# Behavioural Modeling and Analysis through Artificial Intelligence Based Hybrid Approaches

**Ph.D. (Computer Science)**

***By***

Naveed Anwer Butt
69-FBAS/PHDCS/F11

**Supervisor**
Dr. Qamar Abbas, Assistant Professor
Department of CS&SE, IIU, Islamabad

**Co-Supervisor**
Professor. Dr. Mian Muhammad Awais
Department of Computer Science, SBASSE, LUMS, Lahore

Department of Computer Science and Software Engineering
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad
(2021)

بسم الله الرحمن الرحيم

شُروع اللہ کے پاک نام سے جو بڑا مہر بان نہایت رحم والا ہے -

*In the name of ALLAH, The Most Gracious, The Most Merciful.*

INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD
FACULTY OF BASIC & APPLIED SCIENCES
DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING

Date: 06-07-2021

## Final Approval

It is certified that we have read this thesis, entitled **"Behavioural Modeling and Analysis through Artificial Intelligence Based Hybrid Approaches"** submitted by **Mr. Naveed Anwer Butt No. 69-FBAS/PhDCS/F11**. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for the award of the PhD. degree in computer Science.

## Committee

**External Examiner:**
Dr. Adnan Sohail,
AssociateProfessor
Department of Computer Science
IQRA University, Islamabad

**External Examiner:**
Dr. Aun Irtaza,
Associate Professor
Department of Computer Science
UET, Taxila

**Internal Examiner:**
Dr.Asim Munir,
Assistant Professor
Department of Computer Science & Software Engineering
FBAS, IIUI

**Supervisor:**
Dr. Qamar Abbas,
Assistant Professor
Department of Computer Science & Software Engineering
FBAS, IIUI

**Co-Supervisor:**
Prof. Dr. Mian Muhammad Awais,
Professor
Syed Babar Ali School of Science and Engineering
LUMS, Lahore

for

# Acknowledgment

All praises and thanks to **ALMIGHTY ALLAH**, Lord of the entire world, the most Merciful, **Who** blessed me with knowledge, courage and strength to complete this humble task. All respects for **HAZRAT MUHAMMAD (PBUH)**, whose teachings are the permanent source of guidance for us. I deem it my utmost pleasure to express my heartfelt gratitude and profound benediction to my great research supervisor, **Dr. Mian Muhamad Awais**, for his patience, immense knowledge, persistent guidance, and encouragement. Despite his busy schedule and many responsibilities, he always made time for my research. This dissertation would not have been possible without his encouragement and generosity. His constant support and meticulous guidance helped me in the whole gamut of research operations and then the writing of this thesis. I could not have dreamed of having a better mentor and advisor for my Ph.D. study. I am equally gratified for his kind and compassionate behavior towards me; which I mostly came across in our research meetings, online discussions, and also upon committing mistakes. The way he guided me with his technical expertise to creatively think and solve the problems is really a fabulous learning experience in my entire academic career. His protracted, untiring and unstinted support to let me manage my academic life successfully; lie at the crux of my visionary growth and development.

I would also like to show my appreciation to my second supervisor, **Dr. Qamar Abbas**, who gave all his knowledge, guidance and support to boost my confidence and learning. His mentoring and encouragement have been especially valuable, and his early insights launched the greater part of this dissertation.

This acknowledgement would be incomplete without mentioning my parents, **Mr and Mrs Abdul Ghafoor Butt (late)** and most loving mother in law **Mrs Shagufta Jabeen (late)**, whose love, affection and prayers have always been with me just like my own shadow, right from the beginning when I was just a grass root, up to this level of higher education in order to make this day possible. Words are insufficient to express my love and appreciation to my lovely sister, **Mrs Saima** and my supportive brothers in law, **Dr. Asad Imtiaz Khan** and Brother **Mr Nadeem Butt**, for their great support, courage, strength and everything else they have given to me. I owe my sincerest gratitude to my sweetheart wife, **Mrs Sadaf**, for their understanding and endless love at all times.

Also, I would like to express thanks to my father-in-law, **Mr Imtiaz Khan** and **all family members** and **friends** specifically **Dr. Ghulam Mustafa, Dr. Fayyaz Ahmad** for standing by me, supporting me and boosting my morale in hard times during my study.

# Declaration

I, Naveed Anwer Butt, hereby state that my Ph.D. thesis titled **Behavioural Modeling and analysis through Artificial Intelligence Based Hybrid Approaches** is my own work except where due reference is made in the text and it has not been previously submitted by me for taking partial or full credit for the award of any degree at this university or anywhere else in the world. If my statement is found to be incorrect, at any time even after my graduation, the university has the right to revoke my Ph.D. degree.

