DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

Developing Peer-Trained Autonomous Companion Robots to Tackle Isolation

RAVI BIR

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Developing Peer-Trained Autonomous Companion Robots to Tackle Isolation

RAVI BIR

Master’s Programme, Machine Learning, 120 credits
Date: September 17, 2021

Supervisors: Katie Winkle, Sarah Gillet
Examiner: Iolanda Leite

Swedish title: Utveckling av peerutbildad autonom sällskapsrobot mot social isolering

Abstract

There has been a sharp rise in the loneliness and social isolation felt amongst young people, resulting in an increased risk of developing many harmful conditions. Quarantining and remote working due to the recent COVID-19 pandemic have made the situation worse. Therefore, this thesis aims to develop a companion robot that provides social support and reduces the social isolation and loneliness felt by students in their dorm rooms. A Wizard-Of-Oz style study is used to capture the ideal behaviour of the robot. By allowing the human to take full control of the robot and deploying it with a user, the actions and behaviour the human chooses for the robot can be recorded and later used to create autonomous behaviour for the robot.

The autonomous behaviour is controlled by two machine learning classification models. Information collected from the environment and the interactions with the user are combined to create a statespace. Given an input statespace, the first model determines if an action should be performed, and the second model determines which action this should be.

The results show that the companion robot successfully reduced the amount of social isolation and loneliness felt by the students it was deployed with. The robot exhibited both socially engaging and autonomous behaviour, and the behaviour was personalised to each individual user.

Sammanfattning

Ensamhet och social isolation är på uppgång, särskilt bland unga, vilket kan resultera i till exempel mental ohälsa och dylik problematik. Denna utveckling har endast förvärrats av karantän och distansarbete, som tvingats fram av den pandemiska situation som följt COVID-19. Målet med detta arbete är därför att utveckla en sällskapsrobot som kan underlätta denna ensamhet och sociala isolering. Genom Wizard of Oz-experiment fångas robotens ideala beteende. En människa ges full kontroll över roboten i interaktion med en användare, och det beteende som denna människa väljer för roboten registreras och används för att konstruera en autonom robot. Denna autonoma konstruktion bygger på två klassifikationsmodeller inom maskininlärning. Information som samlas in dels från miljön, dels från användarbeteendet, kombineras för att bygga upp en tillståndsrymd. Givet en sådan tillståndsrymd avgör den första modellen huruvida en handling ska utföras, varefter den andra modellen avgör vilken handling som ska utföras. Från resultaten framkommer det att den framtagna sällskapsroboten lyckades minska användaruppfattade känslor av ensamhet och social isolering. Roboten påvisade autonomt personanpassat och socialt engagerande beteende.

Acknowledgments

I would like to thank my supervisors, Katie Winkle and Sarah Gillet, for all the guidance and support they provided throughout this project. By sharing their HRI expertise with me, I gained a far better understanding of this research area. I would also like to thank the HRI department at KTH Royal Institute of Technology for lending me all the equipment needed throughout this project. Lastly, I would like to thank all the participants that helped make this research possible.

Contents

1 Introduction **1**
1.1 Problem . . . 1
1.2 Goals . . . 2
1.3 Research Methodology . . . 2
1.4 Structure of the Thesis . . . 3

2 Background **4**
2.1 Social Robotics . . . 4
2.1.1 Mutual Shaping . . . 4
2.2 Wizard-Of-Oz Framework . . . 5
2.3 Machine Learning . . . 6
2.3.1 K-Nearest Neighbour . . . 7
2.3.2 Decision Tree and Random Forest . . . 7
2.3.3 Support Vector Machine . . . 8
2.3.4 Multi Layer Perceptron . . . 8
2.3.5 Imbalanced Data . . . 9
2.4 Related Works . . . 10
2.4.1 Social Robotics . . . 10
2.4.2 From Wizard of Oz Control to Human in the Loop Learning . . . 12
2.4.3 Research Contribution . . . 15

3 Methodology **16**
3.1 Mutual Shaping, Robot Design and Action Space . . . 17
3.2 State Space and ROS Architecture . . . 21
3.3 Data Collection . . . 26
3.4 Model Selection . . . 30
3.5 Final Deployment . . . 39

4 Results **41**

5 Discussion **52**
5.1 Develop a robot that provide social support and reduces student loneliness and social isolation 52
5.2 The robot should exhibit both socially engaging and autonomous behaviour 53
5.3 The robot should exhibit behaviour that is personalised to each individual user 54

6 Conclusions and Future Work **55**
6.1 Conclusions 55
6.2 Future Work 56

References **58**

A Interview Participant Demographics **64**

B Deployment Questionnaires **65**

Chapter 1

Introduction

1.1 Problem

Feelings of loneliness and social isolation are on the rise and reaching record levels, especially amongst young people. Loneliness can be described as feeling alone and disconnected from others, while social isolation is an objective lack of social connections and interactions with others [1]. Experiencing these feelings in higher rates is linked to an increased risk of depression, anxiety, suicide, heart disease, and many other conditions [2]. Reducing the amounts of loneliness and social isolation an individual experiences will enhance their overall wellbeing and help prevent these conditions from developing. Young adults, especially those living alone in apartments and student dormitories, are currently experiencing very high levels of loneliness and social isolation. Several factors are responsible for this, including the rise of social media, moving away from home for the first time, and spending long periods of time alone in their rooms studying or working [3]. The situation has only been made worse by the COVID-19 pandemic. Social distancing and lockdown restrictions were implemented across the world to help prevent the spread of the virus. University classes switched to online teaching and employers switched to remote working. People were restricted by the number of people they could meet, and in some cases they were prevented from leaving their homes altogether. This all resulted in a major boost in feelings of social isolation and loneliness, which took a toll on many young people's mental health [4].

One major research area in the field of Human Robot Interaction (HRI) is the development of social robots. Whilst before robots were mostly valued for their practical capabilities, the HRI community has realized their potential when it comes to providing social support and companionship for those who may need it [5]. Research has shifted from focusing on functionality to improving well-being and reducing negative emotions [6]. Previous work has demonstrated social robots might be able to alleviate some of these issues induced by the pandemic, but there are many challenges and unanswered questions regarding the development and deployment of autonomous such systems.

1.2 Goals

The main goal of this research is to develop a companion robot that provides social support and reduces feelings of social isolation and loneliness felt by students in their dorm rooms. This can be divided into three main objectives-

1. Develop a robot that provides social support and reduces student loneliness and social isolation
2. The robot should exhibit both socially engaging and autonomous behaviour
3. The robot should exhibit behaviour that is personalised to each individual user

By successfully achieving these objectives and deploying the robot into a student's dorm room, the likelihood of developing any of the conditions associated with high levels of social isolation and loneliness would be significantly reduced. The student's overall wellbeing would also improve, and it would also help them better deal with the isolating effects of the COVID-19 pandemic, such as quarantining and remote learning. Therefore, this project meets the UN Sustainable Development Goal of "Good Health and Wellness."

1.3 Research Methodology

Based on previous research, conducting a Wizard-Of-Oz style study [7] seems like a promising method to develop a robot that exhibits personalised, socially engaging, and autonomous behaviour. The robot's behaviour will be modelled around a human. By allowing the human to take full control of the robot and deploying it with a user, the actions and behaviour the human chooses for the robot can be recorded and later used to create autonomous behaviour for the robot. The human is expected to control the robot and have it interact with a user in a socially engaging manner, by having it behave in a way that they think is best for the specific user they are controlling the robot for. By modelling the robot's behaviour off these interactions chosen by the human, we expect to create autonomous behaviour that is both socially engaging and personalised to each user.

Two machine learning classification models are required to control the autonomous behaviour of the robot. All statespaces and actions from the Wizard-Of-Oz data collection sessions will be saved and used as training data for the machine learning algorithms to model this behaviour. When presented with a statespace, the first model will determine whether an action should be performed, or whether "DoNothing" is the most appropriate action. The second model will map these statespaces to the correct action.

1.4 Structure of the Thesis

The thesis observes the following structure. Chapter 2 presents all the relevant background information and research on Social Robotics, Mutual Shaping, the Wizard-Of-Oz Framework, and Machine Learning classification algorithms, as well as the different approaches to dealing with imbalanced datasets. This chapter also presents relevant related works that have been completed by other researchers. Chapter 3 presents the methodology and methods that were used to achieve the goals of the thesis. This chapter is split into sections that outline the Mutual Shaping robot design, the ROS architecture and state spaces, the Wizard-Of-Oz data collection sessions, the Machine Learning model selection and training, and the final autonomous robot deployment. Chapter 4 presents and analyses the results obtained from the deployment of the autonomous robot into student dorm rooms. Chapter 5 presents a discussion of the results in the context of the aims and objectives of the study to determine whether they have been successfully achieved. Chapter 6 presents the conclusions that can be drawn from this research, and future work that could be performed.

Chapter 2

Background

2.1 Social Robotics

The main goal of social robots is to interact with humans in some form of socially meaningful way. Companion robots are a type of social robot that are capable of social behavior through effective and appropriate interactions with a user in their home environment [8]. Through these interactions they can improve the wellbeing of users that are experiencing social isolation and significantly decrease the chance of them feeling lonely. Companion robots can have the same positive physiological effects as pets, in that they can improve brain functionality and reduce levels of stress, whilst requiring much less maintenance and commitment. A social relationship develops between the robot and the user, resulting in decreased feelings of negative emotions, isolation, and loneliness [9].

2.1.1 Mutual Shaping

The application of social robots can also be beneficial outside the context of providing companionship and reducing loneliness. Winkle et al [10] present a "mutual shaping" approach to designing social robots. They then use this approach to effectively design a robot that can be used as a therapy aid. Following a mutual learning framework increases the chance of the robot being accepted by all stakeholders, and therefore being fully utilised, since they will gain a much broader understanding of its use. It will also maximise the benefit the stakeholders will likely gain from using the robot, as well as help identify additional factors that could affect the deployment of the robot. The main concept of mutual shaping is involving the user in the design process and the early development stages. Users are usually not consulted at these points. The results of this research show that the users involved in the mutual shaping process had a higher acceptance rate, and a higher positive response, to the therapy robot. They were also generally quite enthusiastic and keen to be part of this process.

The mutual shaping process can be effectively integrated into a human in the loop approach. It allows for the mutual shaping process to be carried out even at the automation stage, and allows users to be fully immersed in creating the autonomous robot behaviours.

2.2 Wizard-Of-Oz Framework

First introduced by Dr Kelley in 1984 [7], the Wizard-Of-Oz robot control framework is one of the most widely accepted and utilized techniques in the Human Robotics Interaction (HRI) community. It has been adapted to many different fields including linguistics and experimental psychology, however is now typically used for testing robot behaviours for behavioural studies that are currently too hard to make autonomous. Although the approach can slightly vary depending on the exact task it is being applied to, the general principles outlining its implementation remain the same. A “wizard” (it could be the experimenter, an expert, another participant etc) remotely operates the robot for a participant. The wizard may control various actions and behaviors of the robot, for example its speech, movement, gestures, and many others. The amount of control the wizard has over the robot could lie anywhere on the autonomy spectrum, from full tele-operation and controlling all the robots’ actions, to full autonomy and having no control over the robot at all. Sometimes the participant is given details about the wizard and how the robot will be operated, but usually they are purposefully kept unaware of the wizard and assume the robot is behaving autonomously. This low-level deception encourages more natural behavior from the participant, whilst also managing their expectations. The wizard either operates the robot in the same room as the participant or will tele-operate it from a separate space. A Wizard-Of-Oz study could be performed in a laboratory setting, or in the real environment that the robot would be deployed into. The way in which the wizard is instructed to operate the robot depends heavily on the task and the desired behavior of the autonomous robot.

A researcher will often choose a Wizard-Of-Oz approach when there is no robot available that can perform the type of socially competent, safe, and autonomous behavior that they desire. This may be due to a lack of advancement in the HRI field, but can also be the case in the early stages of a study, when only a small amount of the implementation has been completed. It allows the researcher to effectively envision the behavior that can be achieved by the robot. It can also be used as part of an iterative design process, allowing researchers to experiment with and evaluate different versions of their robot. Although widely accepted as a valid and effective technique in the HRI community, criticisms of the Wizard-Of-Oz approach do exist. Weiss [11] discusses how the robot being controlled cannot be considered an independent entity, but rather as a human proxy. This could actually be a desirable aspect of the technique depending on the research goal, however for research that specifically aims at investigating human-robot interaction, it can be seen more as human-human interaction. Fraser and Gilbert [12] have stressed the importance of the ethical concerns that come with using this technique. When a previously unaware participant is told that another human has been controlling the actions of the robot, they can be left with feelings of embarrassment and gullibility over the fact that they have been deceived.

A Wizard-Of-Oz style study is the ideal technique for modelling a robot's behavior around a human. By allowing the human to take full control of the robot and deploying it with the participant, the actions and behavior the human chooses for the robot can be recorded and later used to create autonomous behavior for the robot. This can be achieved by implementing an appropriate machine learning algorithm which will learn how the human controlled the robot, and will be able to replicate this behavior without any human intervention, thus creating effective autonomous behavior.

2.3 Machine Learning

Previous research has highlighted the effectiveness of machine learning classification algorithms at controlling the autonomous behaviour of robots. HRI researchers have implemented various classification algorithms for different tasks, with each achieving an impressive performance [13] [14] [15] [16] [17]. The algorithms outlined in this section have all been successfully used in the HRI domain to train social robots. Therefore, they could also be utilised in this research. The suitability of an algorithm to a problem depends on the specific attributes of the problem (such as the amount of training time that is allowed), the type of data, the amount of data, the number of classes, and the number of features [18] [19]. All of these factors will be considered when determining the best algorithms to implement for this research.

2.3.1 K-Nearest Neighbour

The K-Nearest Neighbour (KNN) algorithm [20] [21] classifies data based on how its neighbours have been classified. A similarity measure is used to determine the K neighbours of a new datapoint, which is then assigned its class based on the majority class of these K neighbours. K is a parameter that can initially be calculated using an equation, and then have its value updated using an optimization technique. The KNN algorithm is known as a lazy learner, which means is does not have a training phase. It learns from the data only when classifying a new datapoint. It is also one of the easiest algorithms to both understand and implement. However it is not well suited to large datasets, since adding a new datapoint requires the algorithm to calculate the distance with every existing point. The algorithm is also very sensitive to noisy data and outliers, and its performance relies heavily on a good K parameter being chosen. Although this algorithm returns an impressive performance when implemented on different problems, it is usually outperformed by more advanced techniques such as neural networks. KNNs are a popular algorithm choice for training the behaviour of robots and determining which actions they should perform, especially when the learning is performed online and not after all the data has already been collected. Senft et al [13] implement a KNN to train autonomous robot tutors for children, where all the learning is carried out online, while the robots were deployed. Winkle et al [14] use a KNN in an interactive machine learning approach to effectively train a robot as a personal trainer.

