improve human health quality. The medical community is interested in one of the developing aspects of medicine: patient rehabilitation. After being cured of serious health-threatening diseases, patients require an appropriate recovery plan to regain their health. However, this process is quite complicated, necessitating medical evaluation and support to ensure successful recovery. There are related technology systems to support this process, such as rehabilitation bicycles, which are used to restore patients' lower limbs. However, previous systems used simple systems, and when pedaling, there may be insufficient force interaction with the patient, reducing the effectiveness of rehabilitation. As a result, newer, more appropriate systems are required to improve patient recovery. AI is a new technology that is widely used in many fields today, effectively supporting and expanding the capabilities of technological systems in everyday life. As a result, the use of AI to support rehabilitation systems is unavoidable in order to improve the quality and effectiveness of patient recovery.

The goal of this project is to create an AI model capable of recognizing the motion phases of a rehabilitation bicycle, which will aid in the design of a control system to support interactive bicycles when patients use a single-arm rehabilitation bicycle effectively.

The model is intended to analyze the motion phases of the bicycle using sens ors that collect data on the bicycle, thereby determining which phase the bicycle is for optimal interaction.

To support the control system, the finished model will be integrated into embedded software systems or microcontrollers.

Within the framework of this project, we will focus on building an AI model to recognize the motion phases of a rehabilitation bicycle, building a usage orientation for the model for embedded systems or microcontrollers to use for control.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
DL	**D**eep **L**earning
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
GPUs	**G**raphics **P**rocessing **U**nit
GRU	**R**ecurrent **N**eural **N**etwork
HMM	**H**idden **M**arkov **M**odel
ICU	**I**ntensive **C**are **U**nit
LSTM	**L**ong **S**hort **T**erm **M**emory
ML	**M**achine **L**earning
RNN	**R**ecurrent **N**eural **N**etwork
TN	**T**rue **N**egative
TP	**T**rue **P**ositive
TPUs	**T**ensor **P**rocessing **U**nit

CHAPTER 1. INTRODUCTION

1.1. Thesis background

In recent years, the science of rehabilitation has made considerable advances, particularly in the development of equipment to assist patients during treatment in intensive care units (ICU). Cycling-based rehabilitation devices are an important and popular technique for helping patients maintain muscle activity and recover from severe illness. Globally, there are more and more rehabilitation devices that help muscle repair and improve lower limb function [1][2][3].

Passive and active cycling methods in rehabilitation have been widely used due to their benefits, such as enhancing blood circulation, improving cardiovascular health, and assisting patients in maintaining muscles, hence reducing the danger of muscular atrophy when lying in bed for an extended period. There have been studies on improving rehabilitation bicycle equipment [4], technology for passive rehabilitation bicycles (cycling machines) [5], and the effectiveness of the products [6], mobility support systems for the elderly [7], osteoarthritis, and technologies related to rehabilitation equipment with force feedback and motion phase detection algorithms [8][9][10]. However, the items have disadvantages when used with ICU patients, such as a lack of flexibility, fixed functionalities, and difficulty of usage.

In Vietnam, the rehabilitation profession is still in the process of developing and implementing new technology, particularly in larger hospitals and rehabilitation centers. Cycling-based rehabilitation equipment is primarily imported, with limited in-country research and development. Products for indoor exercise that require consumers to pedal have fixed resistance. The systems can suit the needs of a set of subjects, but they lack flexibility, and their performance falls short of both user needs and patient recovery capabilities.

The project's purpose is to investigate and develop a suitable AI model to recognize the mobility phase of the rehabilitation bicycle, allowing the system to interact with the patient more effectively and improve the patient's recovery efficiency. Recognizing the motion phases will assist the system in providing suitable interactions, increasing the system's flexibility, and allowing the system to connect with the user more correctly and effectively.

1.2. Scientific and practical significances

- Scientific significance: Research aimed at establishing research areas and developing artificial intelligence applications in fields such as spatial or temporal series prediction. Develop and perform extensive research on data processing techniques and specialised models for spatial or temporal series data.
- Practical significance: Using AI to improve the efficiency of medical devices contributes significantly to the development of Vietnamese medicine and technology.

1.3. Objectives

This project aims to investigate and develop an AI model to recognize the mobility phase of the rehabilitation bicycle by collecting and analyzing data from sensor attach in rehabilitation bicycle, training and evaluating model to take the best one, then building a usage orientation for the model for embedded systems or microcontrollers to use for control.

1.4. Research methods

- Research and survey articles relate to our topic.
- Collect data and survey factors about our system use in the project with there characteristic.
- Study the advantages and disadvantages of AI models used to train time series data and state space data.

1.5. Structure of the report

The report contains four chapters:

- **CHAPTER 1. INTRODUCTION:** Brief introduction to the study.
- **CHAPTER 2: THEORETICAL BASIS**: This section summarizes medical research on lower limb rehabilitation and rehabilitation device models. This section will also cover theory related to understanding AI models and data processing.
- **CHAPTER 3: PROPOSED METHOD**: Describes the AI model deployment process, including preparation, implementation, and expected outcomes.

- **CHAPTER 4: EXPERIMENTAL RESULT**: Present the project implementation process, results achieved and limitation.
- **CHAPTER 5: CONCLUSION AND RECOMMENDATION:** Evaluate the achievements and limitations of the project, and propose a direction for project development.

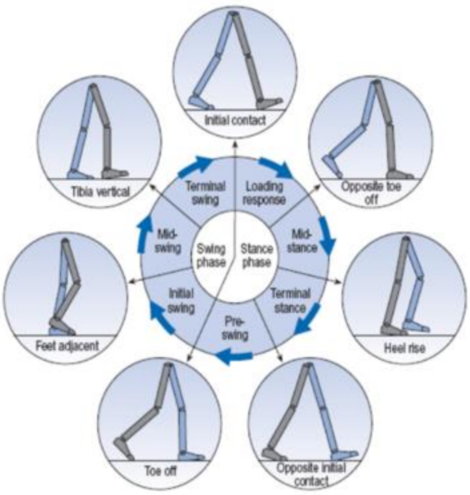
CHAPTER 2. THEORETICAL BASIS

2.1. Introduction of lower limb rehabilitation and lower limb rehabilitation device

2.1.1. Medical aspects of lower-limb rehabilitation

Humans have long had extensive knowledge of human movement, particularly movement on the lower limbs, which can be applied in the design of systems to serve humans.

The article [11] presented an understanding of how humans walk, with the term "gait cycle" referring to the movement cycle of the lower limbs when walking. [11] also thoroughly examined the movement phases in a cycle, related parameters, and medical aspects of this study.