**Naveed Anwer Butt**

**69-FBAS/PHDCS/F11**

# Dedication

***To My late Parents, Wife, Sister,***

***Brother,***

***In Laws and***

***Friends***

# Abstract

This research focused on the endowment of machine learning and hybrid artificial intelligence-based approaches in the domain of imitation of human-like behavior while playing video games without compromising the imitative performance of imitation agent and behavioral neuroscience through functional magnetic resonance imaging (fMRI). To provide a viable solution for the human-like agent movement and to explore the cognitive state classification for decision-making and patterns of activation from the prefrontal cortex of human brain, different variants of artificial neural networks (ANNs) were described and compared to the adaptive neuro-fuzzy inference system (ANFIS). Furthermore, a lazy learning technique such as Case-based reasoning (CBR), has also been used to imitate human-like behaviour. The purpose is to reduce the time required to search the case base without affecting the diversity with mininmal cases.

This study reported on two challenging case studies. To imitate human-like behavior and to preprocess the case base, this study used simulated Pac-man game data and Functional Magnetic Resonance Imaging (fMRI) data to examine the neural basis of rewarding human decisions. Two novel algorithms have been developed to deal with the process of reducing features and cases. For the proposed algorithms, specific techniques such as rough sets and FP-based methods are developed, tested and compared.

The generalized regression neural network (GRNN) surpasses eight neural network architectures by showing best results with the smallest variance in all experiments for both selected case studies. The results reveal the decisive victory of the ANFIS hybrid intelligence system as the best in prediction and classification with very high accuracy for both selected case studies. The consistency of the experimental results of the applied approaches is also observed.

This thesis also examines, with game data, how the imitation of a case-based agent can be improved by pre-processing the case base? In order to develop a competent CBR system, the experimental results are auspicious through case knowledge extraction.

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| AI | Artificial Intelligence |
| ANNs | Artificial Neural Networks |
| ANFIS | Adaptive Neuro-Fuzzy Inference System |
| BT | Behavior Tree |
| CBM | Case Based Management |
| CBR | Case Based Reasoning |
| CS | Cosine Similarity |
| DPD | Dependent Personality Disorder |
| DQN | *Deep Q Network* |
| DTDNN | Design Time Series Distributed Delay Neural Networks |
| EAs | Evolutionary Algorithms |
| EEG | Electroencephalography |
| EPI | Echo Planar Image |
| IRL | Inverse Reinforcement Learning |
| FCM | Fuzzy C-Means Clustering |
| fMRI | Functional Magnetic Resonance Imaging |
| FSC | Fuzzy Subtractive Clustering |
| FSM | Finite State Machine |
| FF | Feed-Forward Multi Perceptron |
| FFTD | Focused Time Delay Neural Network |
| GE | Genetic Algorithms |
| GNB | Gaussian Naive Bayes Classifier |
| GLM | Generalized Linear Model |
| GSP | Generalized Sequential Pattern Algorithm |
| GRNN | Generalized Regression Neural Networks |
| HBMs | Human Behavioural Models |
| HMM | Hidden Markov Models |
| KNN | K- Nearest Neighbor |
| MFs | Membership Functions |
| NARX | Nonlinear Autoregressive Model |
| LfD | Learning From Demostration |
| LDC | Linear Discriminant Classifier |
| LRN | Layer Recurrent Network |
| MLP | Multi-Layer Perceptron |
| MNI | Montreal Neurological Institute |
| NPCs | Non-Player Characters |
| PBIL | Population-Based Incremental Learning |

| | |
|---|---|
| RL | Reinforcement Learning |
| RoI | Region of Interest |
| SVM | Support Vector Machines |
| SPM | Statistical Parametric Mapping |
| SOM | Self-Organizing Maps |
| SPADE | Sequential Pattern Discovery Using Equivalence Classes |

# Research Contribution

The following research papers related to this thesis are published/submitted in international journals during the Ph.D. research.

**Published**

1. Butt, Naveed Anwer; Awais, Mian Muhammad; Abbas, Qamar "Improved Diagnostic Accuracy in Dependent Personality Disorders: A Comparative Study of Neural Architectures and Hybrid Approaches on Functional Magnetic Resonance Imaging Data", Journal of Medical Imaging and Health Informatics, Volume 9, Number 4, May 2019, pp. 697-705(9), DOI: https://doi.org/10.1166/jmihi.2019.2642

**Submitted**

2. Naveed Anwer Butt, Mian Muhammad Awais, Qamar Abbas, "Towards imitation of human-like behaviour in virtual worlds: Experimental comparison of neural architectures and adaptive neuro-fuzzy inference system (ANFIS).