2.3.2 Decision Tree and Random Forest

During the training phase, the decision tree [22] [23] algorithm learns decision rules in order to create a tree like structure, which it then uses to classify unseen datapoints. Each decision node will split into two or more sub nodes, based on a specific condition. Each possible edge that can be taken from a decision node represents a possible answer to the condition. To classify a new datapoint, it follows the path down the tree based on the conditions at each node, until eventually reaching a terminal node which will assign it the correct class label. Decision trees are easy to visualize and understand, and are robust to outliers. However, they have large training time when the dataset is large, and they are very susceptible to overfitting. Random forests can be used to overcome this problem. By using a technique called bagging, random forests combine the output of multiple decision trees, which reduces the likelihood of ovefitting occuring. The training time for random forests is even longer, but they do usually perform well at classifying unseen data. Chavez et al [15] introduce a visual method for diver detection in the context of HRI. They implement a variant of random forests to calculate the region surrounding the diver's body, which can be effectively utilised by underwater robots.

2.3.3 Support Vector Machine

In the training phase of the Support Vector Machine (SVM) [24] [25] algorithm, each datapoint is plotted in the feature space, with the coordinate values corresponding to the feature values. A hyperplane is constructed that separates the classes, and a new datapoint is classified based on where it is plotted in relation to the hyperplane. SVMs take advantage of the kernel trick, which uses a kernel (a function) to transform the low dimensional input space to a higher dimensional space, where the data can be much more easily separated. This means SVMs can be used to classify data that would otherwise be very difficult to separate in the lower dimensional input space. SVMs are more suited to classifying smaller, very high dimensional datasets. They have a long training time, but are very efficient at classifying new datapoints. The performance of SVMs is very dependent on the correct kernel function and hyperparameters being used, so optimization is key. Using a simple discriminative SVM classifier, Chu et al [16] develop robots that can effectively recognise if a person is positively responding to the robot's interaction request.

2.3.4 Multi Layer Perceptron

A Multi Layer Perceptron (MLP) [26] [27] is made up of three types of layers. For classification tasks, the input layer receives the input data, and the output layer will determine which class the input should be assigned to. Between these two layers are a number of hidden layers. The parameters (weights and biases) of these layers are trained using the backpropogation algorithm in order to minimize some error function. MLPs are capable of approximating any continuous function and can solve non-linearly separable problems. MLPs generally outperform other classification methods when it comes to more complex tasks, however they have one of the longest training times. They also require a lot of training data and are prone to overfitting. Romeo et al [17] train a neural network that determines which user, in a room full of people, a robot should interact with. The robot is trained to assess the user that is most willing to interact by using the beginning of its interaction with users to determine their level of engagement.

2.3.5 Imbalanced Data

An imbalanced dataset refers to when the class distributions are not equal, or even similar. This means there are a different number of samples from each class. The minority class (the class that has only a small proportion of the samples) becomes harder to predict. It is harder for the model to learn the characteristics of this class and differentiate it from the other classes. The model may only learn the characteristics of the majority classes (the classes that hold a large proportion of the samples) and completely neglects the minority class, even if predictions of this class are a lot more valuable. Different techniques have been proposed to deal with an imbalanced dataset and overcome the challenges they present [28] [29].

Oversampling

Oversampling involves increasing the number of samples in the minority class. The simplest way this can be achieved is by duplicating samples. Although this will successfully even out the distribution, these new samples do not add any new information to the model. To address this drawback, another method to perform oversampling is to synthesize new samples from the existing ones. The most widely used approach for this type of oversampling is called SMOTE (Synthetic Minority Oversampling Technique) [30]. SMOTE works by randomly selecting a sample from the minority class and then finding the k closest samples to it. One of these k samples is then randomly selected, and the new synthetic sample is created by choosing a random point in the feature space between these two samples. It is an effective approach because it creates samples that are close in the feature space to pre-existing samples. He et al [31] present a modification to SMOTE called Adaptive Synthetic Sampling (ADASYN). It involves generating a lower proportion of synthetic samples in regions of the feature space where there a more samples, and a higher proportion of synthetic samples in regions of the feature space where there a less samples. Synthetic samples are generated adaptively based on the distribution of the samples in the minority class, rather than randomly.

Therefore, more synthetic samples will be generated for existing samples that are harder to learn compared to those generated for easier to learn samples.

Undersampling

Undersampling involves decreasing the number of samples in the majority class. The simplest way this can be achieved is by randomly removing samples from the majority class. Although this will successfully even out the distribution, it does not consider how important these removed samples are when determining the decision boundaries. Therefore, multiple methods have been developed that help determine which samples should be kept and which samples should be deleted. One Sided Selection for Undersampling [32] is a combination of two methods. First, the Tomek Links [33] method is used to remove borderline and noisy samples from the majority class. Then, the Condensed Nearest Neighbor [34] method is used to remove samples from the majority class that are far from the decision border.

Chawla et al [30] suggest that using a combination of undersampling the majority class and oversampling the minority class is the best way to handle an imbalanced dataset.

2.4 Related Works

2.4.1 Social Robotics

Exploring the potential benefits of robots and how they can be utilised to provide social support for humans has become an increasingly popular research topic over the past few years, especially due to the rise in feelings of loneliness and isolation during the COVID-19 pandemic.

During the beginning of the pandemic, Tsoi et al [35] recognised the negative effects that quarantining at home could have on children. To address this problem, they developed VectorConnect, a robot teleportation system that allowed one child to control the Vector robot of another child in a separate location. VectorConnect allowed the two children to securely video chat and engage with eachother, whilst staying safely isolated in their homes. This research shows that robots can be used in effective and creative ways to reduce the social isolation felt by children during the pandemic. They also outline the issues they faced in regards to recruitment of participants and distribution of robots in the midst of the pandemic, as well as privacy and security measures that had to be enforced to protect the users and their data.

Jecker [36] identified that older people suffer from much higher rates of loneliness and social isolation compared to the general population, which would only get worse during a pandemic. This can have a detrimental effects on their health and wellbeing. She suggests that social robots could be used to overcome this. Jecker proposes that the robot should be as life- like as possible, and could ideally touch and have physical contact with an older user without causing injury. The robot should also use sensors to respond to external stimuli and have some human traits in its appearance. Together, these characteristics will encourage the user to develop a significant relationship with the robot, and will therefore decrease feelings of loneliness and isolation.

Schröder et al [37] investigated whether owning a Vector robot during the COVID-19 pandemic resulted in a reduced feeling of loneliness. This was achieved by collecting publicly available online data from blogs, social media and customer reviews. The data was analysed and interpreted to understand the role that the Vector robot played in people's lives during the pandemic. It was found that each user gave Vector the role of either a personal assistant, relational peer, or intimate buddy. Each role created a slightly different relationship between Vector and the user, but all were capable of reducing the loneliness felt by the user in some way.

Research has also been conducted into different ways that the relationship between social robots and humans can be improved [38]. Trust, disclosure, and a sense of companionship are three factors that are known to improve interactions between humans and robots. Martelaro et al [38] aimed to achieve and increase these three factors by investigating the effects that expressivity and vulnerability have on them. These two attributes were tested on students by using robots as tutors. It was found that increasing the robot's vulnerability would increase trust and companionship, whereas increasing expressivity would increase disclosure. These findings could be useful when designing the behaviour of future social robots.

Using social robots to tackle problems relating to social isolation and loneliness has also been investigated outside the context of the recent pandemic. Jeong et al [39] introduce Fribo, a social networking robot that identifies and shares auditory information with close friends of the user. It was found that an increasing number of young people are choosing to live alone, which increases feelings of social isolation. Silence was identified as the biggest contribution to the feeling of loneliness when living alone. Fribo shares the living noise recorded from one user's house with another user. It made the users feel closer to each other, thus reducing feelings of loneliness. Privacy concerns were more easily dealt with because only audio information was shared, however participants still had concerns about what exactly was being shared, and when exactly Fribo was recording.

Similar to [36], Hudson et al [40] identified the increased rates of loneliness experienced by older people. Their solution to this was to provide them with a robot pet. The pets contained many sensors which allowed them to detect when they were being touched. They had both vocal (eg barking) and non-verbal (eg head movements) responses to this. The participants had a mixed response to the pets given to them. Some fully embraced the pet as a companion, resulting in reduced feelings of loneliness. However, a big criticism from some users was that they didn't view the pet as realistic, so could not fully connect to it. It was found that the people that benefited most from having a robot companion shared some similar attributes- they were less active, lived alone, and had fewer social connections.

Lui et al [41] designed a robot that aims to boost the mood of a user. The robot uses a camera to capture the face of the user and a microphone to capture the voice. The image of the face and the voice of the user are then analysed to determine which emotions they are portraying. An Artificial Neural Network (ANN) uses this input information to classify the overall mood of the user. Different outputs would be triggered depending on the identified mood, with the overall goal of improving it. Outputs include playing music through a speaker, releasing different smells from attached perfume bottles, moving around using a motor, and displaying different colours using LED lights. However, the robot was never deployed or tested, so it is unclear how successful this robot would have been at enhancing users' moods.

2.4.2 From Wizard of Oz Control to Human in the Loop Learning

The active involvement of an expert during the learning phase of an algorithm can significantly improve the quality and the speed of the learning. This has encouraged many researchers to adopt either a human in-the-loop approach, a

Wizard-Of-Oz approach, or a combination of both methods, and apply them to a broad range of problems.

Senft et al [13] implement and expand on SPARC (Supervised Progressively Autonomous Robot Competencies) to train autonomous robot tutors for children. At first, a Wizard-of-Oz style mechanism is used to control the robot, where a human (a tutor) will determine all the actions performed by the robot, and at what times they are performed. These actions then become training data for the algorithm, which it then uses to suggest actions it thinks are correct, given a set of inputs. These actions are approved or rejected by the tutor, which then add to the collection of training data used by the algorithm. The tutor can also passively accept actions simply by not rejecting them, which should reduce their workload. This method is different from previously developed ones because the learning is done online. This type of in-situ learning is performed in the real world, not after all the data has already been collected. A K-Nearest Neighbour (KNN) algorithm was used to determine which action would be suggested, given a set of inputs. The learning was a success, as the robot clearly demonstrated social behaviour that supported the children in their education. The robot learned this in only a few sessions, which shows that this technique can be used to quickly teach autonomous and domain-specific behaviour to social robots.

Winkle et al [14] adopted an interactive machine learning approach, based on SPARC [13], to effectively train a robot as a personal trainer. Human in-the-loop learning is well suited to this task, since the trainer (the expert) in charge of validating all the robot's actions can make well informed decisions not only based on how the participants are currently performing, but also based on his previous knowledge from having training sessions with them. During the learning phase the trainer can either determine and execute an action he deems appropriate, or approve/ reject an action suggested by the algorithm. All actions suggested by the trainer, and all accepted actions from the algorithm, were then used as additional training data for the algorithm. This method requires a simple system control architecture, made up of nodes communicating through the Robot Operating System (ROS). A K-Nearest Neighbour (KNN) algorithm was used to determine which action would be suggested, given a set of inputs. The input space consisted of 20 different states, including information about each participants personality. Using this method, they were able to successfully train robots to become fully autonomous, personalised, and engaging trainers. The algorithm was successful in suggesting the correct personalised task, however it did not fully learn when to deliver this tasks at the best times. This may have been due to the KNN algorithm that was chosen for the learning.

Li et al [42] explored using human in-the-loop online learning to improve the performance of conversational bots. Instead of using KNN, they investigate using three other learning algorithms. The first algorithm is Reward-Based Imitation (RBI), which learns to predict the correct output at a specific training time. The second algorithm is REINFORCE, which works by maximising the expected reward. The third algorithm is Forward Prediction (FP), which is used when feedback by the human is given in the form of text, rather than numerical rewards. In the experiments both RBI and FP perform well, while the performance of REINFORCE relied on finding the optimal ϵ value.

Sequeira et al [43] propose a three-phased methodology to model social interaction between human and robots based on the restricted-perception Wizard of Oz approach. When controlling the robot, they suggest that the expert should only have access to the same information that the robot would, as opposed to them being able to observe the whole scenario. Considering the robot's behavioral and perceptual limitations ensures that only the most meaningful information is extracted from the study, which will result in better autonomous behavior. In the Data Collection phase, mock-up interaction scenarios are performed in which an expert performs the task with the user. Using this information, the robot's task specific perceptions and actions can be devised, which are implemented in the form of a "Task AI." This is then used by the expert to control the actions of the robot in sessions with the user, using the restricted-perception (also known as "unrestricted") Wizard of Oz approach. In the Strategy Extraction phase, a hybrid robot controller system is built using the data collected from these sessions. The hybrid system consists of a rule-based module that encodes specific strategies and behavior patterns identified in the Data Collection phase, and a machine learning module responsible for the autonomous behavior of the robot, determining which actions the robot should perform, and at which times they should be executed. In the Strategy Extraction phase, the autonomous robot controlled by the hybrid system is deployed and evaluated. The robot's behavior is then refined in an iterative process. This three-phased methodology was tested and implemented in the context of a robot tutor. By following the outlined steps, a fully autonomous robot tutor was created that was able to successfully interact with students and display empathic, collaborative and social behavior.

2.4.3 Research Contribution

This research is different from related works in several ways. Two classification algorithms are required to control the autonomous behaviour of the robot in this research, unlike in previous research where they only used one. When presented with a state space, the first classification algorithm will determine whether an action should be performed, or whether “DoNothing” is the most appropriate action. The decision to implement this first classifier is based on issues faced in previous work [13] [14]. Researchers have had difficulty with a robot learning whether an action should be performed or not. Usually only one classifier is trained that learns which action should be performed, and “DoNothing” is added as one of the actions in the action space. Training a completely separate classifier whose sole purpose is to learn this task should help overcome this difficulty. The second classification algorithm will then map these state spaces to the correct action.

Another aspect of this research that makes it different from other work is that the Wizard-of-Oz technique has previously not been used like this in a human in the loop approach. In this research, learning is implemented from these Wizard-of-Oz sessions. Also, students will act as both the wizard and the user, and they will be doing this with their peers. In previous research, the same person has not acted as both a wizard and a user, and the wizard has never been the users’ peer.

This research also follows an end-to-end mutual shaping approach. The main concept of mutual shaping is involving the user in the design process and all throughout the development stages of the robot. In this research the user is involved throughout the whole process. The features of the robot are co-designed with the users using interviews, the users are then involved in training the robots, and finally the autonomous robot is deployed with those same users.