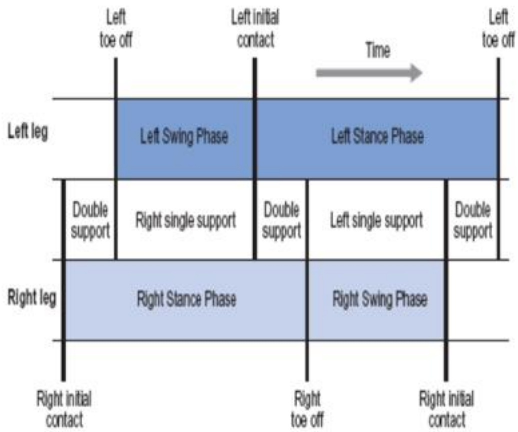


Figure 1: Gait cycle

In a study [12], to create a video-based cycling motion analysis monitoring system, they referred to cycling motion theory and proposed their design. Unlike walking, cycling has fewer phases and is easier to perform.

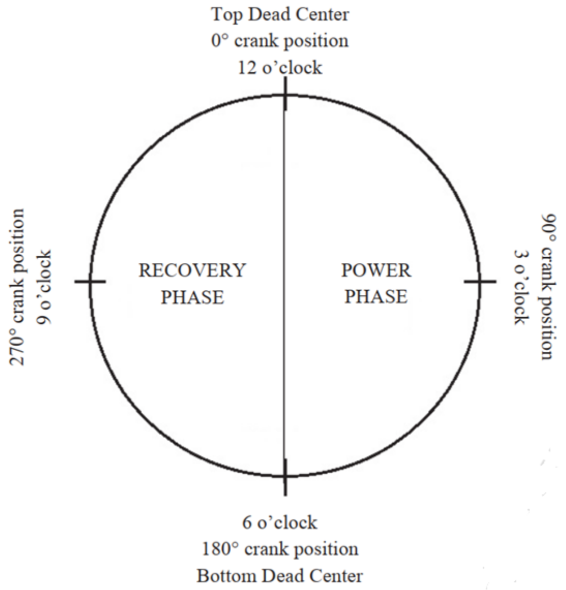


Figure 2: Period of cycling biomechanics

The article [13] presented in detail the concepts of cycling biomechanics, cycling phases/pedalling, and cycling anatomy, giving us a better understanding of the medical aspects of lower limb movement when cycling.

With this understanding, people can apply it to medical treatment, particularly the problem of lower limb rehabilitation.

2.1.2. Lower limb rehabilitation device

In this project, we will use a rehabilitation bicycle model to gather survey data on human pedaling motion. The goal is to analyze model data and use AI to learn motion phase recognition for the control system. The rehabilitation bicycle model is shown below.

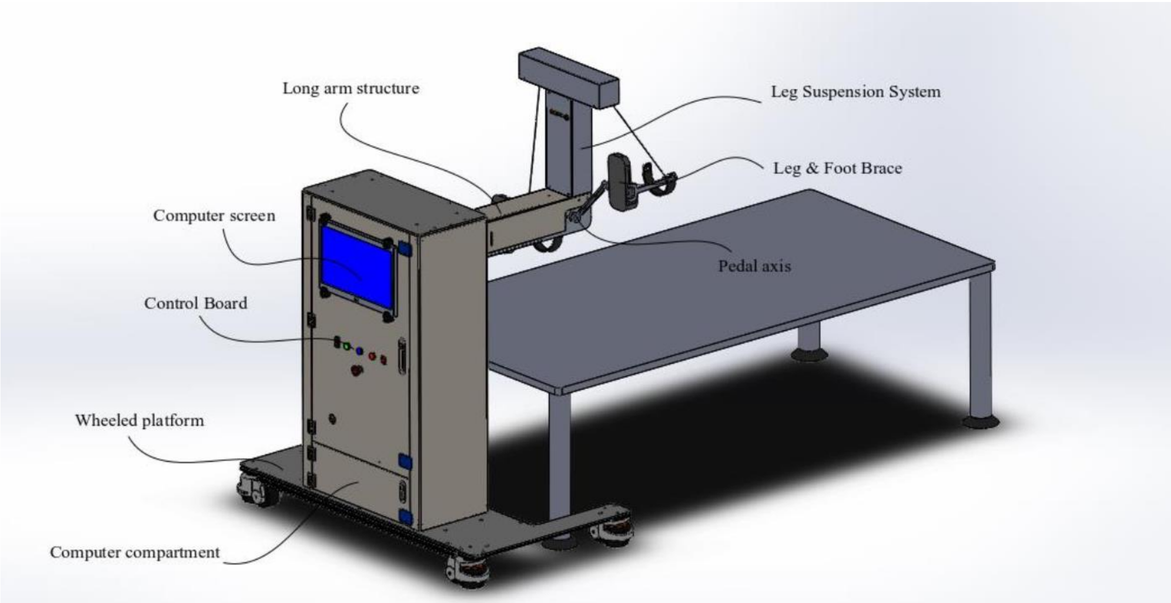


Figure 3: Rehabilitation bicycle model

The rehabilitation bike's main components are:

- Mechanical system
- Sensors mounted on the rehabilitation bike to collect data, including a torque sensor on the pedal hub and a sensor that measures knee joint tension.
- A computer for monitoring and collecting data.

2.2. Data analysis of lower limb rehabilitation device

2.2.1. Overview of data

The data obtained from the sensors is string data. The processed data produces raw_data, as shown below.

Discription 1: Sample of Data

3.992	3.19	-28.78	-13.93	1.94	18
4.097	3.3	-28.78	-13.93	4.59	40
4.200	3.3	-28.77	-13.98	5.11	60
4.321	3.47	-28.76	-14.08	4.72	80
4.414	3.47	-28.74	-14.21	3.7	95
4.519	3.45	-28.73	-14.33	5.01	119
4.628	3.45	-28.72	-14.45	7.88	156
4.748	3.45	-28.7	-14.6	11.84	218
4.962	3.62	-28.63	-14.79	16.32	355
5.011	3.6	-28.63	-14.79	16.32	385
5.075	3.6	-28.55	-15.38	14.94	421
5.167	3.6	-28.48	-15.68	11.96	468
5.299	3.6	-28.41	-15.92	9.48	512
5.410	3.37	-28.32	-16.08	7.74	546
5.519	3.37	-28.24	-16.21	7.58	579
5.627	3.37	-28.1	-16.37	8.5	618
5.736	3.42	-28.02	-16.44	10.34	663
5.928	3.48	-28	-16.51	11.97	766
5.997	3.48	-28	-16.51	14.18	809
6.078	3.5	-28	-16.51	16.27	863

We have a complete dataset after combining additional sensor-related information.