3. Naveed Anwer Butt, Mian Muhammad Awais, Qamar Abbas, "A Case-based Reasoning framework for believable AI Agents behaviour in Virtual worlds", An International Journal Applied Artificial Intelligence.

# Chapter 1: Introduction

## 1.1. Overview

In computer science, especially in terms of human-human and human-machine interaction, the analysis, synthesis, and simulation of human behaviour requires a great deal of effort. However, this has been far from being resolved and governed by social, psychological, and cognitive, phenomena's which is not observed. This thesis study was inspired to address issues of imitating human-like behaviour and behavioural neuroscience to provide viable solutions to synthesize a more precise classification and prediction model. Furthermore, this research also examines the impact of the pre-processing in the transition of raw cases without compromising the imitative performance of the imitation agent. The research hypothesis presented is to investigate the effectiveness of machine learning techniques in a hybrid way. The significance of the hypothesis is based on the reasoning that artificial intelligence-based hybrid approaches have not been fully exploited by the application of imitation learning and neuroscience behaviour. Artificial intelligence-based hybrid approaches are also emerging as a dominant approach to construct a computational model to reduce model-based programming efforts intuitively.

The focus of this thesis is on developing a common framework for issues relevant to human player behaviour and explores the cognitive state classification for decision-making and patterns of activation from the prefrontal cortex of human brain. This framework allows pervasive computing environments that can predict user actions and intentions, automatically adapt to the behaviour of their users, and human behavioural agents that supplement our own behaviour and goals. It is really fascinating to observe that human behaviour provides a real learning experience for predicting their next act. This research also proposes two novel algorithms to pre-process the case base used by an imitative agent, in the computer game domain. These algorithms offer a significant improvement in imitation ability compared to the case base that is not processed.

To find the most activated brain regions, their specific cognitive processes and stimulus-triggered activity, several experiments were designed by using functional brain imaging data. These experiments were conducted with the aim of predicting the subject's cognitive status through training classifiers based on fMRI sequences. Although this is a new field of research, many studies have been conducted to find a unique association between fMRI data and the cognitive status of the human brain. In fact, if such an association exists, the applicability of machine learning techniques can lead to the construction of "intelligent machines" that function as lie detectors or mind readers, i.e., machines that can help patient behavior and evolution in mental activity.

## 1.2. Research Motivation

The world is currently on the verge of a ubiquitous agent era. These agents with autonomous characteristics assist in almost all endeavors of our today's daily life e.g., at home, at work, in games and in the social environment. But today's agents are perceived more "mechanical" or "robotic ", which leading to frustration, inappropriate expectations and more importantly failure in engagement and training. Reliable methods to create more realistic and believable agents can ultimately help to create human-like peculiarities, such as learning ability, decisions, mistakes, adjusting their own strategies by reacting to actions of opponents, and so on.

The ability of agents to rely on artificial intelligence to behave like humans raises many future technological trends[1]. Various techniques are available for learning such as Artificial Neural Networks (ANN), Reinforcement learning (RL), Fuzzy Logic, Genetic Algorithms (GA), Hidden Markov Models (HMM), Support Vector Machines (SVM), etc. For example, ANN universal problem solvers [2] provide a strong set of neural architectures and compatible algorithms for supervised learning. ANN has shown a good affinity for implementing an agent. It is worth mentioning that AI-based trained agents using machine learning algorithms deems to behave non-human like manner unnaturally or mechanically. So far, success is far from guaranteed when attempting to create believable agent. There are several neural architectures grouped under the ANN umbrella and compatible hybrid learning algorithms. However, certain architectures may function better for certain learning problems than others. Therefore, it is a real challenge to find the best architecture and training method. With the proliferation of tools and applications for developing, there is a need for faster, automated means to implement them. The aim of this research is taking those theories and build an equivalent model and compare the results in two application case studies i.e., Game AI and Cognitive Behavioural Neuroscience.