Chapter 3

Methodology

The overall aim of this research is to reduce feelings of social isolation and loneliness felt by students. Based on the analysis performed in the previous section, the best way to achieve this is by developing a companion robot that the user can interact with in their dorm room. Multiple researchers have demonstrated the effectiveness of companion robots at improving the well being of users, whilst also decreasing the isolation, loneliness, and negative emotions felt by them [35] [36] [37] [39] [40] [41]. The companion robot must be capable of exhibiting both socially engaging and autonomous behaviour. The behaviour must also be personalised to each individual user. Based on the research, this can be accomplished by conducting a Wizard-Of-Oz style study [7], through which training data can be collected for training an autonomous machine learning agent

In order to follow a mutual shaping framework, interviews will be conducted with potential users. This will give users the opportunity to express their ideas and opinions about companion robots, and will help in identifying which features the robot should have. Once the specifications and the features of the robot have been finalised, the ROS architecture needs to be designed and built. This will be used to manage all the equipment, save any necessary data, and control the actions of the robot. Next, the Wizard-Of-Oz training sessions can begin, where a human will control the actions of the robot for a user. All statespaces and actions will be saved and used as training data for the machine learning algorithms to model this behaviour. The most appropriate machine learning algorithms must then be identified, and the models will be trained. Finally, the robot can be deployed in the rooms of the users, using the trained machine learning models to autonomously determine their behaviour. This behaviour can then be analysed and evaluated.

3.1 Mutual Shaping, Robot Design and Action Space

Based on the research, the best way to design the robot and determine its actions and attributes is by following a mutual shaping user-centred approach [10]. As discussed above, the main benefits of this approach include getting ideas directly from the user about exactly what they want from the robot, and increasing the chance of the robot being fully utilised by the user since they are involved in the early design stage. Winkle et al [10] present a methodology that outlines how mutual shaping can be used during the design stage of a social robot in an initial interview. They describe the format of this interview, and what type of questions should be asked at each stage of it. Based on this methodology, an interview was designed that would be performed on 8 participants. The demographics of the participants can be found in the Appendix section. Each participant fits the target criteria of being a student that lives alone in a student dorm room, so they could all be a potential end user of the robot. Each interview included a short demonstration of the robot and the actions it can perform. The interview aims to help establish the features of the robot, how users respond to the robot, and exactly how they would use the robot. The format of the interview, including details of the sessions and the exact questions that were asked, can be found below.

Before Session
Consent Form
Collect demographics- Age, Gender, Nationality
Begin audio recording

Pre-Demo Discussion
Initial question to make them feel more comfortable- How have you customised the space in your room to make it feel more homely? (Ask for a drawing)
Do you ever feel lonely when you are in your room? What are some methods/activities you use to reduce this feeling?

Briefly introduce Cozmo

How do you think Cozmo could be integrated into your apartment?

Do you think Cozmo could improve the experience of (insert activity they previously mentioned doing in their room)? If they do not mention anything, be general and ask- How do you think Cozmo could improve time spent alone in the room?
What features would you like Cozmo to have?

Demonstration
Demo of Cozmo performing some tricks, playing a game, showing different emotions, and speaking some phrases. Explain the different behaviour approaches that can control Cozmo

Post-Demo Discussion
Repeat some questions from Pre-Demo

How do you think Cozmo could be integrated into your apartment?
Do you think Cozmo could improve the experience of (insert activity they previously mentioned doing in their room)? If they do not mention anything, be general and ask- How do you think Cozmo could improve time spent alone in the room?
What features would you like Cozmo to have?

Then ask some new questions-
How would you like Cozmo to behave? Would you prefer if Cozmo acted autonomously, would you prefer to control Cozmo, or if someone else controlled your Cozmo?
If Cozmo saw you were upset, how would you want it to respond? (eg would you like it to behave upset too/ try to cheer you up/ would you want to control it etc)
Would you feel comfortable if Cozmo had a camera and a microphone to capture image and audio data?

End audio recording

After analysing the 8 participant's responses from the interviews, the attributes of the robot, as well as what the users wanted from the it, became a lot clearer. The main findings from the interviews were-

- Many participants admitted feelings of loneliness when they were alone in their dorm room, and thought a companion robot could help with that. They all preferred the robot to behave autonomously rather than have its actions controlled by a person. This further consolidated the main aims of this research- to create an autonomously behaving robot that reduces user loneliness in their rooms.

- Most participants did not want the robot active for the whole day, and instead wanted a “Do Not Disturb” mode where they could pause its actions. This will be incorporated into the final robot design.
- During the demonstration, participants were unsure which action the robot was performing when they were required to interact with it. Therefore, the final robot will “announce” some actions, so the user can respond appropriately.
- Some participants were slightly uncomfortable with a camera recording their facial expressions. They all agreed that it would be acceptable as long as the videos were not saved.
- One participant suggested that the robot could give wellbeing reminders throughout the day, such as drinking water and going for walks. This will be incorporated into the final robot design.
- It is clear from the interviews which robot actions the participants preferred and which ones they did not.

Based on the research, the interviews, and the equipment available, the design and action space of the robot can be finalised. The robot used in this research will be a Cozmo robot, developed by the Anki robotics company [44]. However, the architecture built for this research can be used with any robot. It is possible to control the actions of Cozmo from a laptop by using a Python SDK. Cozmo can be put into SDK mode using the Anki app on a tablet. Once the tablet is then connected to a laptop, python programs run on the laptop can directly connect to Cozmo, and the robot will execute the programmed actions.

Below is a table containing the finalised action space, outlining the name of each of the 17 actions and a brief description of their execution. Some actions can be demonstrated in several different ways. For example, when “Happy” is executed, an action is randomly selected and executed out of five possible actions that demonstrate that Cozmo is happy. A combination of the below actions can be successfully used to create socially engaging behaviour for the user.

Action	**Description**	**Options**
Happy	Happy emotion demonstrated	5
Sad	Sad emotion demonstrated	5
Angry	Angry emotion demonstrated	5
Shocked	Shocked emotion demonstrated	3
Monster	Acts like a zombie or vampire	2
Animal	Acts like either a dog, cat, snake, elephant, tiger, or chicken	6
Hyperactive	Acts very hyperactive, moves around a lot, makes lots of noise	2
Ill	Acts ill and sick	3
Song	Sings a short song	3
Joke	First announces it has a joke to say. Then says the joke e.g., “What do you call a bear without any teeth? A gummy bear!”	25
Reminder	First announces it has a reminder to say. Then says a reminder to improve the wellbeing of the user e.g, “Remember to stretch your legs if you’ve been sitting down for too long!”	5
Hand Interaction	First asks user to move finger around in front of them. Then Cozmo will pounce on finger	1
Game	First asks user if they want to play a game. Then will run through a game of “Quick Tap”	1
Short Bricks	First looks for its cubes. If it cannot find them, asks user to help find its cubes. Then will either roll a cube, or do a wheelie	2
Long Bricks	First looks for its cubes. If it cannot find them, asks user to help find its cubes. Then will stack two cubes on top of each other. If completed successfully, Cozmo will celebrate, then knock the cubes over	1
Background	Performs background activities, such as looking around, raising and lowering the lift, and moving in a square	3
Charger	Asks the user to place it on the charger	1

Table 3.1 – Cozmo Action Space

3.2 State Space and ROS Architecture

The aims of the machine learning models are to classify each statespace into whether an action should be performed or not, and then classify the statespace to the correct action, given that one should be performed. Therefore, choosing the information that makes up the statespace is crucial, because it will be used to determine all the actions performed by the robot. After trialling different features of the user and the environment, the finalised state space can be found in the table below.

State Name	**Description**	**Options**
Action	The action that was performed when this state is saved. If no action is performed, it is saved as "0"	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
Last Action	The last action that was performed	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
Time Since Last Action	The time, in minutes, that has passed since Cozmo last performed an action	
Time Since Last Charge	The time, in minutes, that has passed since Cozmo was last on the charger	
Webcam Mood	The mood of the user. Determined by taking the stream from the webcam as input, and classifying it using a mood classifying model	Neutral, Happy, Sad, Angry, Surprised, Disgust, Fear
Webcam Gaze	Shows whether the user is facing Cozmo or not. The webcam is placed near Cozmo. If the user is facing Cozmo (and therefore the camera), there is a high chance the webcam mood classifier will be able to classify the mood. So if the mood is successfully classified, the Gaze will be "1". If the user is facing away from Cozmo (and therefore the camera), there is a low chance the webcam mood classifier will be able to classify the mood. So if the mood is unsuccessfully classified, the Gaze will be "0".	0, 1
Microphone Volume	The volume detected. Determined by extracting the volume from the microphone on the webcam. It is then classified into one of three categories	None, Low, High
Input Mood	The mood that is selected by the user. The user is presented with a laptop with buttons on the screen. They can click on the button that best represents their current mood	Positive, Neutral, Negative
Input Activity	The activity that is selected by the user. The user is presented with a laptop with buttons on the screen. They can click on the button that best represents their current activity	Studying, Relaxing, Eating, Do Not Disturb

Table 3.2 – Variables of the Statespace

In order for the mood detector model to successfully classify a camera stream, the image needs to be of a high quality. Although Cozmo does have a camera, the images taken from it are of a very poor quality, and the mood detector model was unable to classify any images from it. Therefore, a webcam will be used instead. It has a much higher image quality than the Cozmo camera, and can be successfully classified by the mood detector model. Participants did state in the interview that they were comfortable with a webcam in the room if the video stream from it was not being saved. Also, a microphone is needed to measure the volume of the room. Cozmo does not have a microphone, so the microphone from the webcam must be used.

Although the webcam and the mood detector model will capture the mood of the user, it is important to get some information directly from the user. If it comes from the user, it is much more likely to be accurate. Therefore, a laptop will be set up in the user's room. The laptop has buttons on the screen that represented different moods and activities that could represent how they user is feeling and what the user is doing. This direct information will be very useful for the machine learning models.

An architecture is required that facilitates effective transmission and communication between all the equipment, data, and models. The best platform to build this architecture on is the Robot Operating System (ROS) [45]. ROS is a collection of software libraries and tools that is designed to effectively build robot applications. It is run on the Linux operating system. The architecture will be built on the ROS Noetic distribution [46]). Several ROS nodes will be set up. The role of each ROS node is described in the table below. The State Manager node is used in the data collection stage, while the State Manager 2 node is used in the final deployment stage. An outline of the communication between the nodes, including the data each node sends and receives, can be found in the two node diagrams below. The first node diagram outlines the ROS architecture used for the data collection stage, where a human is selecting the actions for Cozmo. The second node diagram outlines the ROS architecture used during the final deployment stage, where the machine learning models determine the actions that Cozmo executes, and therefore exhibits autonomous behaviour.

Node	Description
State Manager (used during Data Collection Stage)	Receives all the data from all the nodes. Also calculates the last action, the time, the time since the action, and the time since the last charge. Every 5Hz, the latest data received from all the nodes is combined and saved into a csv file to make up the normal statespaces. Every time an action is performed, the latest data received from all the nodes is combined and saved into a csv file to make up the action statespaces.
State Manager 2 (used during Final Deployment Stage to autonomously control the robot)	Receives all the data from all the nodes. Also calculates the last action, the time, the time since the action, and the time since the last charge. Every 5Hz, the latest data received from all the nodes is combined to make up a statespace. The preprocessing steps mentioned in the “Model Selection” section are performed on the statespace, including applying the same Standard Scalar to normalise the data to fit the first model. The normalised statespace is then given as an input into the first model, which will determine whether an action should be performed or not. If the statespace is classified into the “Do Nothing” class, it is saved into a csv file to make up the normal statespaces. If the statespace is classified into the “Action” class, the preprocessing steps mentioned in the “Model Selection” section are performed on the original statespace, including applying the same Standard Scalar to normalise the data to fit the second model, and it is then given as an input into the second model, which will determine which action should be performed. It will then send this action to the “Listener” node so it can be executed. The statespace is then saved into a csv file to make up the action statespaces. The node will then sleep for a certain amount of time until the action has finished being executed. This prevents lots of actions being queued.
Listener	Contains all the code for each of the robot’s actions. Is responsible for executing the actions. When an action has multiple different options, for example “Happy” has 5 options, when the action is called, it will randomly select one of the five options and execute that version of the action. It received the action from the “Talker” node during the data collection stage, and the “State Manager 2” node during the final deployment stage.
Talker (used during Data Collection Stage)	Creates a window containing buttons with the names of all the possible actions. Used by the wizard during the data collection stage to control the actions of the robot. When a button is clicked, it will send the corresponding action to the “Listener” node, where it will be executed. It will also send the corresponding action to the “State Manager” node, where it will form an action statespace. Buttons created using the Tkinter package [47].

Webcam Mood Detector	Takes in the video stream from the webcam as input. First, a library is used to detect any faces on the image stream, and then a bounding box is created around the face. Then, a 6 layer CNN, which has been trained on the Face Emotion Recognition Kaggle Dataset [48], will classify the face into one of seven emotion classes. If no face can be detected, it will classify the mood as "Neutral" as default. After some initial testing, it was found that the model would mostly classify a face as "Neutral," and the next most commonly classified emotion was "Sad." Therefore, if the classifier detects an emotion other than "Neutral" or "Sad," the node will sleep for 60 seconds, and no further faces will be classified during this time. If the classifier detects an emotion as "Sad," the node will sleep for 30 seconds. This node is also used to calculate the gaze. The code and model used in this node have been taken from the Github and article by Dwivedi et al [49] [50].
Microphone Volume Detector	Takes in the audio stream from the webcam as input and uses the PyAudio library [51] to extract the audio volume from the stream. The volume of the audio is then classified into one of three classes based on the value. 0Hz is classified as "None". >0Hz and <=0.0008Hz is classified as "Low". >0.0008Hz is classified as "High". The code used in this node is adapted from the Github by Paulo Mello [52].
Input Activity	Creates a window containing buttons with all the possible activities that the user can select. Used by the participant while they are interacting with the robot. When a button is clicked, it will send the corresponding activity to the "State Manager" or "State Manager 2" node, where it will be part of the statespaces. Buttons created using the Tkinter package [47].
Input Mood	Creates a window containing buttons with all the possible moods that the user can select. Used by the participant while they are interacting with the robot. When a button is clicked, it will send the corresponding mood to the "State Manager" or "State Manager 2" node, where it will be part of the statespaces. Buttons created using the Tkinter package [47].