Discription 2: Data overview

#	Column	Non-Null	Count Dtype
---	------	--------------	-----
0	date	1136 non-null	object
1	t	1136 non-null	float64
2	Tau_Motor	1136 non-null	float64
3	Tau_1	1136 non-null	float64
4	Tau_2	1136 non-null	float64
5	vel	1136 non-null	float64
6	encoder_count	1136 non-null	int64
7	mode	1136 non-null	object
8	level	1136 non-null	int64
9	turn	1136 non-null	int64
10	period	1136 non-null	int64
11	push_leg	1136 non-null	object
12	degree	1136 non-null	float64
13	phase	1136 non-null	int64
14	Tau_Motor_deriv	1136 non-null	float64
15	Tau_1_deriv	1136 non-null	float64
16	Tau_2_deriv	1136 non-null	float64
17	vel_deriv	1136 non-null	float64

dtypes: float64(10), int64(5), object(3)

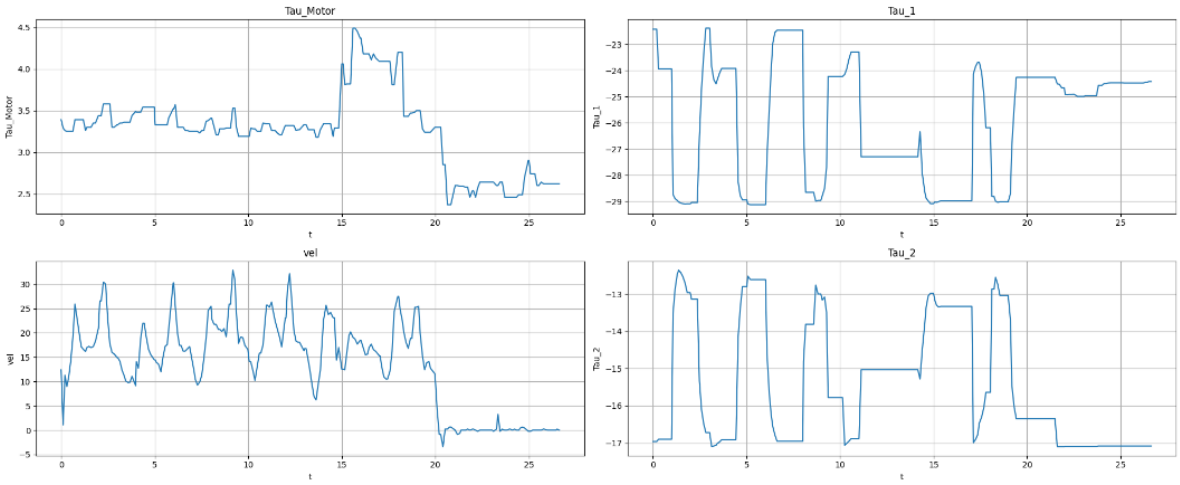


Figure 4: Plot of each data column

2.2.2. Methods for analyzing time series data

After obtaining the data, we must analyze and observe it, as well as process it before training.

The obtained data shows that there is a consistent and stable cycle when pedaling. This is ideal when we need the model to learn the data's characteristics in order to recognize motion phases.

In addition, we can use a statistical parameter chart to help us understand the data's statistical properties.

____________ Statistics of numeric features ____________

	t	Tau_Motor	Tau_1	Tau_2	vel \
count	16128.000000	16128.000000	16128.000000	16128.000000	16128.000000
mean	34.022997	2.732377	-26.507776	-14.750412	15.191560
std	23.050236	0.512732	2.598550	1.807684	12.020508
min	0.000000	0.380000	-29.330000	-17.190000	-6.940000
25%	14.505000	2.490000	-29.090000	-16.640000	0.230000
50%	30.992000	2.650000	-27.300000	-14.110000	17.470000
75%	49.792000	3.130000	-23.980000	-12.860000	24.830000
max	92.736000	4.710000	-22.300000	-12.240000	44.990000

	level	turn
count	16128.000000	16128.000000
mean	1.702877	19.337550
std	1.148009	8.446206
min	0.000000	1.000000
25%	1.000000	13.000000
50%	2.000000	21.000000
75%	3.000000	27.000000
max	4.000000	31.000000

Figure 5:Statistics of feature

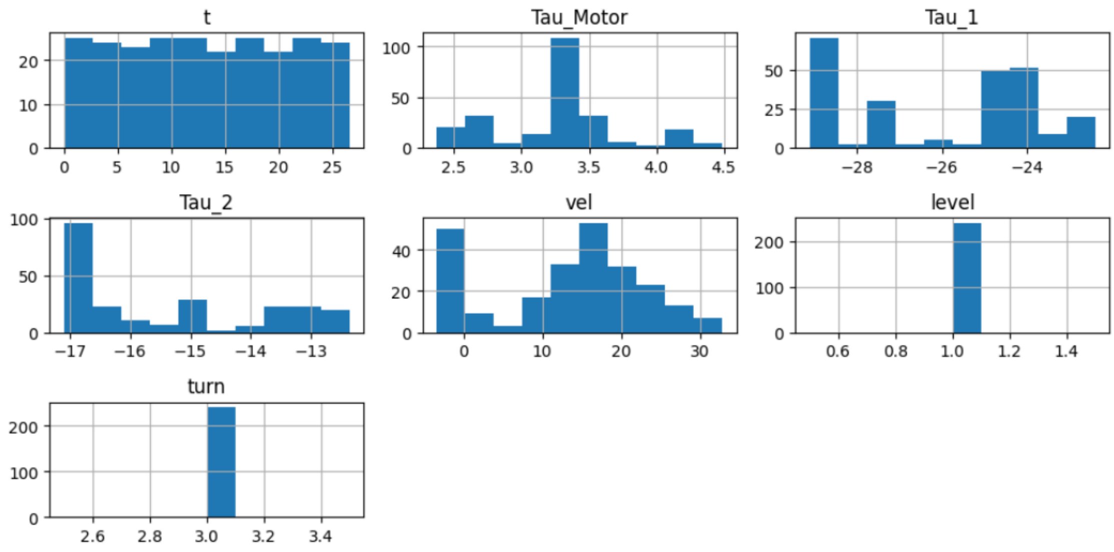


Figure 6:Histogram of feature

2.2.3. Label data

Since the train data is used to identify the phase based on the option we want, the method for dividing the phase to label is also necessary. There are numerous ways to divide the phase depending on how to aid in the recovery of the lower limb. In the context of this project and our model, we will use a simple phase division model to survey and design, with the goal of identifying the phases that will support the necessary interactions. The phase division diagram is illustrated in the figure below.