## 1.3. Human Behaviour Analysis and Modeling in Different Environments

### 1.3.1 The current state of AI in Games

In recent decades, the purpose of video games is rapidly changing from entertainment to education and training due to advent of gaming technology. This impressive development in virtual worlds presents new challenges for the creation of artificial intelligence-controlled agents (AI-agents). Video games provide an excellent testbed and are increasingly become the domain of choice for evaluation and showcasing new Artificial Intelligence (AI) techniques. The focus of small but growing community of researchers has shifted to imitate video game characters/players to increase the perceived value of entertainment and satisfaction [3]. This requires player modeling to assess personal experience in games accurately. Behavioral modeling of players is almost a necessity when the goal of AI is to entertain the human player and not to defeat the human player. Such AI-agents can also be used to autonomously play the roles of the virtual characters also known as non-player characters (NPCs), in the training scenario. Virtual training systems give a real soldier the opportunity to play the game and interact with computer-controlled characters that can simulate either friend or foe. Therefore, it is crucial for the realism and effectiveness of the exercise that NPCs demonstrate realistic human behavior and adapt to situations based on environmental factors, capabilities, and priorities.

Game competitions are being organized to develop AI-agents capable of imitating human-like behavior. The game industry needs AI-agents to behave less “robotic” and more human-like playing styles. Computer games like chess, fighting or soccer required this scenario. There is a natural assumption to choose either human or AI-agent as an opponent. The human likeness of AI-agent is associated with the principle of believability [4] which is characterized by human-like peculiarities, such as its ability to learn, make decisions, mistakes and adjust its strategies by responding to the opponents' actions. [5]. Stewart et al. [6] derived knowledge of the player's in-game behavior can ideally result in improved gaming experiences for players irrespective of gender, age or experience [7], gaming testing and game design practices, as well as better monetization and marketing strategies.

Fruitful, attempts have been made for the development of AI-agents that can imitate human-like behavior, in the games [4, 8, 9]. To develop autonomous and assisted model players, Ad-hoc authoring, tree search, evolutionary computation, supervised learning, reinforcement learning, and unsupervised learning are most prominent AI techniques which have been used to achieve this goal [10]. Imitation and prediction are two main game player modeling subtask and have achieved mainly via supervised and unsupervised learning methods. Unsupervised techniques, i.e., clustering has been used for behavioural classification depending on their attributes and association mining for finding frequent patterns of actions to determine player behaviour in the game. Supervised learning and reinforcement learning appear to be a dominant trait in player behaviour modeling and believable agent development. Neuroevolution, scripting techniques, finite state machines, and rule-based approaches are also in use by developers. These approaches exhibit some limitations; unable to generalized unseen circumstances, dependence on fitness function either during the evolution of human-like agents or evaluation of the human-like AI-agent.

### 1.3.2 Behaviour Analysis and Behavioural Neuroscience

Behavioral analysis focuses on behavior as a standalone subject, rather than as an index of cognitive events, and thus is not dualistic. Behavioral analysis involves several learning laws learning discovered by researchers using experimental designs with a single object. We argue that by investigating the neural bases of behaviors that can be described in cognitive terms, behaviour analysis can provide neuroscientists with a novel experimental and a theoretical framework.

The nervous system has evolved to perform two functions related to the ability of an organism to interact with its environment: to recognize energy changes and to control movement, with specific sensory and motor areas of the cortex dedicated to each of these functions. Therefore, behavioral analysis as an indicator of derived cognitive structures or processes, is best suited to explain this interaction sparingly. For this several different types of brain activity measurements and signal recordings may be investigated in real time such as electroencephalography (EEG) and Functional Magnetic Resonance Imaging (fMRI). Since fMRI has proven to be one of the best for collecting large data of human brain activity by various studies and being entirely noninvasive, therefore multiple studies on a single subject can be performed.

The triggering of systematically cross-linked activated brain regions, has itself established a new trend in fMRI imaging through the analysis of multi-voxel patterns. The key observation of these studies is the brain regions which are activated for specific mental activities. Due to the lack of tools, previous attempts to investigate the collaborative functioning of the human brain have been hampered. The advancements in instrumentation and in analysis techniques, have enabled a complete examination of brain while performing specific tasks. The high-resolution fMRI can access specific areas of the brain that are activated when a task is performed.

We hope that, for the future, closer alliance between neuroscience and AI researchers, as well as the identification of a common language between the two, will enable a righteous cycle in which research will be accelerated by shared theoretical insights and joint empirical progress[11]. For algorithmic construction, refining intelligence by comparison with the human brain could provide intuition to understand the secrets of the mind, such as dreams, the essence of creativity, and even the consciousness of one day.

### 1.3.3 Classifying Cognitive States from fMRI Data using Neural Networks

The human brain can process various kind of impulses cause activation and functioning of multiple organs in the body. The functional connections of the brain regions enabled the brain to accomplish these tasks [12]. The fMRI is considered to be a powerful experimental technique for analyzing stimulus-based activation in the brain. The triggering of systematically cross-linked activated brain regions has established itself as a new trend in fMRI imaging through the analysis of multi-voxel patterns. The key observation of these studies is that brain regions are activated for specific mental activities. Due to the lack of tools, previous attempts to investigate this collaborative mode of functioning of the human brain have been hampered. Recent advances in instrumentation and data analysis techniques have enabled a complete brain examination while performing specific tasks [13]. The high-resolution fMRI can access specific areas of the brain that are activated when a task is performed.