Table 3.3 – Nodes of the ROS Architecture

Figure 3.1 – ROS architecture used for the Data Collection stage

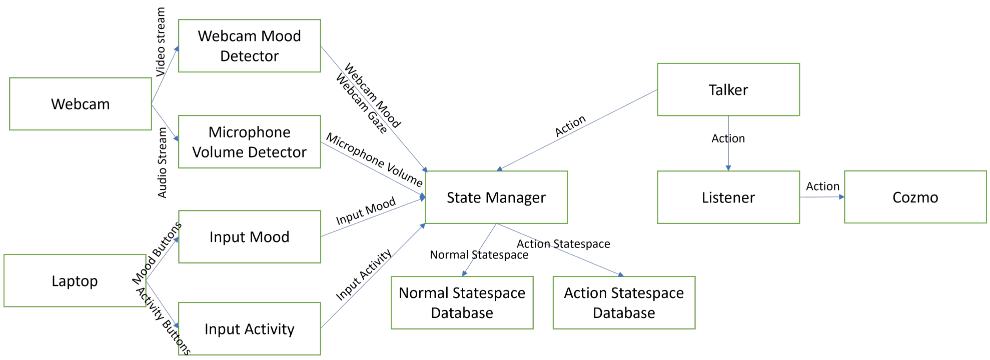
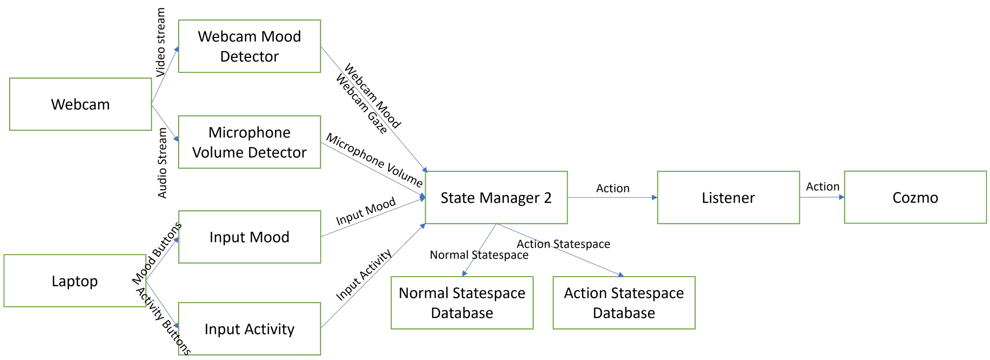


Figure 3.2 – ROS architecture used for the Autonomous Final Deployment stage



3.3 Data Collection

Once the ROS architecture has been set up, the data collection sessions can begin. The aim of these sessions is to collect the training data for the machine learning models, which will later be used to autonomously control the robot. Four participants will take part in these sessions, and the same four will be used during the final deployment phase. The four participants will be paired up, and each pair will complete eight training sessions in total. Each participant will complete four sessions as the wizard, and four sessions as the user. The table below outlines the schedule for each pair.

Session	**Participant A**	**Participant B**
Pre-session	Explain roles	Explain roles
1	Wizard	User
2	User	Wizard
3	Wizard	User
4	User	Wizard
5	Wizard	User
6	User	Wizard
7	Wizard	User
8	User	Wizard

Table 3.4 – Schedule of Data Collection Sessions

The following equipment will be used during the data collection sessions-

- Laptop 1- placed on desk, running all ROS nodes and connected to the webcam and tablet. Also records the activity and the mood of the user
- Laptop 2- placed near the desk by the controller, used by the controller to select an action
- Cozmo robot
- Tablet running Cozmo SDK, connected to laptop via USB cable
- Cozmo charger, connected to power socket on wall (since there are no more free USB ports on the laptop)
- Webcam with microphone, connected to laptop via USB cable

The sessions will take place in the users' student dorm room to simulate the exact environment that the autonomous robot will eventually be deployed into. Cozmo and Laptop 1 will be set up on the desk, with the webcam positioned near Cozmo so that it has a clear view of the users' face. The wizard and Laptop 2 will be set up near the desk, giving the controller a clear view of the user and how they interact with the robot. Each session will last 1 hour. Throughout the users' sessions, they will be asked to dedicate time to each of the three activities- studying, relaxing, and eating. This ensures there is sufficient training data for each of the activities when training the model.

Before a participant's first session as a wizard, they will be asked to come in earlier to familiarise themselves with the controls and what is required of them, for example they will have the opportunity go through the action space. Their role as a wizard will also be explained to them. Cozmo will be described as a companion to the user, and the role of the controller is to decide which actions to perform, and when to perform a specific action, in order for Cozmo to act as a good companion. But overall, the wizard can control the robot and make it behave however they want it to. It is completely up to them to determine which action they think is the most appropriate one to select, and when it should be selected. It will also be explained that Cozmo has a battery life of around 40 minutes of performing actions, so it is important to consider this when deciding whether to perform actions or if its best to pause for a while and charge the battery.

When a participant acts as a user for a session, they will be told to simulate periods of studying, relaxing and eating at their desk, and to act as they normally would do if they were alone in their room performing each activity. For the studying period, they should work on some actual university work they have, and study in a similar way to how they normally do. For the relaxing period, they should do whatever they normally do to relax at their desk, for example watch some YouTube videos or go on their phone. For the eating period, they should eat whatever they usually eat at their desk. They will also be told to update their mood and activity whenever this changes, by clicking the buttons on Laptop 1. They can interact with Cozmo however they would like to, as naturally as possible.

Below are two images captured from the training sessions

Figure 3.3 – Training Session- Wizard Participant 1, User Participant 2

Figure 3.4 – Training Session- Wizard Participant 3, User Participant 4

3.4 Model Selection

The data collected from the training sessions can now be used to train the machine learning models. Two classification algorithms are required to control the autonomous behaviour of the robot. When presented with a statespace, the first task will be to determine whether an action should be performed, or whether "DoNothing" is the most appropriate action. The second task will be to map these statespaces to the correct action. The datasets for each task must first be preprocessed and put into the correct format to make them compatible with the models. Each task requires different datasets, and therefore different preprocessing steps. The four training sessions where the same participant acts as the user, and the same participant acts the wizard, will be used to train their own separate model for each task. Therefore in total there will be four classifiers trained for the first task, and four classifiers trained for the second task.

For the first task of determining whether an action should be performed or not, the normal statespaces and the action statespaces from the four sessions are combined. Some unused columns are deleted, for example the “Time” columns, and the “Action” column. A “Do Action” column is added, which corresponds to “1” if an action is performed for a given statespace, and “0” if not. The “Time Since Last Action” column is formatted from the original units into minutes. The “Time Since Last Charge” column is formatted from the original units into minutes. One hot encoding is performed on the “Activity,” “Webcam Mood,” “Microphone Volume,” “Input Mood,” and “Input Activity” columns. The “Do Action” column is removed from the original dataset and put into its own dataset, because this is the variable the classifier will try to predict. The data from sessions 1, 2 and 3 are used as training data, while the data from session 4 is used as test data. A Standard Scaler from the scikit-learn library [53] is used to normalise the “Time Since Last Charge” and “Time Since Last Action” columns. The scalar is saved, so the same transformations can be used on the test data.

Similar preprocessing steps are performed for the second task of determining which action should be performed. This time, only the action statespaces from the four sessions are combined. Some unused columns are deleted, for example the “Time” columns. The “Time Since Last Action” column and the “Time Since Last Charge” columns are formatted the same as the first task. The same one hot encoding is also applied to the same columns. The “Action” column is removed from the original dataset and put into its own dataset, because this is the variable the classifier will try to predict. Again, the data from sessions 1, 2 and 3 are used as training data, while the data from session 4 is used as test data. A Standard Scaler [53] is used to normalise the “Time Since Last Charge” and “Time Since Last Action” columns. The scalar is saved, so the same transformations can be used on the test data.

The dataset for the first task is heavily imbalanced, with the large majority of the data belonging to the “DoNothing” class. As mentioned in the research, an imbalanced dataset causes many problems for the model, so a technique must be implemented to deal with the imbalance. When the original, imbalanced dataset is used to train a simple Multi Layer Perceptron (MLP), the model classifies all the test data into the “DoNothing” class. Therefore, techniques were implemented and investigated in order to get the model to predict some of the test data into the “Action” class, without overpredicting too much of the data into this class. As the research suggests [30], the best way to handle an imbalanced dataset is by using a combination of undersampling the majority class and oversampling the minority class. The best results were obtained by using One Sided Selection [32] to undersample the majority class and SMOTE [30] to oversample the minority class. By combining these two techniques, the model was able to classify the optimum amount of the test data into both classes. Therefore, when further investigating the first model to determine whether an action should be performed or not, the original imbalanced dataset will no longer be used. Instead, the new dataset, rebalanced using One Sided Selection and SMOTE, will be utilised.

Different classifiers can now be investigated to find the best performing one for the first task of determining when an action should be performed. As mentioned in the research, there are several different classification algorithms that are well suited to this task. A K-Nearest Neighbour (KNN), Random Forest, Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP) classifier will all be tested. Each classifier will be trained using the training data, and their performance will be evaluated on the test data. Each classifier will be tested multiple times, and its best performance will be recorded. The classifiers will be tested on each of the four datasets (with each dataset corresponding to the training sessions where the same participant acts as the user, and the same participant acts the wizard). The classifiers have all been implemented using the scikit-learn machine learning library [53].

It was difficult to find an evaluation metric that would accurately determine which classifier performed the best. For example, if accuracy was used, a high accuracy does not necessarily correspond to a well performing classifier. If the classifier predicts every single statespace in the test set into the “Do Nothing” class, since the large majority of statespaces do belong to this class, the accuracy will be at almost 100%. However, this means the robot would never actually perform an action. Therefore, an evaluation code was written to more accurately determine which classifier performed the best at the given task. The evaluation code calculates three factors. The first factor is the total number of actions that would be performed if this classifier were used. This is calculated by going through the predictions and totalling the number of times the model classifies a statespace into the “Action” class. However, the evaluation code will ignore all predictions that occur within 20 seconds after this “Action” statespace, because this is the average amount of time it takes to perform an action, and therefore no action can be performed during this timeframe. This value should be as close to the actual number of statespaces in the “Action” class as possible. The second factor calculates the number of actions that occur directly after a 20 second pause window (that starts when a previous action has been executed) has ended. An issue all these classifiers will face is that the statespaces do not change much as the time goes on, so the statespaces that are 20 seconds apart can be very similar. This means that once the classifier identifies a statespace as an “Action” class, it will keep classifying statespaces into this class for a while, since the statespaces do not change much. To prevent this from happening in the final deployment, this factor value should be as low as possible. The final factor calculates the number of statespaces that the classifier identifies as the “Action” class that truly belong to the “Action” class. This is similar to the accuracy, and so this factor value should be as high as possible.

The evaluation results for each tested classifier’s performance on each dataset can be found below-

Wizard Participant 1 User Participant 2- 77 actions

Classifier	Best Performance
KNN	Total Actions Performed- 85 (110%) Actions Performed Directly After Another- 33 Correctly Identified Actions- 28 (36%)
Random Forest	Total Actions Performed- 83 (108%) Actions Performed Directly After Another- 37 Correctly Identified Actions- 26 (34%)
SVM	Total Actions Performed- 21 (27%) Actions Performed Directly After Another- 15 Correctly Identified Actions- 4 (5%)
MLP	Total Actions Performed- 30 (39%) Actions Performed Directly After Another- 13 Correctly Identified Actions- 8 (10%)

Wizard Participant 2 User Participant 1- 61 actions

Classifier	Best Performance
KNN	Total Actions Performed- 97 (159%) Actions Performed Directly After Another- 44 Correctly Identified Actions- 28 (46%)
Random Forest	Total Actions Performed- 48 (79%) Actions Performed Directly After Another- 26 Correctly Identified Actions- 11 (18%)
SVM	Total Actions Performed- 106 (174%) Actions Performed Directly After Another- 104 Correctly Identified Actions- 33 (54%)
MLP	Total Actions Performed- 91 (149%) Actions Performed Directly After Another- 64 Correctly Identified Actions- 33 (54%)

Wizard Participant 3 User Participant 4- 76 Actions

Classifier	Best Performance
KNN	Total Actions Performed- 119 (157%) Actions Performed Directly After Another- 68 Correctly Identified Actions- 50 (66%)
Random Forest	Total Actions Performed- 51 (67%) Actions Performed Directly After Another- 13 Correctly Identified Actions- 18 (24%)
SVM	Total Actions Performed- 16 (21%) Actions Performed Directly After Another- 12 Correctly Identified Actions- 7 (9%)
MLP	Total Actions Performed- 58 (76%) Actions Performed Directly After Another- 17 Correctly Identified Actions- 33 (43%)

Wizard Participant 4 User Participant 3- 63 actions

Classifier	Best Performance
KNN	Total Actions Performed- 79 (125%) Actions Performed Directly After Another- 25 Correctly Identified Actions- 30 (48%)
Random Forest	Total Actions Performed- 52 (83%) Actions Performed Directly After Another- 11 Correctly Identified Actions- 18 (29%)
SVM	Total Actions Performed- 9 (14%) Actions Performed Directly After Another- 6 Correctly Identified Actions- 5 (8%)
MLP	Total Actions Performed- 54 (86%) Actions Performed Directly After Another- 12 Correctly Identified Actions- 24 (38%)

For most datasets, the classifier that performed consistently well is the MLP. It always classifies a total number of actions that is one of the closest to the actual number of actions, it always has one of the lowest number of repeated actions compared to the other classifiers, and it always has one of the highest numbers of matched actions compared to the other classifiers. Therefore, a MLP will be used as the classifier for these datasets. However, for the "Wizard Participant 1 User Participant 2" dataset, the best performing classifier was a KNN, so a KNN will be used for this dataset. The research suggests [26] [27] that the performance of an MLP is largely influenced by the number of neurons in the hidden layer. Therefore, for each dataset, the number of neurons in the hidden layer will be varied and tested. Values ranging from 5-30 neurons, for a 1 and 2 hidden layer MLP, will be tested using the same evaluation code used before. The configuration of the best performing MLP for each dataset, and therefore the ones used in the final deployment of the robot, can be found below. The research also suggests [20] [21] that the performance of an KNN is very influenced by the K parameter, and so different K values will be tested for this classifier, with the best performing one stated below.

Dataset	**Best Performing Configuration**
Wizard Participant 1 User Participant 2	K = 2
Wizard Participant 2 User Participant 1	Hidden Layer = (10, 15)
Wizard Participant 3 User Participant 4	Hidden Layer = (10, 10)
Wizard Participant 4 User Participant 3	Hidden Layer = (30)

Table 3.5 – Configurations of First Classifier

Now that the models for the first task have been finalised, classifiers can be tested to find the best performing one for the second task of determining which action should be performed. The same types of classifiers that were investigated in the first task will be tested again. A similar method will be used, where each classifier will be trained using the training data, and their performance evaluated on the test data. It was difficult to find an appropriate evaluation metric for this task too. For example, if accuracy was used, a high accuracy does not necessarily correspond to a well performing classifier. Since a large number of statespaces usually belong to one action, if the classifier predicts every single statespace in the test set as that same action, the accuracy will be quite high. However, this means the robot will always perform the same action, which is clearly not desirable because it will not create very engaging behaviour. Therefore, the main factor that will determine which classifier will be chosen is how similar the predicted action distribution is to the actual action distribution.