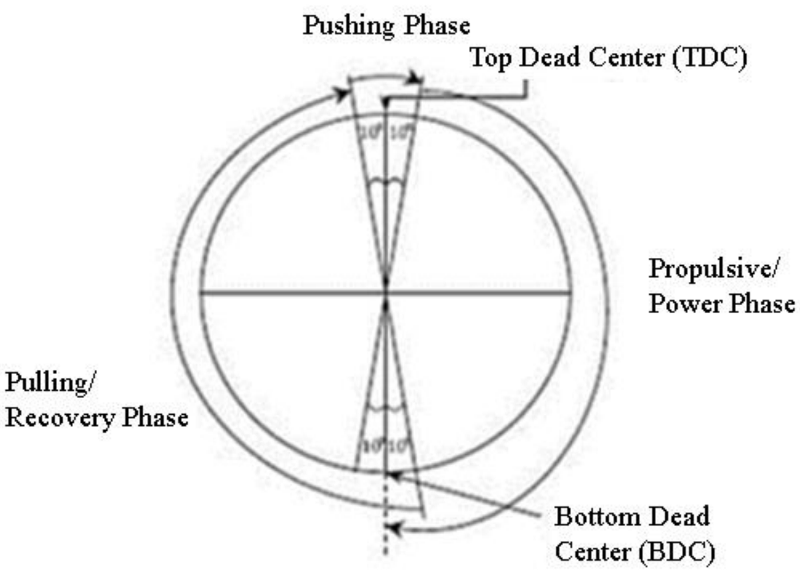


Figure 7: Diagram of phases divide

We divide it into four phases: two pedaling phases (Top Dead Center and Bottom Dead Center) and two balancing pedaling phases (power phase and recovery phase). To label the phase, data must be analyzed in cycles, and synchronized video while pedaling can be used to determine which phase the pedaler's foot is in.

After dividing the phase by the data, we get the phase distribution chart, which is shown below.

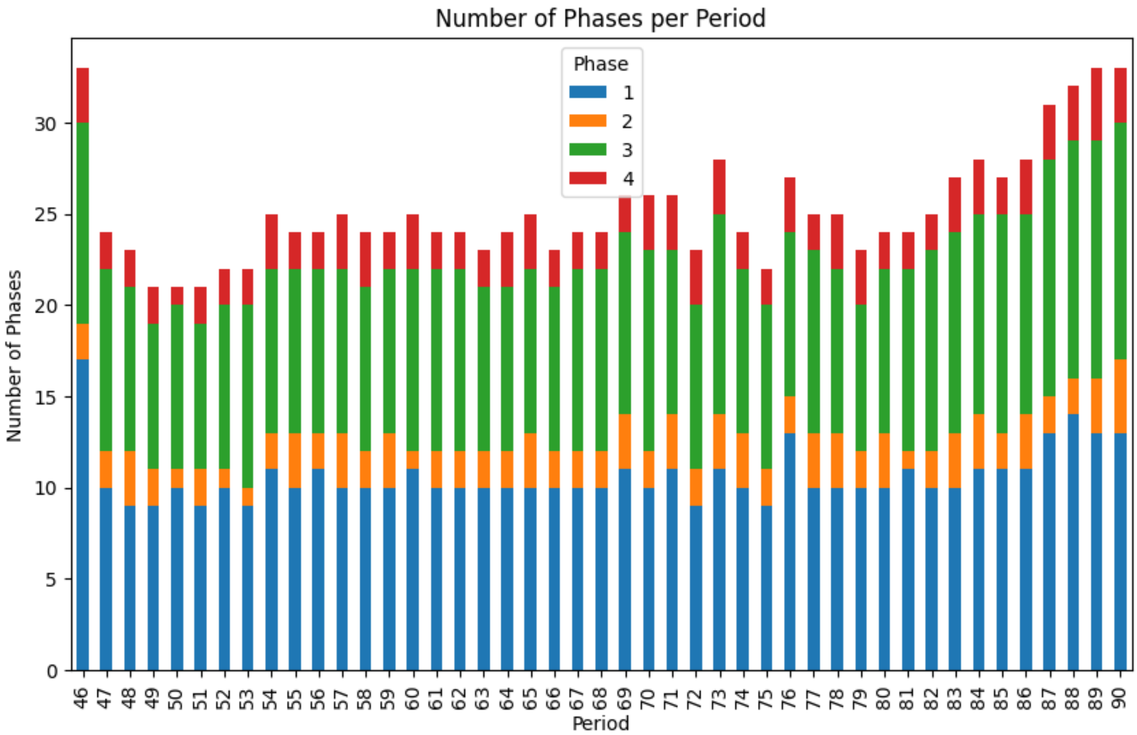


Figure 8: Distribution diagram of phase's label

2.3. AI model

We use a combination of state space and time serial data. As a result, we need to select an AI model that is appropriate for our data.

The Markov chain model, for example, can be used with state space data. The model is frequently used when the data has states that are dependent on the previous state; with the Markov chain, the model is even more unique when it only depends on the previous state (short-term time); and the data is probabilistic when it transitions between states.

The Hidden Markov Model is an extension of the Markov chain model. HMM is used when we do not directly predict the output state based on the previous state sequence, as in the Markov chain, but rather on the relevant data collected (for example, the sensor collected in that state). This is ideal when we need to predict the output state without knowing the previous state, but only using the input data collected. However, because HMM is an unsupervised learning model, the output is more focused on input properties rather than classification. As a result, it can be used in conjunction with a supervised learning or neural network model to achieve the desired classification.

Another model for training this type of data is the Recurrent Neural Network (RNN). RNN is a type of artificial neural network capable of processing sequential data while retaining information from previous time steps. As a result, the model can be used to identify phases when it is capable of learning from previous experiences. RNN variants include Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

Because the amount of data collected is limited and the system's nature necessitates quick processing and short-term information, we chose the RNN model for this project.