To perform such a fMRI data analysis, Friston et al. [14] developed an SPM package that maps the activated brain areas of the fMRI data. Neuroimaging determines the relationship between the responses of a particular task to the predefined associative areas. This is possible because, during the execution of a particular task, tandem activations in specific brain regions are performed [15]. While performing a particular task, it is necessary to isolate the resolution of brain regions from induced activations in brain regions.

To find strongly activated brain regions, their specific cognitive processes and stimulus-triggered activities, several experiments were designed using functional brain imaging data [16-18]. Identification of signals from high-resolution fMRI data allows us to classify them into pre-defined areas of the brain, such as the visual area or decision area. Belilovsky et al., proposed a Voxel wise autoregressive study for fMRI to locate activations[19]. Since the image (fMRI) contains thousands of voxels, the analysis of such a data-volume leads to an over-fitting of the classification algorithm[20], high level of noise and curse of dimensionality in the field of radiology [21]. To accurately assess neural activation, the most popular techniques of Artificial intelligence technologies, i.e., machine learning and convolution neural networks were applied to decipher brain patterns for medical image analysis. For a more accurate classification, state-of-the-art ensemble techniques were employed [22, 23].

The artificial neural network has been applied to a wide range of fMRI problems and to develop diagnostic models. To identify the complex interactions among input data, ANN distinguishes the mechanisms of time series and results. This makes it possible to recognize the hidden relationships which are usually invisible during traditional statistical analysis. ANN seems to be a promising tool for clinical decision making and have been in use for several areas, such as Alzheimer's disease [24] , cognitive state classification [17], fMRI pattern classification [25], face recognition [26] and word prediction from fMRI data[27].

Different approaches have adapted to the cognitive recognition problem namely: linear Discriminant Classifier (LDC), linear Support Vector Machine (SVM), Polynomial Support Vector Machine, k- Nearest Neighbor (KNN) and Gaussian Naive Bayes Classifier (GNB). The literature review reveals that in tackling problems related to neuroscience there is potential for implementations of hybrid artificial intelligence techniques. There is also a greater need to recognize the appropriate Artificial Intelligence algorithms used in cognitive neuroscience. Also, the hybrid algorithm domain needed to be explored. The ensemble/hybrid techniques have recently gained more attention in computational neuroscience due to their exceptional advantages in processing with complex data structures [22, 28].

However, hybrid techniques have been proposed as tools to improve the classification performance. They aim to combine the results of different classification members trained for the same task, which is one of the most popular fields in pattern recognition. This integration improves overall accuracy compared to any single classifier [29]. One of the most significant used hybrid intelligent system is the neuro-fuzzy combination. These systems use fuzzy logic to model the environment, having non-linear vagueness on data [30], interconnected neural network processing elements and information connections[17].

## 1.4. Problem Statement

During recent times, prompt growth has been carried out in the allied arenas of imitating human-like behaviour and human neuroimaging technology. Artificial Intelligence offers extensive tools for machine learning and has been broadly used in contributions to these fields. However, artificial intelligence learning tools have generally been used as standalone solutions for imitation and neuroscience. Recently, the related fields of imitation learning and neuroscience using artificial intelligence have achieved rapid growth. Many of the early explorer were straddling these two areas, as collaborations between these disciplines being proved very productive. However, interactions have become less common lately because the complexity of the two topics has increased significantly and the boundaries of scientific disciplines are getting stronger. The growing importance of neuroscience and imitation learning is critical to generating and accelerating ideas in AI research [31]. To gain insight into innumerable important facets of general intelligence on a higher level, the study of cognition and its neural implementation also plays an important role. The development of AI has two advantages in scrutinizing general biological intelligence. First, imitation learning and neuroscience are a rich source of stimulation for the application of new algorithms and architectures, as biological computations have been viewed as critical in supporting cognitive function. This will be an excellent test field for integration into artificial systems. Second, both domains can be used as a test environment to validate existing AI techniques. Indubitably, from a practical viewpoint when setting up an AI system, we don't have to indiscriminately force compliance with biological believability. What interests us is a system for understanding the neuroscience level of the brain, to be precise the algorithm, the architecture, and the functions. While concentrating on computing and algorithmic levels, we get insight that can be transferred to the normal mechanisms of brain work, while leaving silico in place of intelligent machines to accommodate unique opportunities and challenges. Therefore, there is a greater need for an extensible framework that addresses the issues validated in both areas. The framework should also be tested in both areas to compare performance with existing applications and to get results from a minimal and many trails. This will create a stronger, theoretically sounder basis for applied AI and advance the field in solving its outstanding problems. This will also help empirically verify an unverified subset of the theory from the imitative and neuroscience.