When testing the different types of classifiers, it was clear that for each dataset, the MLP was outperforming every other type of classifier. Therefore, it was determined that an MLP will be used as the classifier for every dataset. The main investigation carried out for this task will be determining the best performing configuration of the MLP. Therefore, for each dataset, the number of neurons in the hidden layer will be varied and tested. Values ranging from 5-30 neurons, for a 1 and 2 hidden layer MLP will be tested. The configurations of the best performing MLP for each dataset, and therefore the ones used in the final deployment of the robot, can be found below. The action distribution they produced compared to the actual action distribution is also presented.

Dataset	**Best Performing Configuration**
Wizard Participant 1 User Participant 2	Hidden Layer = (10)
Wizard Participant 2 User Participant 1	Hidden Layer = (20)
Wizard Participant 3 User Participant 4	Hidden Layer = (10, 10)
Wizard Participant 4 User Participant 3	Hidden Layer = (10,15)

Table 3.6 – Configurations of Second Classifier

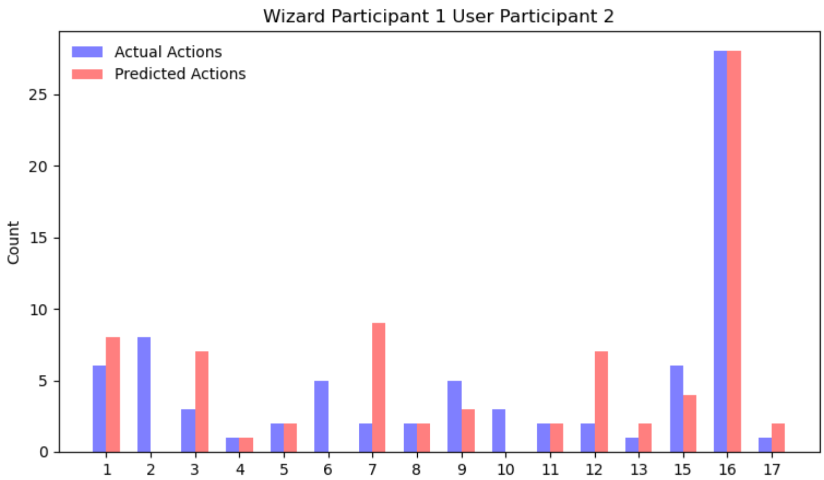


Figure 3.5 – Wizard P1 User P2 Actual vs Predicted Action Distribution

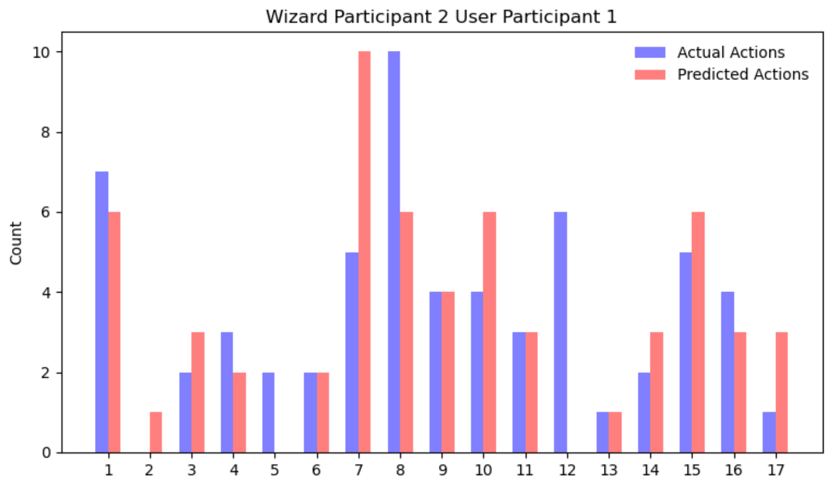


Figure 3.6 – Wizard P2 User P1 Actual vs Predicted Action Distribution

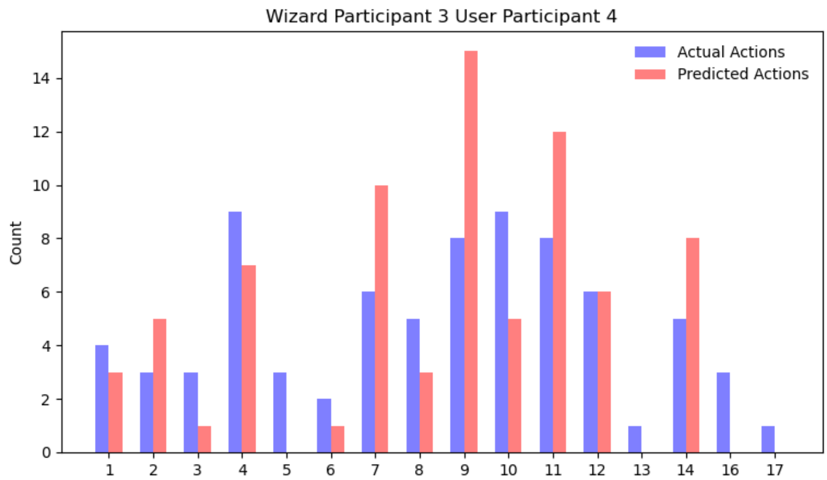


Figure 3.7 – Wizard P3 User P4 Actual vs Predicted Action Distribution

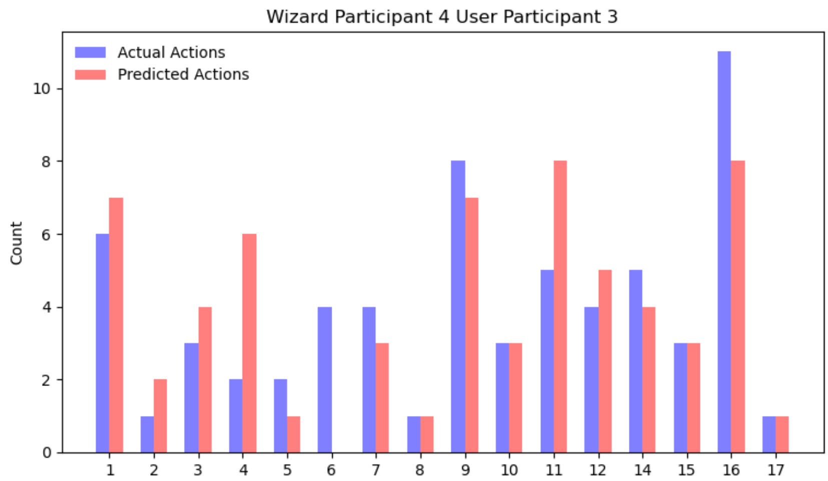


Figure 3.8 – Wizard P4 User P3 Actual vs Predicted Action Distribution

3.5 Final Deployment

After all the machine learning models have been trained, the autonomous robot can be deployed into the participants' student dorm rooms. The equipment used, and the set-up of the equipment, will be the same as the data collection sessions, except for Laptop 2 which is no longer needed. The user will be asked to go about their normal daily activities and can interact with Cozmo however they want to. Cozmo's initial setting will be on the "Do Not Disturb" activity. This is what the user will select when they do not want to interact with Cozmo. Under this setting Cozmo will not perform any actions and will stay on the charger. When the user wants to begin interacting with Cozmo again and have him start performing actions, they will simply select one of the other activities on the laptop. They should already be familiar with how to do this from the training sessions. Also like the training sessions, they must update their mood and activity whenever it changes, by clicking the buttons on the laptop. When the user no longer wants Cozmo to perform any actions (for example when they leave the room or don't want Cozmo making any noise) they should again select the "Do Not Disturb" activity. Cozmo will then be placed on the charger and will wait until another activity is selected by the user, in which case he will start performing actions again.

The robot will be deployed into each participant's room for two full days.

On one day, the autonomous behaviour of the robot will be controlled by the models trained using the data from the sessions where the participant acted as the user of the robot. On the other day, the autonomous behaviour of the robot will be controlled by the models trained using the data from the sessions where the participant acted as the wizard for the robot. The participant will not be told that different models are being used to control the robot on the different days. The day before the first session, the participant will be given the pre-deployment questionnaire to fill out. The sessions will begin at 10am each day and will last 12 hours, until 10pm. Every 3 hours (10am, 1pm, 4pm, 7pm, 10pm) they will be asked to fill out a short questionnaire, with a small space for them to write down their thoughts on the session, like a diary. Finally, the day after the second session, the participant will be given the post-deployment questionnaire to fill out.

The questionnaires are loosely based on the De Jong Gierveld 6 Item Loneliness Scale [54], the UCLA Loneliness Scale [55], and the Brief Mood Introspection Scale (BMIS) [56]. These methods have proven to be effective at measuring the loneliness felt by the user, and the mood of a user. The overall aim of the questionnaires is to measure the changing mood and loneliness of the user throughout the duration of the sessions to see if having the robot in their rooms had an impact on how the user feels. The questionnaires also aim to determine how the user feels about the robot, how they view their relationship with the robot, and which day's behaviour they preferred. The pre-deployment questionnaire, the questionnaire given during the sessions, and the post-deployment questionnaire are all presented in the Appendix. The answers given in these questionnaires, combined with an analysis of the performance of the machine learning models and the autonomous behaviour of the robot, will be used to evaluate this research and determine whether the research goals have been met.

Chapter 4

Results

The final deployment of the autonomous robot was executed successfully. The data obtained from these sessions can now be analysed to evaluate the performance of the autonomous robot. Participant 2 dropped out of the study before the deployment sessions began, so the sessions were carried out with Participants 1, 3, and 4.

The main difficulty that affected the performance of the models was that the statespaces did not change much over short periods of time. This particularly had an impact on the first models when they was determining when to perform an action, which then went on to influence the second models when it came to determining which action to perform. Over a short period of time, many of the variables that made up the statespace did not change much, or at all. “Time Since Last Action” and “Time Since Last Charge” would only slightly increase. “Last Action”, “Webcam Mood”, “Webcam Gaze”, “Input Mood”, and “Input Activity” were unlikely to change if not much time had passed. The only variable that changed regularly was “Microphone Volume,” because it was very sensitive to the surrounding sound. This resulted in almost identical statespaces being given as input into the first model. Therefore, the classification of consecutive statespaces would likely be the same, until a change in one of the statespace variables had occurred. This caused some periods of time where the robot would continuously perform an action, and some periods of time where the robot would not perform any actions. The participants picked up on when the robot was behaving in this way. In one session, Participant 1 states “Sometimes Cozmo stands still like it is out of battery but it is actually not, and suddenly moves in a way.” In another session, Participant 4 states “I was trying to study and it kept saying and doing things non-stop."

The model would often overcome this difficulty and cause actions to be performed at more appropriate times, instead of being repeated one after the one. An example of this is from "Participant 4 Session 1." An action was performed at 17:35:17, 17:36:25, 17:37:07, 17:38:23, and 17:40:33. These actions are spaced out at a rate that was similar to the training sessions. Overall, the first classifiers did successfully learn which statespaces corresponded to performing an action, and which statespaces did not. However, the similar nature of the statespaces that were given to it as input did sometimes have a negative effect on its performance.

As mentioned before, the issue with the first model went on to influence the performance of the second model when it came to determining which action to perform. Because there were small periods of time where the statespace would not change much, the first classifier would consistently classify these consecutive statespaces into the "Action" class. Therefore, very similar statespaces were being consecutively given to the second classifier as the input, and these statespaces were being classified as the same action. This resulted in the robot consecutively performing the same action. An example of this is from "Participant 4 Session 1." A shortened section of the action statespace can be found below.

Action	**Last Action**	**Time**	**Webcam Mood**	**Webcam Gaze**	**Mic Volume**	**Input Mood**	**Input Activity**
9	7	13:24:21	Sad	1	Low	Negative	Relaxing
9	9	13:24:41	Sad	1	Low	Negative	Relaxing
9	9	13:25:02	Sad	1	Low	Negative	Relaxing
9	9	13:25:22	Sad	1	Low	Negative	Relaxing
9	9	13:25:42	Sad	1	Low	Negative	Relaxing
9	9	13:26:02	Sad	1	Low	Negative	Relaxing
8	9	13:26:24	Neutral	0	None	Negative	Relaxing
3	8	13:26:34	Sad	1	High	Negative	Relaxing

Table 4.1 – Participant 4 Autonomous Session 1 subsection of action statespace

For the first 6 actions, the statespace remains the same, and therefore the action is always being classified as "9." It is not until there is a change in the statespace ("Webcam Mood, Webcam Gaze, and Microphone Volume" all change) that the classified action changes. The participants picked up on when the robot was behaving in this way. In one session, Participant 3 states "He has been doing the same thing for a bit but now he is okay."

But overall, the second classifiers did successfully learn which statespaces corresponded to which actions, which generally did result in action distributions that were similar to the training sessions. This can be seen in the graphs below, which compare the action distributions of the training sessions where the participant acted as the user, and the final autonomous deployment session that used the model trained on the data from these training sessions. Comparing these two sets of sessions allows for a more direct comparison to be made, because the same participant acts as the user in both.