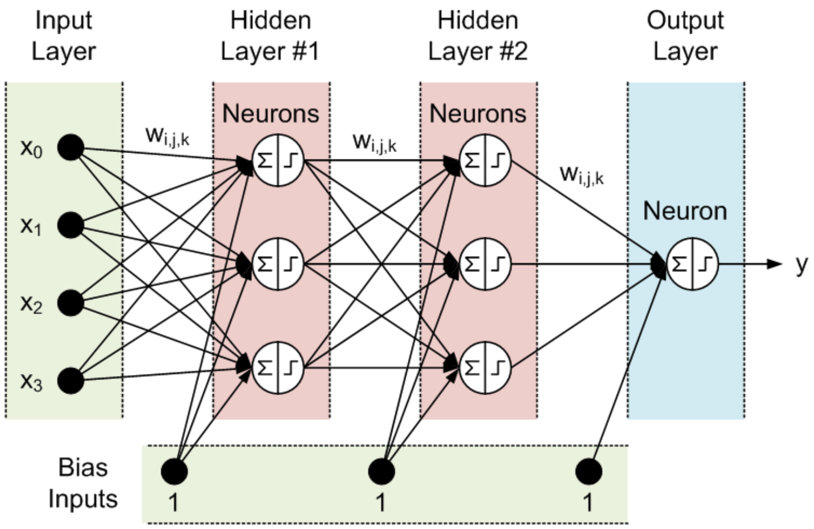


Figure 9: Structure of RNN

Unlike feedforward neural networks, which process data in a single direction, RNNs have a feedback loop that allows them to "remember" past information. Key Main components:

- Input Layer: Receives the input data at each time step, can be a single value, a vector, or a sequence of values.
- Hidden Layer: The core of the RNN, responsible for processing the input and maintaining a "memory" of past information. It consists of recurrent units that are connected to each other in a loop. Each unit receives input from the

previous time step's hidden state and the current input. It produces an output that is passed to the next time step's hidden state and the output layer.

- Output Layer: Produces the final output of the network. The output can be a single value, a vector, or a sequence of values.
- Recurrent Unit: the fundamental building block of an RNN. It typically consists of weights and biases to be learned during training to adjust the network's behavior; activation function introduces non-linearity into the network, allowing it to learn complex patterns.

- Feedback Loop: allows the hidden state to "remember" past information. At each time step, the hidden state is updated based on the current input and the previous hidden state. This allows the RNN to capture long-range dependencies in the input sequence.

CHAPTER 3. PROPOSED METHOD

3.1. Requirement of system design

3.1.1. AI model for lower limb rehabilitation device

With the goal of recognizing the mobility phase of the rehabilitation bicycle, allowing the system to interact with the patient more effectively and improving the patient's recovery efficiency, the designed AI model must classify the phase at each time with high accuracy (85 - 90%) and in the correct order in the data sequence.

3.1.2. Target for AI deployment

After training, the AI model should be simple, lightweight, and deployable on microprocessors, embedded boards, or microcontrollers. Computation speed is also an important consideration for this model.

3.2. Proposed AI model

I conducted a survey and research on models commonly used in articles about time series and state space data. The following comparison table summarizes the evaluation of different models based on key metrics and their brief performance remarks follow the Table 1

Table 1: AI model comparison

Model	Strengths	Weaknesses
HMM-ANN	Combines the strengths of HMMs (modeling hidden states) and ANNs (learning complex patterns). Can capture complex relationships between hidden states.	Difficult to train, requires large amounts of data.
RNN	Handles sequential data well, captures dependencies between time steps.	Can suffer from vanishing/exploding gradient problems for long sequences.
LSTM	Variant of RNN, addresses vanishing/exploding gradient problem with gates. Can learn long-term dependencies.	More complex than RNN, computationally expensive.

GRU	Variant of RNN, simpler than LSTM but still captures long-term dependencies.	Performance might not be as good as LSTM in some cases.

To achieve the required phase recognition, in addition to the time series - the state space of the data, ensuring learning based on the time factors and state order of the data with accuracy in short time series and poor data, the RNN model is considered the most suitable because of its simplicity while still ensuring accuracy with this type of data. As a result, we decided to use an RNN model for our project.

Regarding input data processing before training, I propose a method according to *Figure 10*.

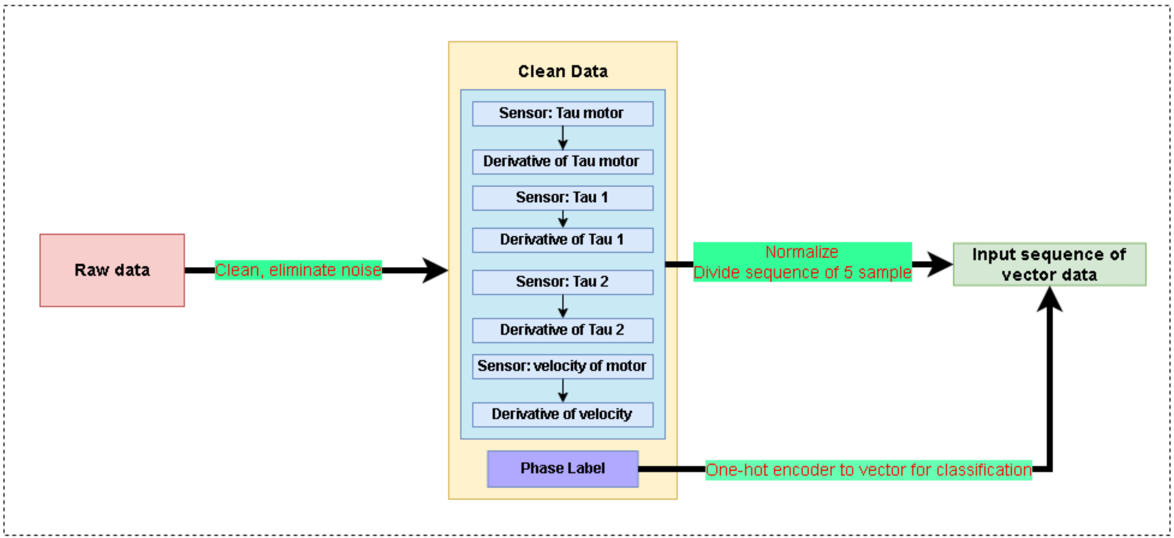


Figure 10: Data processing pipeline

Raw data obtained during processing will be separated into data sequences for each sensor and label phase. The sensor data sequences will be enhanced with derivative features before being trained. The sensor data will then be normalized to ensure that it falls within a uniform range (0,1), making the training process easier. After normalization, the data is still being divided into sequences in preparation for training. In the label phase, the data will be converted to vector form using a one-hot encoder technique to facilitate classification training. Following processing, the data will be reassembled and organized in a format suitable for training the RNN model.