## 1.5. Research Aims and Objectives

The primary aim of this research is to investigate how to implement AI based hybrid approaches for behavioural modeling, analysis and cognitive neuroscience of human decision making.

For this purpose, the following objectives are considered important:

**Objective 1:** Propose and implement the framework for developing an AI-agent that imitate human behaviour perceived as an actual human by using video game data of human players.

**Objective 2:** Improving the Performance of Case-based Imitation Agent through Case Knowledge Extraction proposing two novel algorithms without compromising the imitative performance of imitation agent.

**Objective 3:** Development of a model in the domain of behavioural neuroscience, to find strongly activated brain regions, their specific cognitive processes and stimulus-triggered activities during rewardless-related decision making using functional magnetic resonance imaging (fMRI) data.

## 1.6. Contributions

In the context of the suggested solution, several contributions have been made successfully during this study, considering the section above. They are classified as follows.

1. An extensible framework to produce human-like agent movements in computer game to imitate individual human winning and losing behavior to intuitively reduce model-based programming effort using precise machine learning techniques, such as variants of Artificial neural networks, Neuro-Fuzzy hybrid system, and Case Based reasoning. A conversion process of raw logs to higher-level representation has also been proposed to develop an automated process for generating a corpus to perform experiments with reproducible results using standard algorithms.
2. To derive a measure of imitation of human behavior from the likelihood ratios between the models and which training method gives the best results with a comparatively smaller dataset so that model can be learned in smaller time.
3. A case-based framework is proposed to improve the state of the art for evaluation of imitation learning using similarity measure from the likelihood ratios which gives the best results after comparing seven standard similarity measures. Subsequently,

proposed evaluation strategy is tested agent to imitate the normal playing style of a population of human players in the Pac-man game in case of winning and losing games.

4. Two new algorithms have been proposed by extracting the non-informational information without affecting the performance of the Imitation Agent through Case Knowledge Extraction. To reduce the case base, first algorithm is designed using rough set-based feature approach, based on the principle of indiscernibility relations. The second approach is developed using the concept of Frequent Patterns to distinguish pivotal cases from the initial cases and reduce the case base.
5. Developed an extensible framework to find and classify the strongly activated brain regions during rewardless-related decision-making using fMRI Data of human brain region using precise machine learning techniques, i.e. variants of Artificial neural networks and Neuro-Fuzzy hybrid system.

## 1.7. Thesis Organization

**Chapter 1** presents the research hypothesis, the research goals and objectives, the introduction of related work to correlate the fertility of such studies, and the research methodology to be used.

**Chapter 2** of this thesis address the literature review of the AI based hybrid approaches in the domain of imitation of human-like behaviour in video games and behavioral neuroscience to correlate the aim of this thesis that will be used as case studies.

**Chapter 3** address Objective1 by describing the framework for agent imitation of the selected case study used in this thesis.

**Chapter 4** address Objective 2 by providing some preliminary knowledge of rough set and FP-growth theory which is used in development of the techniques for case knowledge extraction. Using the concepts of rough set and FP-growth, an effective method is developed to reduce the size of case base. Some experiments are performed using player's game data and real-life data sets to evaluate the performance of developed algorithms.

**Chapter 5** address Objective 3 by designing and implementations of models in the domain of behavioural neuroscience, to find strongly activated brain regions, their specific cognitive processes and stimulus-triggered activities during rewardless-related decision making using functional magnetic resonance imaging (fMRI) data.

**Chapter 6** gives conclusions, summarizes the research, and outlines future work areas.

# Chapter 2: Literature Review

## 2.1. Overview

In this section we have thoroughly investigated literature on popular AI algorithms, their roots, and applications in Human Behaviour studies. The literature review has mainly focused into application categories of computer games, and neuroscience of Human Behaviour domains.

## 2.2. Game AI (AI and Games)

Artificial Intelligence (AI) has been considered from the beginning as "a machine that is able to reason" [32, 33]. During the last decade, the advancement of AI has furnished diversifying effects on several aspects of computing. Additionally, the rapid advances in AI are associated with increased computing power due to hardware advances. AI success story can be experienced in everyday life through many practical applications, enabling a better understanding of images, web searching, autonomous cars, game-playing, AI-assisted game design and many others. Some of the machines have the status "human-level" or beyond.