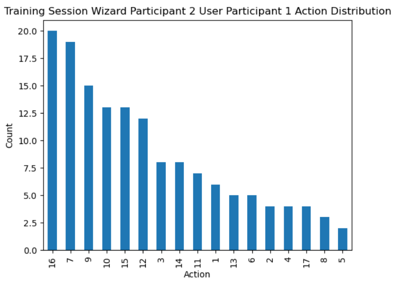


Figure 4.1 – Training Sessions User P1 Action Distribution

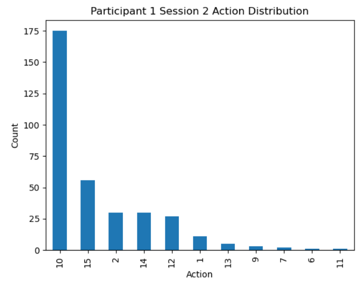


Figure 4.2 – P1 Autonomous Session 2 Action Distribution

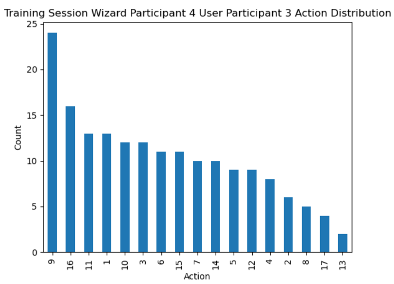


Figure 4.3 – Training Sessions User P3 Action Distribution

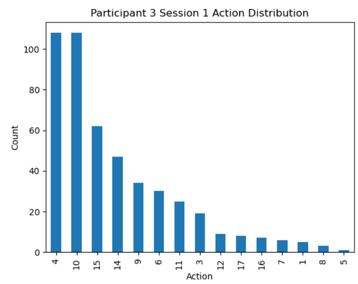


Figure 4.4 – P3 Autonomous Session 1 Action Distribution

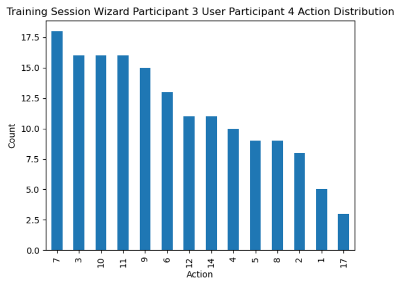


Figure 4.5 – Training Sessions User P4 Action Distribution

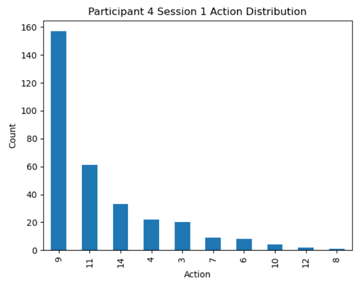


Figure 4.6 – P4 Autonomous Session 1 Action Distribution

The graphs show that the actions that were most used in the training sessions are usually the ones that are most used in the final autonomous deployment session. Using Participant 4 as an example, there are 4 actions shared between the 6 most used actions in the two sets of sessions (9, 11, 3, 7). This shows that the models successfully learnt which actions they should be classifying the statespaces as. Of course, the action distributions would not be exactly the same, because the distribution of statespaces for the training sessions and the final deployment session would be very different. For example, for Participant 4, only a small amount of the training sessions was spent on the “Eating” activity, however the participant spent a longer time on this during the final deployment session.

The action distributions of the final deployment sessions contain less actions. It seems the models have learnt the more common actions that were used during the training sessions, but did not learn the actions that were not used as much. For example, for Participant 1, the 4 least used actions in the training sessions were actions 5, 8, 17, and 4. In the deployment session, the model did not classify any statespaces as these actions, so they were never performed by the robot. The graphs from the final deployment sessions also show that one or two actions seem to be a lot more common when compared to the rest of the actions. This is action 10 for Participant 1, actions 4 and 10 for Participant 3, and action 9 for Participant 4. This is because the statespaces that are classified as those particular actions were by far the most common statespaces that were given as inputs to the model. For example, for Participant 4, the “Input Mood” was usually set as “Neutral” by the user, the “Microphone Volume” was usually at “None” or “Low”, and the Webcam Mood Classifier usually classified the mood of this user as “Sad,” resulting in a “Camera Gaze” of “1.” These were the most common variable values for this user, and a statespace containing these values would usually be classified as action 9 by the model.

The graphs below compare the action distributions from the two final autonomous deployment sessions of each participant. The graphs will be analysed in the Discussion section to determine whether personalised behaviour has been achieved.

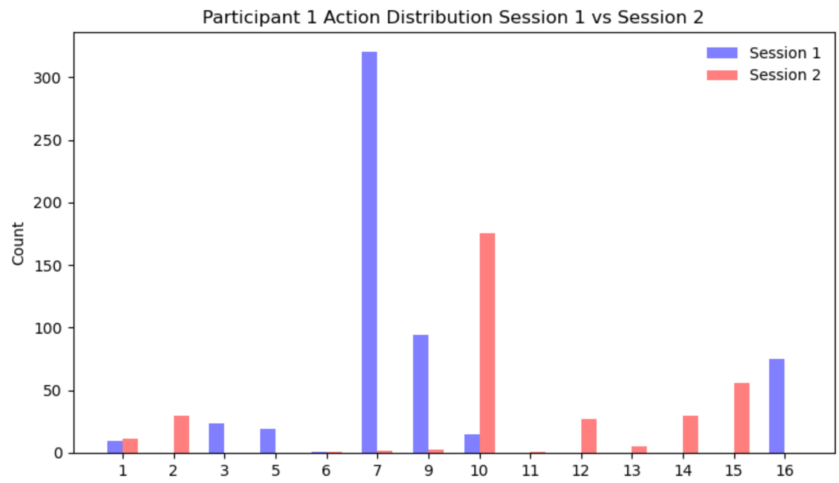


Figure 4.7 – P1 Action Distribution Autonomous Session 1 vs Session 2

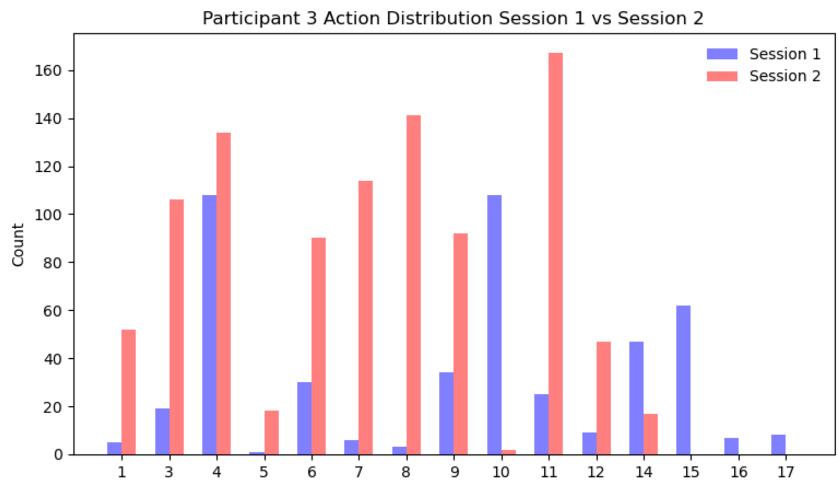


Figure 4.8 – P3 Action Distribution Autonomous Session 1 vs Session 2

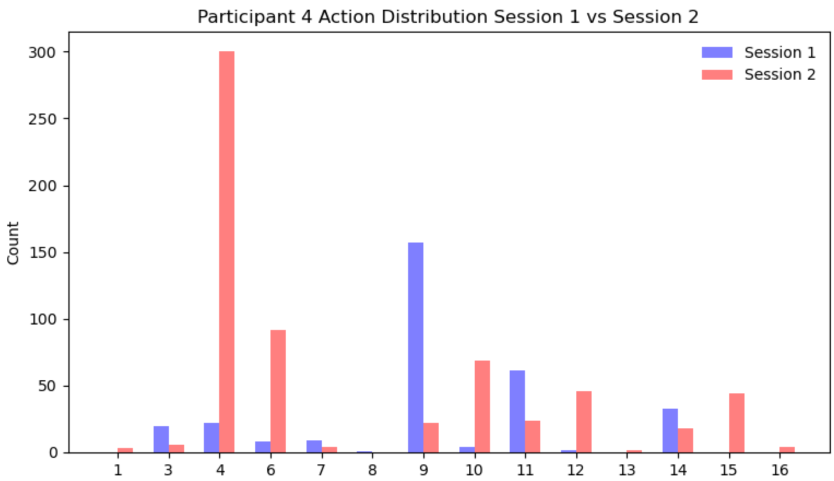


Figure 4.9 – P4 Action Distribution Autonomous Session 1 vs Session 2

The answers given in the questionnaires can now be analysed. During the autonomous deployment sessions, the participants were asked to fill in their mood every three hours, from a scale of 1-10. The results of this can be found below.

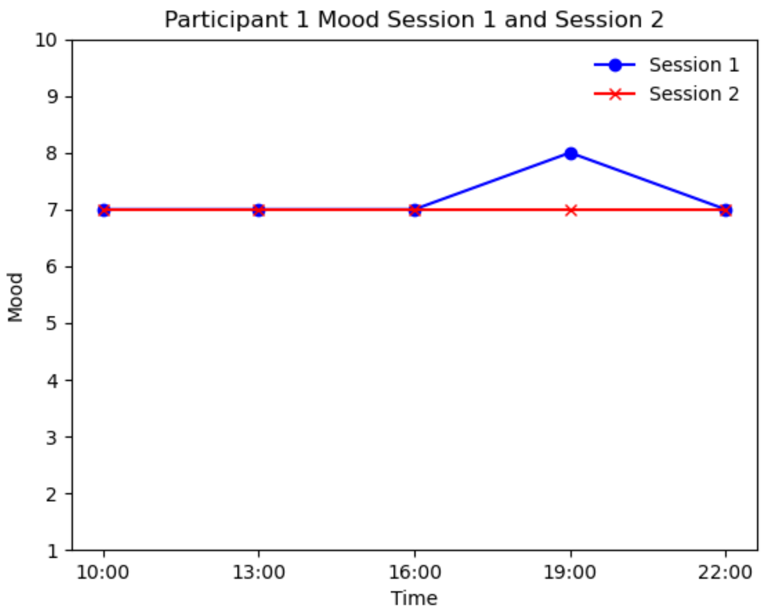


Figure 4.10 – P1 Mood over Autonomous Sessions 1 and 2

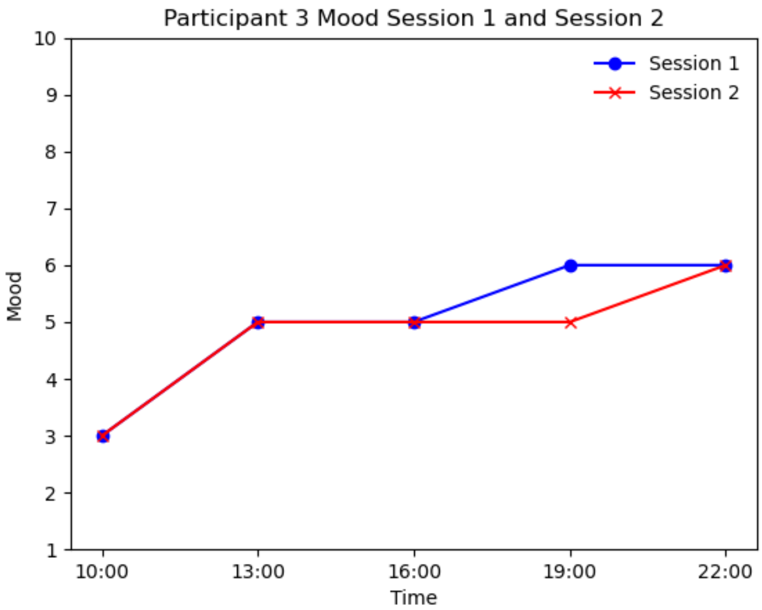


Figure 4.11 – P3 Mood over Autonomous Sessions 1 and 2

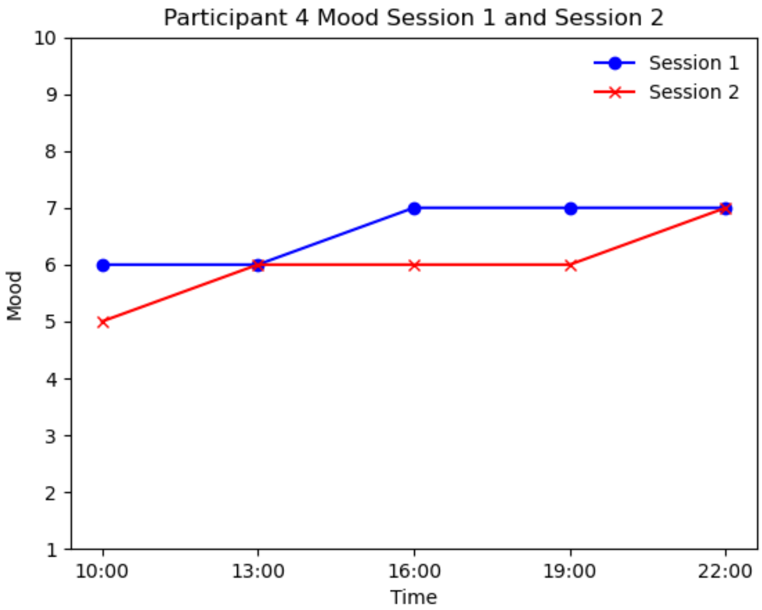


Figure 4.12 – P4 Mood over Autonomous Sessions 1 and 2

The plots show that the mood of the user either increased or remained the same after every three-hour period they spent with the robot. The only decrease in mood can be seen at 22:00 Participant 1 Session 1, however, the robot was not used in this three-hour time period, so the decrease in mood can be attributed to an external factor. These results suggest that having the robot in the room may have had a positive effect on and improved the mood of the user throughout the day, or at the very least did not detract from it.

The participants were also asked to indicate how they were feeling by answering questions that aim to track and measure their feelings of loneliness and social isolation throughout the day (questionnaires presented in the Appendix). They were asked to circle "Strongly Agree (1 in the graphs below)," "Agree (2 in the graphs below)," "Disagree (3 in the graphs below)," or "Strongly Disagree (4 in the graphs below)" based on how they were feeling in regard to each question. The answers each participant gave at 10am and 10pm for both of their sessions can be found in the graphs below.