Scaling data before training the model helps to balance the input values, accelerate convergence, and improve model stability. It reduces the influence of unit differences, avoids numerical precision problems, and makes data processing more efficient, thereby improving prediction accuracy and performance. I will perform data normalization according formula (3.1)

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

Where:

- X is the original value of the data
- X_{min} and X_{max} are the minimum and maximum values in the data
- X_{scaled} is the normalized value, which lies within the range [0; 1]

When dividing sequence data, we follow the structure shown below:

Figure 11: Structure of input data train

We split the sequence by adding 4 previous data vectors to the data vector to form a sequence, from which the model can classify which phase the sample is in at that time.

About output, we use one-hot encoder technique for data label conversion. One-hot encoding is a technique used in machine learning to convert categorical data into a numerical format that can be understood by algorithms. It determines all the unique values within the categorical variable, and then create a new binary column for each unique category.

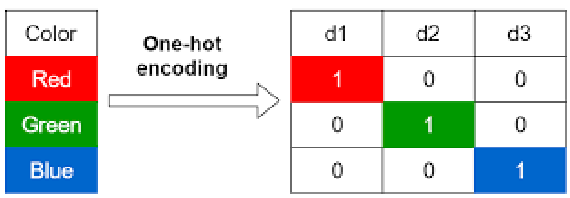


Figure 12: One-hot encoder technique

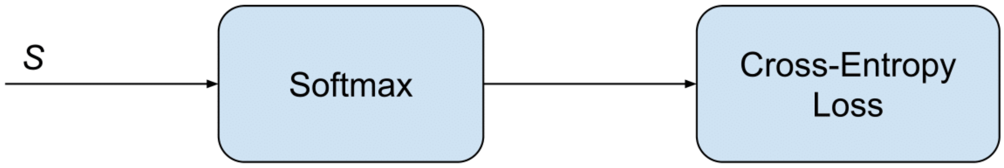


Figure 13: Categorical Cross-Entropy loss function

About the loss function, I choose Categorical Cross-Entropy, also called Softmax Loss. It is used for multi-class classification. For the electricity consumption forecasting task, it is commonly used due to its ability to penalize larger errors more severely, making it suitable for time series forecasting problems. The formular of this function shows as (3.2)

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$$

$$CE = -\sum_i^C t_i \log(f(s)_i) \tag{2}$$

Where:

- s: scores or logits output by the model for each class
- $f(s)_i$: softmax function applied to the scores
- CE: Cross-Entropy scores
- t_i: true label for the class. It is a binary value

The loss function aims to minimize the difference between the predicted probabilities $f(s)_i$ and the true labels t_i. When the model correctly predicts the class ($t_i = 1$), the loss for that class is $-log(f(s)_i)$. If the model predicts the correct class with high probability ($f(s)_i$ close to 1), the loss for that class will be close to 0. When the model incorrectly predicts the class ($t_i = 0$), the loss for that class is 0.

3.3. Evaluation metrics model

I choose commonly used model evaluation metrics for classification data including: accuracy, precision, and recall.

First, we must understand how we evaluate classify data. There are four categories associated with the actual and predicted class of an evaluation:

- True positive (TP): Cases where the model correctly predicts the positive class.

- False Positive (FP): The given observation is negative but the predicted value is positive.
- True Negative (TN): The given observation is predicted to be positive, despite, in fact, being positive.
- False Negative (FN): Both the actual and predicted values of the given observation are negative.

One commonly used evaluation metric for binary classification is accuracy, which measures the proportion of correctly classified instances out of the total number of instances. While accuracy provides a general measure of the model's correctness, it may not be sufficient in cases where the classes are imbalanced or when the costs of misclassification vary significantly.

$$Accurancy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

Precision and recall are two additional important evaluation metrics for binary classification. Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive, emphasizing the model's ability to minimize false positives. Recall, on the other hand, calculates the proportion of correctly predicted positive instances out of all actual positive instances, focusing on the model's ability to capture all positive instances.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

The consfuse matrix allows us to view the combined outcomes of the four categories which correspond to the actual and predicted class.

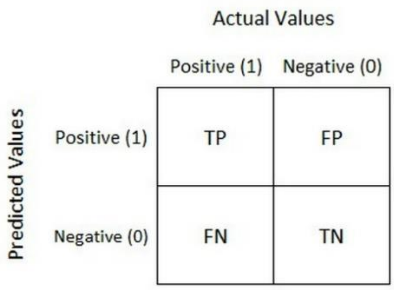


Figure 14: Consfuse matrix

CHAPTER 4. EXPERIMENTAL RESULT

4.1. Environment setting

We use Google's ecosystem for storage, data interaction, and ease of use, including Google Colab for data usage and training code.

The free cloud-based platform Google Colab, short for Colaboratory, enables users to write and run Python code right within their browser. It offers access to GPUs and TPUs for accelerated computation, which makes it especially helpful for machine learning (ML) and deep learning tasks. This is a quick overview of how to train machine learning models with Google Colab. It is simple to scale with GPUs and TPUs, doesn't require setup, integrates with well-known Python libraries, and is perfect for learning, prototyping, and experimentation.

4.2. Data processing and model training

4.2.1. Data processing

Following data collection, the raw data is in the format specified in section 2.2.1. The data is then cleaned and preprocessed in accordance with the procedure to get it ready for the training phase. Following processing, the data will be split into three sections: testing, validation, and training. Eighty percent of the data is used for training, and twenty percent is used for testing. To train on the training set with a 20% train set ratio, the validation set is randomly split.

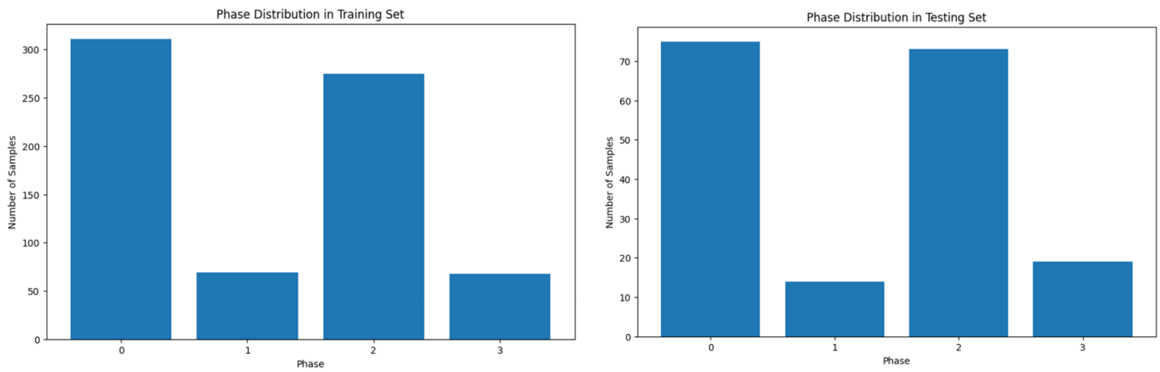


Figure 15:Diagram of phase distribution

4.2.2. Model training

After processing the data, we proceed to train the model. In addition to the main model, we use, which is RNN, we also try the HMM-ANN model for comparison.