AI researchers in the early days envisioned the computer systems that show human intelligence for decision-making and problem-solving. A set of formal mathematical notations were presented to machines to achieve this capability rather than symbolic representation. After the incursion of formalized symbolic representation, AI succeeds in many domains. Of course, games - especially board games - was a popular domain for early AI experimentation because it is a highly constrained and complex decision-making environment. Over the years AI poses questions to ask such as: "How can AI detect and express emotion?" and "How can AI educate people and be creative?". How AI can play games that have never been seen before?", "How can AI learn from a minimal number of trials?" All these questions present AI challenges and tasks that are not easy to formalize or determine objectively. Again, games have created a popular domain to study such abilities because they have aspects of a subjective nature that cannot be easily formalized. Some of these consist of experience of game playing or the creative process of game design [34].

Games have offered excellent testing beds since the birth of the artificial intelligence and increasingly become the domain of choice for evaluation of AI techniques, optimization, and exemplar new algorithms. Simultaneously, video games provide a dynamic, competitive and reproducible environment that does not depend on external factors [32, 35]. These materials have made games a distinctive and popular area for the study of AI. But not only the AI is progressing by games; games as well has advanced through AI research. We think that AI helped the game get it better on some fronts: in the way we play them, to understand their inner functions, design, and interaction.

## 2.3. Problems and limitations of Digital Game Environment

The research has shown that the presence of several game agents in the game environment increased the interest of player's where the game environment is dynamic and non-deterministic. These kinds of environments possibly have discontinuous inputs [36, 37]. The agents must respond quickly to changes in the game environment due to the nature of the real-time game. This imposes a requirement for each learning technique used to be able to utilize potentially useful disconnected input features [36, 38] .Generally, digital games contain a large number of internal states, with each state associated with a corresponding set of actions. Therefore, the use of exhaustive search techniques during learning in such a high-dimensional space is impractical. The nature of game learning is unpredictable, potentially resulting in orthodox game agent behaviour and is one of the key issue of the commercial game industry [39] .

This has the consequence that game developers have no further control during the learning process and thus can provide an implausible and inadequate gaming experience. Subsequently, the developer must define the constraints on the learning process to ensure that the behaviour of the learned game agent does not cause the game to deteriorate or worsen the player's overall gaming experience [40]. Inappropriate or inconsistent game agent behaviour may also result in increased development cost, resources for error reproduction and testing. The use of learning can moderate the associated resources in the development of game agent behaviour, and can also be used to detect flaws and variability in game agent behaviour [40].

The computer game industry expects agents to learn behaviour that is neither mechanistic nor predictable throughout the game. Similarly, the behaviour of the game agents must be believable, visible and appealing to the player and not just show the optimal behaviour [36, 41].

The level of visibility given by the learning process to game designers and developers should also be considered. At the very least, to reconcile AI games with designers and developers, meaningful representation schemes are highly desirable in order to enable them to understand the learning process. In order to make the development process fruitful, the simple and uncomplicated learning techniques should be implemented and maintained for various AI learning tasks [42, 43].

## 2.4. Human Behaviour Modelling in Gaming using Agents

In recent decades, video games have increasingly produced complex virtual worlds. This eminent development of virtual worlds presents new challenges for the creation of artificial intelligence (AI) controlled virtual agents. The term game agents and NPC's (Non-Player Characters) are often used interchangeably in-game AI research. Nevertheless, considering the many years of agent research, NPCs means characters who are not human players.

The development of intelligent agents for video games is a practical application of human behaviour modeling and has become the subject of special interest to researchers in recent years. AI research is focused in imitation human-like behaviour while playing video games to increase the perceived value of entertainment and satisfaction. Game competitions are being organized to develop AI-agents capable of imitating human-like behavior. In some ways more researchers are working on AI than ever before. Video games has proved attractive because it poses many challenges, thus, alleviate the development of new learning methods in an environment that are easily reproducible. The video games consists mainly of real people represented as human-controlled agents as noted by [44, 45] and may also be populated by NPCs (computer-controlled agents).

The NPCs has been used for different research purposes [46] and their role depends on the nature of the given world of computer games.