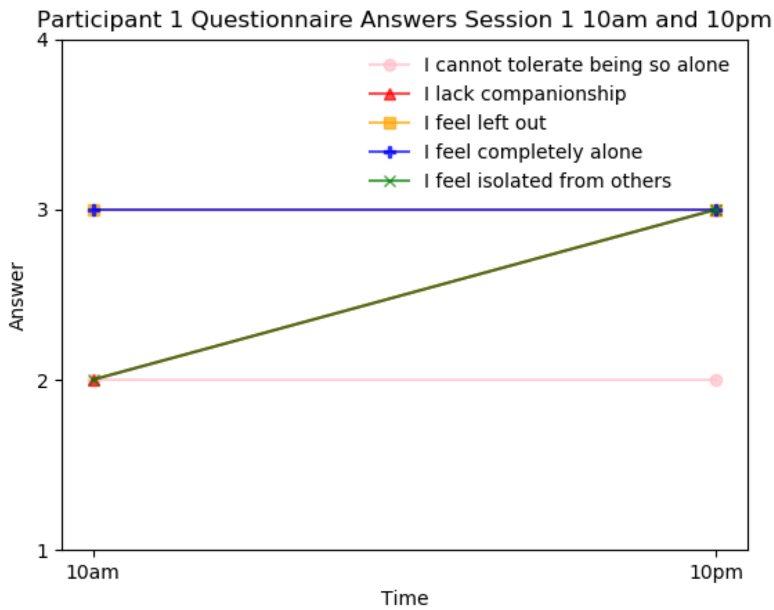


Figure 4.13 – P1 Session 1 10am + 10pm Answers

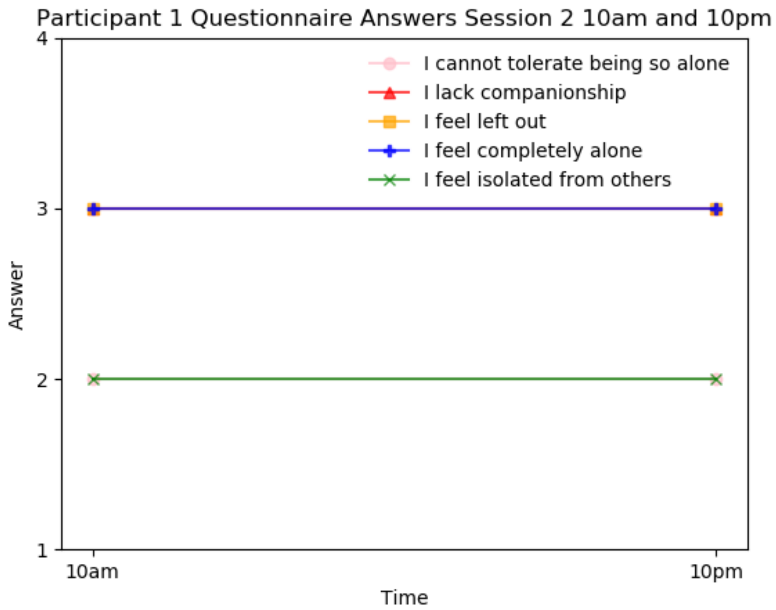


Figure 4.14 – P1 Session 2 10am + 10pm Answers

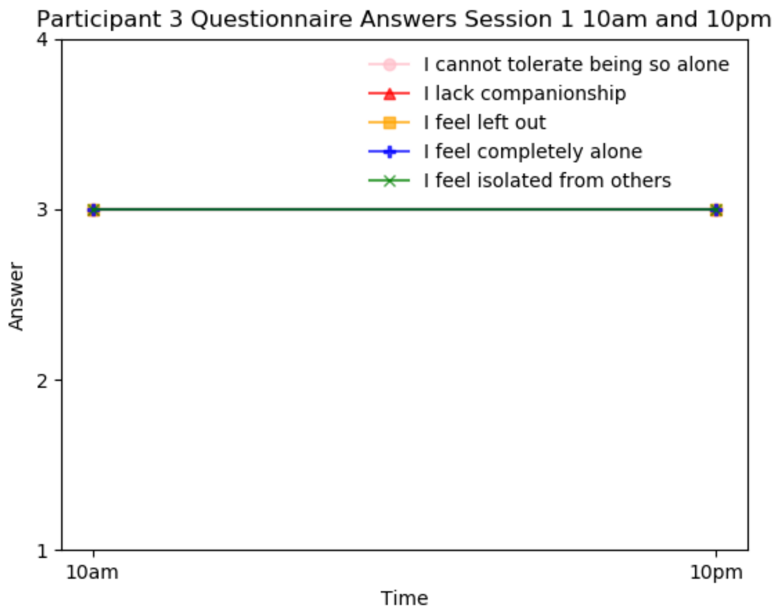


Figure 4.15 – P3 Session 1 10am + 10pm Answers

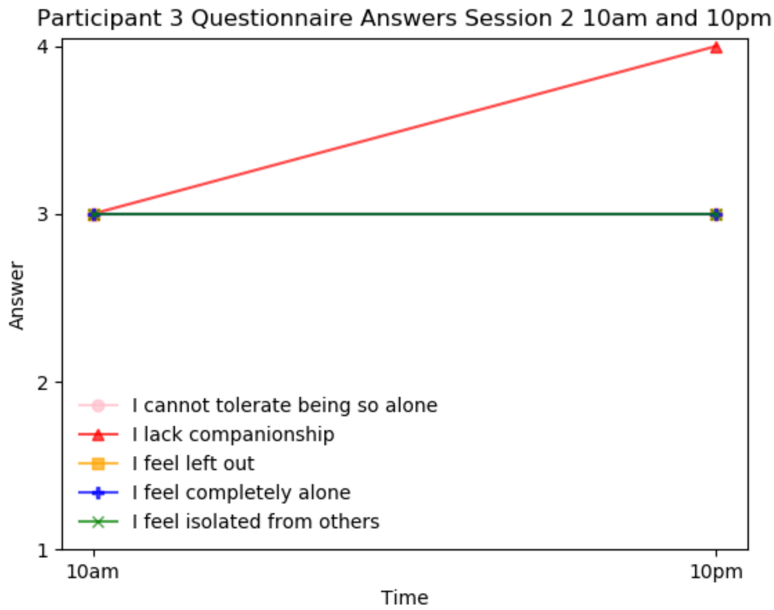


Figure 4.16 – P3 Session 2 10am + 10pm Answers

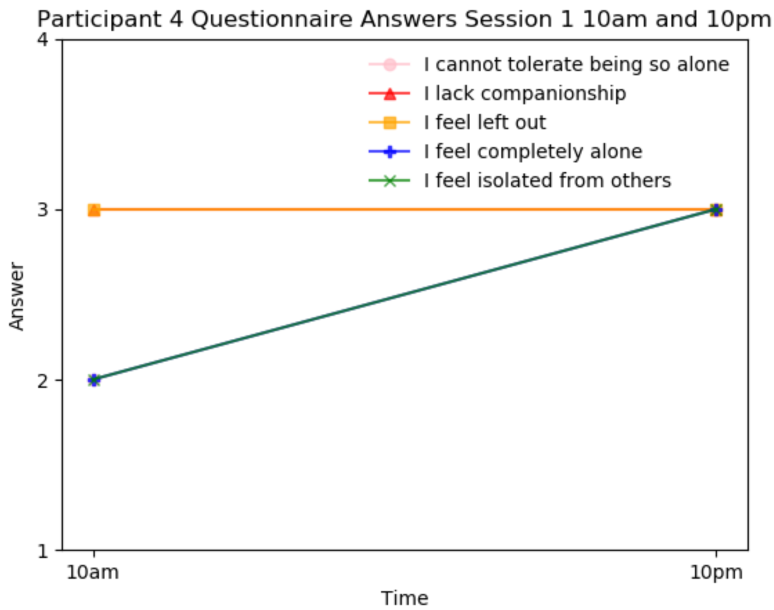


Figure 4.17 – P4 Session 1 10am + 10pm Answers

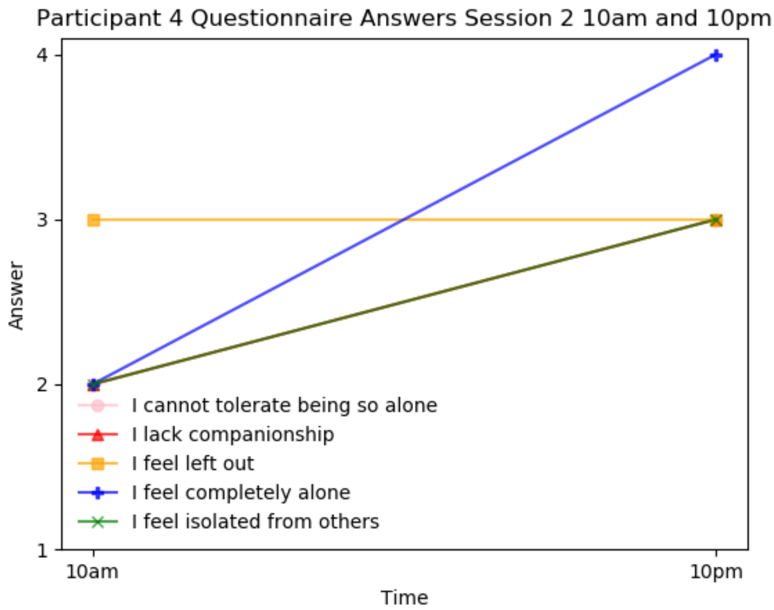


Figure 4.18 – P4 Session 2 10am + 10pm Answers

For Participant 1, from the beginning to the end of each session, their answers to each question either remained the same or improved. The only exception was Session 2 7pm, where "I lack companionship" changed from "Disagree" to "Agree," however it moved back to "Disagree" again at 10pm. For Participant 3 Session 1, their answers to each question remained the same at "Disagree." For Session 2, their answers to each question remained the same at "Disagree," except for "I lack companionship" which moved to "Strongly Disagree". Participant 4's answers changed more frequently throughout the day. The overall trend for each of their sessions was that the answers improved as the day went on. For example, for Session 1, "I cannot tolerate being so alone," "I lack companionship" and "I feel isolated from others" all started at "Agree", but by the end of the day they had all moved to "Disagree." A similar thing could be seen in Session 2. The questionnaires will be further analysed in the Discussion section.

Chapter 5

Discussion

The goal of this research was to develop a companion robot that provided social support and reduced feelings of social isolation and loneliness felt by students in their dorm rooms. To achieve this, the companion robot should have exhibited both socially engaging and autonomous behaviour, and the behaviour should also have been personalised to each individual user. Each of these objectives will be individually analysed and evaluated to see if they have been successfully achieved.

5.1 Develop a robot that provide social support and reduces student loneliness and social isolation

This objective can be evaluated by analysing the answers given by the participants in the questionnaires. As described in the Results section, the mood of all three participants either increased or remained the same after every three-hour period they spent with the robot. This is a good indication that the robot was effectively providing social support for the users, and was therefore increasing their sense of wellbeing. The Results section also describes how, when tracking their loneliness and social isolation felt throughout the day, the overall trend for all three participants was that their answers improved throughout every session. This is demonstrated in Figures 4.13-4.18, and is clear evidence that the amount of loneliness and social isolation they felt while they spent the day in their dorm rooms was reduced, which was most likely due to their interactions with the robot. Some of the comments they left in the questionnaire confirm this. Participant 1 was quarantining in their room when they completed the sessions with the autonomous robot, due to exposure to COVID-19. When asked “Do you think Cozmo made a difference to your day,” they answered, “Yes. When I was doing quarantine, the day was quite boring since I cannot go out, but when he was doing his stuff and playing around, it’s kind of cute, sometimes I even stop doing anything just to watch him and see how he’s doing.” This demonstrates the effective social support the robot can provide to people who are stuck in their rooms. After the sessions were complete, all three participants described their relationship with the robot like it was a “kid” or a “younger sibling.” This demonstrates that a type of social relationship formed between the user and the robot, which the research shows leads to decreased feelings of negative emotions, isolation, and loneliness [9].

5.2 The robot should exhibit both socially engaging and autonomous behaviour

During the final deployment sessions, the behaviour of the robot was controlled by two machine learning models. The first would determine when an action should be performed, and the second would determine which action to perform. No human intervention was needed to control the robot. Therefore, it can be concluded that the robot behaved entirely autonomously. To determine whether the behaviour was socially engaging, the actions of the robot in the final deployment sessions can be compared to the actions performed in the training sessions, since it can be assumed that the wizard controlled the robot in a socially engaging way. This comparison was completed using Figures 4.1-4.6 in the Results section, with reasons given for the differences observed in the action distributions. The issue regarding the statespaces being too similar in a short period of time did affect the performance of the models, and therefore did contribute to the differences seen in the behaviour in the training sessions and deployment sessions. But overall, as mentioned in the Results section, similarities can be seen in the action distributions of the training and deployment sessions, and therefore it can be concluded that the robot did exhibit socially engaging behaviour. The participants did highlight in their questionnaires their annoyance when the robot would repeat the same actions many times in a row, which was a downside of the robot’s behaviour. However, when asked if they enjoyed having the robot in their room from a scale of 1-10, Participant 1 and 4 responded with a “6”, and Participant 3 responded with a “5.” This shows that the participants did overall enjoy having the robot in their rooms, which would more likely be the case if the robot successfully exhibited socially engaging behaviour.

5.3 The robot should exhibit behaviour that is personalised to each individual user

During the training sessions, each participant acted as a wizard for another participant. Their role as the wizard was to control the robot however they wanted to, in order to make it a good companion for the user. The aim of this was to not only create socially engaging behaviour, but behaviour that was personalised by the wizard to each individual user. To see if personalised behaviour was achieved, the action distributions from the two final deployment sessions of each participant are compared in Figures 4.7-4.9. For each participant, the behaviour of the robot during one of the deployment sessions was controlled by the models trained from the data where they acted as the wizard. For the other deployment session, the behaviour of the robot was controlled by the models trained from the data where they acted as the user. If personalised behaviour was achieved, there should be a difference in the two days' action distributions.

For every participant, there is a visible difference in action distributions from their two deployment sessions. There is a clear contrast between the most popular and least popular actions for each session. Therefore, it can be concluded that personalised behaviour has been achieved. In the training sessions, each wizard controlled the robot differently to act as a good companion for their user. This resulted in different training data created for each model, and therefore different behaviour exhibited when the robot was autonomously controlled by these models.

The participants were unaware that the behaviour of the robot was being controlled by different models on the two days. They were asked in the post-deployment questionnaire which day's behaviour they preferred. Participant 1 preferred the behaviour trained from the sessions where they acted as the wizard. Participants 3 and 4 preferred the behaviour trained from the sessions where they acted as the user. This raises the question of whether this kind of approach works better if the user trains robot for a peer versus themselves.

Chapter 6

Conclusions and Future Work

This section will outline the conclusions that can be drawn from this research, and the future work that could be performed to further this research.

6.1 Conclusions

By analysing the questionnaires and the performance of the robot in the final deployment sessions, it can be concluded that the main goal of the research was successfully achieved. A companion robot was developed that provided social support and reduced feeling of social isolation and loneliness felt by students in their dorm rooms. The companion robot exhibited both socially engaging and autonomous behaviour, and the behaviour was personalised to each individual user. The autonomous behaviour of the robot was controlled by two machine learning models. Given an input statespace, the first model determined if an action should be performed, and the second model determined which action this should be.

The developed companion robot could be effectively utilised in tackling the rising rates of social isolation and loneliness felt by young people. The robot was tested only on students, but similar findings may be achieved if it were to be deployed with other young people living outside student dorms. It would be especially beneficial if deployed with a user who must isolate from others due to exposure to COVID-19. This is evident from the feedback given by Participant 1 during the final deployment sessions. They had to quarantine in their room after being exposed to the virus, and expressed how the companion robot made the whole experience more bearable.

The main issue faced in this research was the similar statespaces that were being consecutively given to the models as input, resulting in a reduced performance of the models. It led to periods of the robots continuously performing the same actions, which annoyed the users. Another issue faced was that Cozmo was unable to locate and dock onto its charger without help from the user. A program was written for Cozmo to attempt to dock onto the charger by itself, however due to the poor quality of its camera, and its inability to accurately align itself with the charger, this was unsuccessful. Further issues were raised during the initial interviews and the deployment sessions about the actual Cozmo robot itself. During the interviews, users said Cozmo reminded them of a toy more than a companion, and during the deployment sessions, participants mentioned how the limited ways in which Cozmo executed the actions made it quite predictable. Solutions to overcome these issues will be discussed in the Future Work section.