When training, we will assess the index on the test set after using the design model on the training set and validating on the validation set. TensorFlow, PyTorch, Scikit-learn, and other frameworks are used to deploy training code in order to use and design the aforementioned model for training. We employ the TensorFlow framework in our project because it’s well-suited for deep learning tasks and highly optimized for GPU/TPU in Colab.

Next, I continue to fine-tune the model to find parameters for the models to perform best on my dataset. Below are the parameters for RNN model that I have fine-tuned:

RNN model:

- Number of layers: 2
- Number of neurons in each layer: 64
- Dropout: 0.2
- Activation function: Tanh, Sigmoid
- Optimizer: Adam
- Loss function: Categorical Cross-Entropy
- Epochs: 70
- Batch size: 64
- Leaning rate: 0.0006
- Callbacks: Early stop after 10 patience

Finally, we test the model in testset and have the result:

- Test Accuracy: 0.9006 ≈ 90.06%
- Test Precision: 0.9056 ≈ 90.56%
- Test Recall: 0.9006 ≈ 90.06%

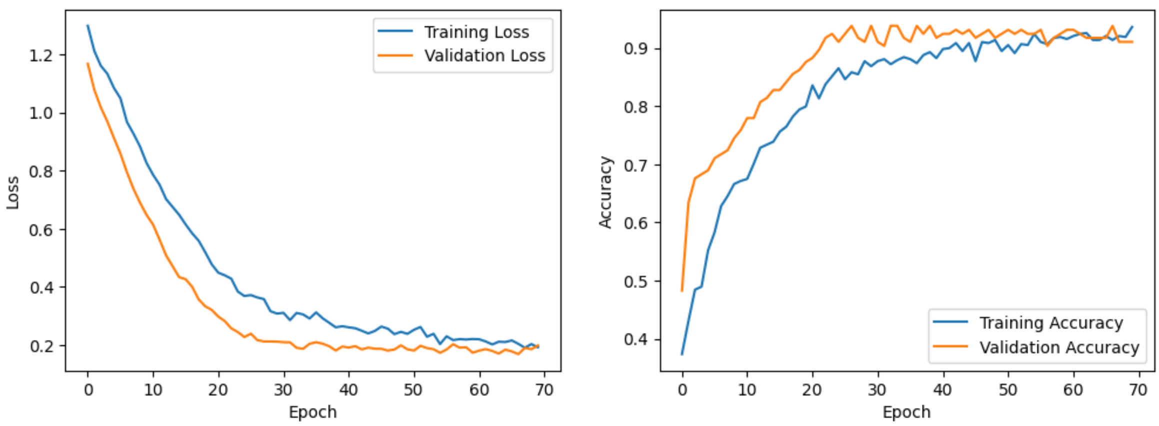


Figure 16:Training process

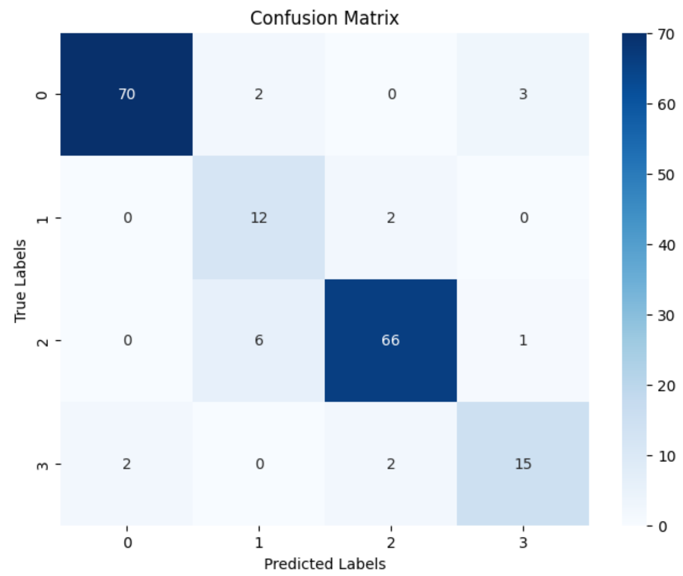


Figure 18:Confuse matrix result after training

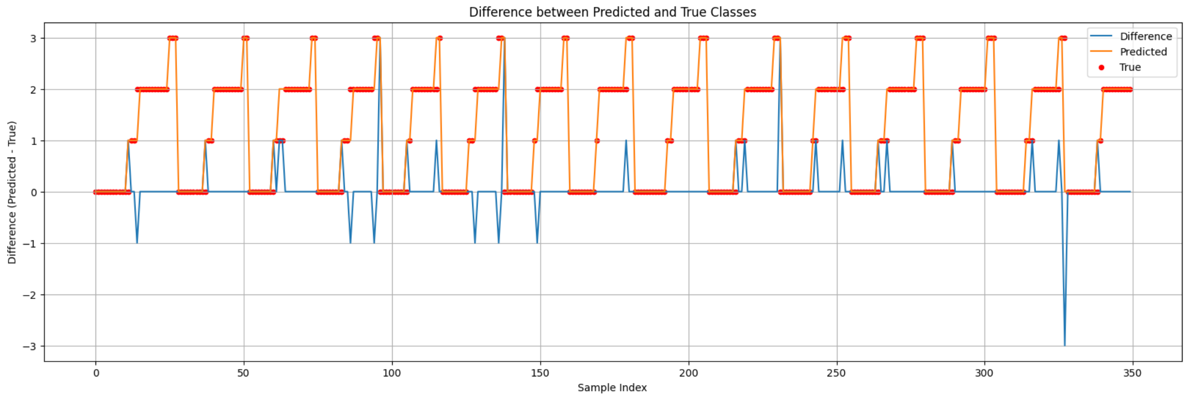


Figure 17: Plot of comparition over time

between predicted and true phase with RNN model

According to the aforementioned parameters, the model has a high accuracy rate in identifying phases and misses or confuses them less frequently.

A recognition graph also demonstrates the accuracy of the model as shown in Figure 18.

The diagram shows that the model classifies the phases very accurately, ensuring that the phases are in the correct order during operation; prediction errors are primarily caused by the model's slow or fast phase.

Compare to HMM-ANN, we have the result:

- Test Accuracy: 0.8242
- Test Precision: 0.8693
- Test Recall: 0.7308

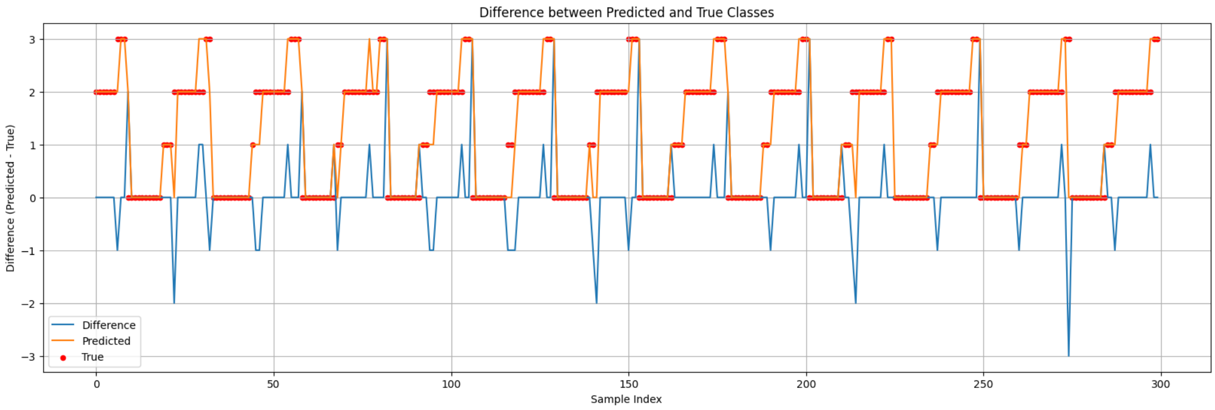


Figure 19: Plot of comparition over time

between predicted and true phase with HMM-ANN model

We can see that the model has low accuracy and does not keep the exact order as when using RNN.

4.3. Implement model

Once the model has been obtained, it is loaded onto embedded systems or microcontrollers. Because the model runs on the TensorFlow framework, it is simple and lightweight to package for system deployment.

4.4. Limitation

Although the model produces positive results, there are some limitations that may affect the model's output. Because the model was trained on small amounts of data, it may not provide the same level of accuracy when tested in real-world scenarios. Furthermore, the model deployment is only at the expected level, which is an impediment to deploying in practice because flexibility may be compromised, and the model may not be suitable for some deployment hardware. When deploying the model in practice, you should also consider the maintenance and update process.

CHAPTER 5. CONCLUSION AND RECOMMENDATION

To summarize, in the most recent project, we created an AI model utilizing RNN architecture to recognize the mobility phase of a rehabilitation bicycle, which we then used to allow interactive system control. The model achieved the project's objectives by providing test results with high accuracy (90%), ensuring the order of phases over time. We hope that the research effort adds practical value to the field of AI research for modern medicine. In the future, we will continue to deploy the model on a real operating system model, addressing existing flaws and optimizing the model's capabilities.

REFERENCES

[1] D. A. Brown and S. A. Kautz, *Increased workload enhances force output during pedaling exercise in persons with post-stroke hemiplegia*, Stroke, vol. 29, no. 3, 1998.

[2] T. W. Janssen, J. M. Beltman, P. Elich, P. A. Koppe, H. Konijnenbelt, A. de Haan, K. H. Gerrits, *Effect of Electric Stimulation – Assisted Cycling Training in People with Chronic Stroke*, Archives of Physical Medicine and Rehabilitation, Vol. 89, 2008

[3] T. Fujiwara, M. Liu, and N. Chino, *Effect of pedaling exercise on the hemiplegic lower limb*, American Journal of Physical Medicine and Rehabilitation, vol. 82, no. 5, pp. 357–363, 2003.

[4] S. Nakatani, N. Araki, T. Hoshino, O. Fukayama, and K. Mabuchi, *Brain-controlled cycling system for rehabilitation following paraplegia with delay time prediction*, Journal of Neural Engineering, vol. 18(1), 2021.

[5] C. Allen, A. Blasius, and K. Puttre, *Passive Assistive Pedaling Device*, A Major Qualifying Project Report Submitted to the Faculty of the WORCESTER POLYTECHNIC INSTITUTE in partial fulfillment of the requirements for the Degree of Bachelor of Science in Mechanical Engineering, 2014.

[6] Laura Jurema dos Santos, Fernando de Aguiar Lemos, T. Bianchi, A. Sachetti, Ana Maria Dall' Acqua, Wagner da Silva Naue, Alexandre Simões Dias, and Silvia Regina Rios Vieira, *Early rehabilitation using a passive cycle ergometer on muscle morphology in mechanically ventilated critically ill patients in the Intensive Care Unit (MoVe-ICU study): study protocol for a randomized controlled trial*, 2014.

[7] Carola Engbers, *Keep cycling: how technology can support safe and comfortable cycling for older adults,* 2019.

[8] M. Casadio, V. Sanguinetia, P.G. Morassoa, and V. Arrichiellob, *A new haptic workstation for neuromotor rehabilitation. Technology and Health Care - IOS Press*, (14) :123–142, 2006.

[9] A. Gupta, M. K. O'Malley, V. Patoglu, C. Burgar. *Design, Control and Performance of RiceWrist: A Force Feedback Wrist Exoskeleton for Rehabilitation and Training*, International Journal of Robotics Research. 2008;27(2):233-251.

[10] Truong *et al.*, *Functional outcome and muscle wasting in adults with tetanus*, Transactions of The Royal Society of Tropical Medicine and Hygiene, Volume 113, Issue 11, November 2019, Pages 706–713.

[11] Ashutosh Kharb, Vipin Saini, Y.K Jain, Surender Dhiman, *A review of gait cycle and its parameters*, IJCEM International Journal of Computational Engineering & Management, Vol. 13, July 2011.

[12] Karolina Kopacz1, Magdalena Fronczek-Wojciechowska, Piotr Kosielski, Gianluca Padula, *Movement analysis in tandem track cyclists using video analysis*, Antropomotoryka. Journal of Kinesiology and Exercise Sciences, JKES 68 (24): 59-64, 2014.

[13] Cycling Biomechanics. Physiopedia. https://www.physio-pedia.com/Cycling_Biomechanics

[14] Do Duong (Jul 30, 2018). Recurrent Neural Network - Overview and applications https://viblo.asia/p/recurrent-neural-networkphan-1-tong-quan-va-ung-dung-jvElaB4m5kw