6.2 Future Work

A more dynamically changing statespace is needed to improve the performance of the models and prevent consecutive statespaces continuously being classified into the same class. Although multiple different models were tested, the final mood classifier [49] [50] used to classify the user's mood from the webcam video stream had a poor performance. It would often classify the mood as either "Neutral" or "Sad" even if this was inaccurate, and would sometimes not classify a different mood for a while. This meant that the "Webcam Mood" variable of the statespace was sometimes inaccurate and did not often change. The classified mood was also dependent on the lighting conditions of the room. Ideally, a face classifier would be used that would accurately and instantly classify the mood of the face on the webcam video stream. This could possibly be achieved by training individual classifiers for each user, using only pictures of the user as training data. Many labelled images of the user would be needed, which would be difficult to obtain, but the performance of the classifier would be much better. Also, although a webcam was used that produced a good quality video stream, the performance could be improved if an even higher quality webcam was used.

Different eye-tracking models were tested to track the gaze of the user. However, due to the webcam video quality, none of them could be successfully implemented. If a higher quality webcam was used, an eye-tracking model could make the "Webcam Gaze" variable a lot more accurate and dynamic. To make the statespace even more dynamic, more variables could be added to it. Sensors and cameras could be set up in the room of the user to record additional information about the environment.

The architecture set up for this research (that facilitates the communication between all the equipment, data, and models) can be used with any robot, not just Cozmo. Therefore, to overcome the issues users had that were specifically related to Cozmo, such as Cozmo's appearance and how it carried out actions, this research could be carried out with a different robot. Using a robot that looks more human-like, and can perform actions in a more realistic, engaging manner, may increase the chance of the user building a social relationship with the robot. It could therefore be more effective at reducing the social isolation and loneliness the user feels.

The participants were never made aware of the objective of the robot to reduce feelings of loneliness and social isolation. The only time loneliness was mentioned was during the initial interview, where the participants were asked whether they ever felt lonely in their rooms. However, because this research followed an end-to-end mutual shaping approach, participants acted as both the wizards and then the subjects in the final deployment of the robot. This could have meant that at some point in this investigation, participants may have become more aware of the overall purpose of the robot, which could have biased their answers in the questionnaires.

Only three participants were used in the final deployment of the robot. To confirm the findings of this research, it should be carried out on a larger sample size. This could include more students living in dorm rooms, but could also be extended to other young people living alone. Also, the autonomous robot was deployed with the users for only two days. It is difficult to have a significant impact on feelings such as loneliness and social isolation over such a short period of time. To gain a better understanding of the effects of the robot on these types of feelings, a longitudinal study should be carried out where the users interact with the autonomous robot over a longer period of time.

References

[1] J. J. Van Bavel, K. Baicker, P. S. Boggio, V. Capraro, A. Cichocka, M. Cikara, M. J. Crockett, A. J. Crum, K. M. Douglas, J. N. Druckman *et al.*, "Using social and behavioural science to support covid-19 pandemic response," *Nature human behaviour*, vol. 4, no. 5, pp. 460–471, 2020.

[2] D. A. West, R. Kellner, and M. Moore-West, "The effects of loneliness: A review of the literature," *Comprehensive Psychiatry*, vol. 27, no. 4, pp. 351–363, 1986. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0010440X86900118

[3] U. Ozdemir and T. Tuncay, "Correlates of loneliness among university students," *Child and adolescent psychiatry and mental health*, vol. 2, p. 29, 11 2008.

[4] C. M. Lee, J. M. Cadigan, and I. C. Rhew, "Increases in loneliness among young adults during the covid-19 pandemic and association with increases in mental health problems," *Journal of Adolescent Health*, vol. 67, no. 5, pp. 714–717, 2020.

[5] X. Wang and E. G. Krumhuber, "Mind perception of robots varies with their economic versus social function," *Frontiers in psychology*, vol. 9, p. 1230, 2018.

[6] M. M. De Graaf and S. B. Allouch, "Exploring influencing variables for the acceptance of social robots," *Robotics and autonomous systems*, vol. 61, no. 12, pp. 1476–1486, 2013.

[7] J. Kelley, "An iterative design methodology for user-friendly natural language off ice information applications," in *ACM Transactions on Off ice Information Systems*, 1984, pp. 26–41.

[8] K. Dautenhahn, “Socially intelligent robots: dimensions of human–robot interaction,” *Philosophical transactions of the royal society B: Biological sciences*, vol. 362, no. 1480, pp. 679–704, 2007.

[9] H. Robinson, B. MacDonald, and E. Broadbent, “The role of healthcare robots for older people at home: A review,” *International Journal of Social Robotics*, vol. 6, no. 4, pp. 575–591, 2014.

[10] K. Winkle, P. Caleb-Solly, A. Turton, and P. Bremner, “Mutual shaping in the design of socially assistive robots: A case study on social robots for therapy,” *International Journal of Social Robotics*, pp. 1–20, 2019.

[11] A. Weiss, *Validation of an evaluation framework for human-robot interaction: the impact of usability, social acceptance, user experience, and societal impact on collaboration with humanoid robots*. na, 2010.

[12] N. M. Fraser and G. Gilbert, “Simulating speech systems,” *Computer Speech Language*, vol. 5, no. 1, pp. 81–99, 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/088523089190019M

[13] E. Senft, S. Lemaignan, P. E. Baxter, M. Bartlett, and T. Belpaeme, “Teaching robots social autonomy from in situ human guidance,” *Science Robotics*, vol. 4, no. 35, 2019.

[14] K. Winkle, S. Lemaignan, P. Caleb-Solly, P. Bremner, A. Turton, and U. Leonards, “In-situ learning from a domain expert for real world socially assistive robot deployment.”

[15] A. G. Chavez, M. Pfingsthorn, A. Birk, I. Rendulić, and N. Misković, “Visual diver detection using multi-descriptor nearest-class-mean random forests in the context of underwater human robot interaction (hri),” in *OCEANS 2015 - Genova*, 2015, pp. 1–7.

[16] V. Chu, K. Bullard, and A. L. Thomaz, “Multimodal real-time contingency detection for hri,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3327–3332.

[17] M. Romeo, D. Hernández García, R. Jones, and A. Cangelosi, “Deploying a deep learning agent for hri with potential" end-users" at multiple sheltered housing sites,” in *Proceedings of the 7th International Conference on Human-Agent Interaction*, 2019, pp. 81–88.

[18] Y. Kinha. (2020) An easy guide to choose the right machine learning algorithm. [Online]. Available: https://www.kdnuggets.com/2020/05/guide-choose-right-machine-learning-algorithm.html

[19] scikit learn. Choosing the right estimator. [Online]. Available: https://scikit-learn.org/stable/tutorial/machine_learning_map/

[20] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 1255–1260.

[21] P. Cunningham and S. Delany, "k-nearest neighbour classifiers," *Mult Classif Syst*, 04 2007.

[22] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *International Journal of Computer Science Issues(IJCSI)*, vol. 9, 09 2012.

[23] S. Bernard, L. Heutte, and S. Adam, "On the selection of decision trees in random forests," in *2009 International Joint Conference on Neural Networks*, 2009, pp. 302–307.

[24] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," vol. 2049, 01 2001, pp. 249–257.

[25] Y. Zhang, "Support vector machine classification algorithm and its application," in *International Conference on Information Computing and Applications*. Springer, 2012, pp. 179–186.

[26] J. Kaur, D. Singh, P. Bains, and G. Singh, "Analysis of multi layer perceptron network," *International Jounral of Computer and Technology*, vol. 7, pp. 600–606, 06 2008.

[27] N. K. Kain. (2018) Understanding of multilayer perceptron (mlp). [Online]. Available: https://medium.com/@AI_with_Kain/understanding-of-multilayer-perceptron-mlp-8f179c4a135f

[28] S. Tyagi and S. Mittal, "Sampling approaches for imbalanced data classification problem in machine learning," in *Proceedings of ICRIC 2019*. Springer, 2020, pp. 209–221.

[29] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[31] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.

[32] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *Icml*, vol. 97. Citeseer, 1997, pp. 179–186.

[33] "Two modifications of cnn," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 11, pp. 769–772, 1976.

[34] P. Hart, "The condensed nearest neighbor rule (corresp.)," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, 1968.

[35] N. Tsoi, J. Connolly, E. Adéníran, A. Hansen, K. T. Pineda, T. Adamson, S. Thompson, R. Ramnauth, M. Vázquez, and B. Scassellati, "Challenges deploying robots during a pandemic: An effort to fight social isolation among children," 2021.

[36] N. S. Jecker, "You've got a friend in me: sociable robots for older adults in an age of global pandemics," *Ethics and Information Technology*, pp. 1–9, 2020.

[37] G. Odekerken-Schröder, C. Mele, T. Russo-Spena, D. Mahr, and A. Ruggiero, "Mitigating loneliness with companion robots in the covid-19 pandemic and beyond: an integrative framework and research agenda," *Journal of Service Management*, 2020.

[38] N. Martelaro, V. C. Nneji, W. Ju, and P. Hinds, "Tell me more designing hri to encourage more trust, disclosure, and companionship," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 181–188.

[39] K. Jeong, J. Sung, H.-S. Lee, A. Kim, H. Kim, C. Park, Y. Jeong, J. Lee, and J. Kim, "Fribo: A social networking robot for increasing social connectedness through sharing daily home activities from living noise data," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 114–122.

[40] J. Hudson, R. Ungar, L. Albright, R. Tkatch, J. Schaeffer, and E. R. Wicker, "Robotic pet use among community-dwelling older adults," *The Journals of Gerontology: Series B*, vol. 75, no. 9, pp. 2018–2028, 2020.

[41] J. H. Lui, H. Samani, and K.-Y. Tien, "An affective mood booster robot based on emotional processing unit," in *2017 International Automatic Control Conference (CACS)*. IEEE, 2017, pp. 1–6.

[42] J. Li, A. H. Miller, S. Chopra, M. Ranzato, and J. Weston, "Dialogue learning with human-in-the-loop," *arXiv preprint arXiv:1611.09823*, 2016.

[43] P. Sequeira, P. Alves-Oliveira, T. Ribeiro, E. Di Tullio, S. Petisca, F. S. Melo, G. Castellano, and A. Paiva, "Discovering social interaction strategies for robots from restricted-perception wizard-of-oz studies," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 197–204.

[44] Anki. Anki developer forum. [Online]. Available: https://developer.anki.com/

[45] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org

[46] "Ros noetic ninjemys." [Online]. Available: http://wiki.ros.org/noetic

[47] "Tkinter python package." [Online]. Available: https://docs.python.org/3/library/tkinter.html

[48] P.-L. Carrier and A. Courville. Kaggle face emotion recognition (fer) dataset. [Online]. Available: https://github.com/npinto/fer2013

[49] P. Dwivedi. Face detection, recognition and emotion detection. [Online]. Available: https://towardsdatascience.com/face-detection-recognition-and-emotion-detection-in-8-lines-of-code-b2ce32d4d5de

[50] ——. Face and emotion detection. [Online]. Available: https://github.com/priya-dwivedi/face_and_emotion_detection

[51] “Pyaudio python package.” [Online]. Available: https://pypi.org/project/PyAudio/

[52] P. A. Mello, “Pitch volume detection pyaudio,” https://gist.github.com/palexandremello/ada4b7028203272677034c8f0e2aaef8, 2020.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[54] J. D. J. Gierveld and T. V. Tilburg, “A 6-item scale for overall, emotional, and social loneliness: Confirmatory tests on survey data,” *Research on Aging*, vol. 28, no. 5, pp. 582–598, 2006.

[55] D. W. Russell, “Ucla loneliness scale (version 3): Reliability, validity, and factor structure,” *Journal of personality assessment*, vol. 66, no. 1, pp. 20–40, 1996.

[56] J. D. Mayer and Y. N. Gaschke, “Brief mood introspection scale (bmis),” *Psychology*, vol. 19, no. 3, pp. 1995–1995, 2001.

Appendix A

Interview Participant Demographics

The following table outlines the demographics of the 8 participants that were used in the initial interviews.

Interview Number	**Age**	**Gender**	**Nationality**	**Residence**
1	29	Female	Danish	Single Corridor Room
2	28	Female	Taiwanese	Single Corridor Room
3	25	Female	Indian	Single Corridor Room
4	23	Male	Lebanese	Single Corridor Room
5	22	Male	Spanish	Single Corridor Room
6	26	Female	Taiwanese	Single Corridor Room
7	23	Male	Italian	Single Corridor Room
8	26	Male	Indian	Single Corridor Room

Table A.1 – Interview Participant Demographics

Appendix B

Deployment Questionnaires

Pre-Deployment Questionnaire

Please indicate how you are currently feeling based on the options below-

- I cannot tolerate being so alone
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way
- I lack companionship
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way
- I feel left out
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way
- I feel completely alone
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way
- I feel isolated from others
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way

What kind of relationship or role do you expect Cozmo to play?

Session Questionnaire Pg 1

Time: Actual Time:

Please rate your overall mood on a scale from 1-10, 1 being very unpleasant and 10 being very pleasant

1 2 3 4 5 6 7 8 9 10

Please indicate how you are currently feeling based on the options below-

- I cannot tolerate being so alone
  Strongly Disagree Disagree Agree Strongly Agree
- I lack companionship
  Strongly Disagree Disagree Agree Strongly Agree
- I feel left out
  Strongly Disagree Disagree Agree Strongly Agree
- I feel completely alone
  Strongly Disagree Disagree Agree Strongly Agree
- I feel isolated from others
  Strongly Disagree Disagree Agree Strongly Agree

Describe how you've spent the past 3 hours

Have you interacted with Cozmo in the past 3 hours?

Yes / No

Session Questionnaire Pg 2

If yes, please write down any thoughts you have about your interactions with Cozmo

Post-Deployment Questionnaire Pg 1

Please rate your overall mood on a scale from 1-10, 1 being very unpleasant and 10 being very pleasant
1 2 3 4 5 6 7 8 9 10

Please indicate how you are currently feeling based on the options below-

- I cannot tolerate being so alone
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way
- I lack companionship
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way
- I feel left out
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way
- I feel completely alone
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way
- I feel isolated from others
  Often feel this way, Sometimes feel this way, Rarely feel this way, Never feel this way

Did you enjoy having Cozmo in your room?

Yes No

What kind of relationship or role do you think Cozmo played?

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

Post-Deployment Questionnaire Pg 2

Do you think Cozmo made a difference to your day?

Out of the two days Cozmo was in your room, which day's behaviour did you prefer? Why?

Post-Deployment Questionnaire Pg 3

Do you think being part of the training process impacted your interactions with Cozmo?