A new trend in computer games is the development of AI systems that can replace human players. This scenario commonly exists in computer games like chess match, fighting game, or a soccer tournament. There is a natural assumption to choose either human or computer-controlled opponent. But these games demand that AI agents imitates and control the character to user satisfaction. Although user satisfaction is unclear, it is not necessary for game characters to be the best in successful AI systems [47]. The question of how to improve the quality of the virtual world is a challenge and an exciting opportunity in this field. The quality of the AI agents is not only depending upon the ability to achieve own goals in the game, but to make the game enjoyable and more realistic. If the endowment of AI in games does not make the game better, the notions of intelligence are irrelevant. The overall game enjoyment factor is not easily revealed and has been the subject of many decades of research activity. A number of authors have expressed this issue specifically in their work [33, 48]. This research reveals that more pleasure comes with non-trivial adaptive opponents, which that they have different behavioral patterns and can respond to changes in human strategy. Many researchers have tried to find other features that make AI an enjoyable opponent in addition to its adaptability. It was concluded that human-like computer controlled opponents are one of the main attribute of a successful AI system [49]. The main Game AI research areas are Generate Content (autonomous content generation and assisted content generation) and Model Players includes player experience modeling and player behavior modeling [50] as shown in Figure 2.1.

Back propagation (BP) is currently the most common approach for training of neural network parameters. However, there are many approaches for tuning the hyper-parameters, including evolutionary algorithms (EAs). Specifically, a synergistic approach is the use of a memetic algorithm (MA) in which evolution is run as an outer optimization algorithm, and individual solutions can be optimized by other means, such as BP, in the inner loop [12]. And for exploration and global search properties, the MA can used by BP's efficient local search properties.

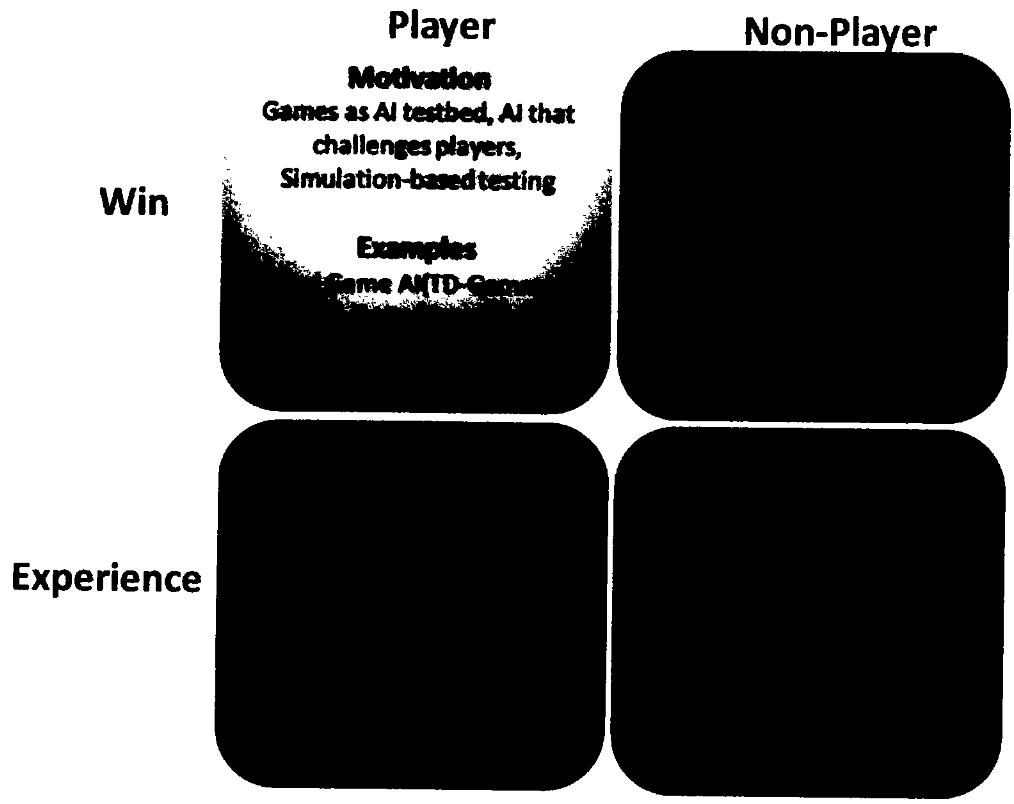



**Figure 2.1: AI Roles (Player, Non-Player), Goals (Win, Experience), and aims to play games.**

PBT, used to train agents in AlphaStar, is an MA using Lamarckian evolution [51]. In the inner loop, neural networks are constantly trained with BP, and in the outer loop, networks are chosen using one of several methods of selection (e.g., binary tournament selection), with the winner overwriting parameters of the loser. The loser therefore gets a mutated clone of the hyper-parameters of the winner. Initially, PBT was seen on several supervised learning and RL functions, tuning networks with better efficiency than had previously been accomplished. It may be particularly useful in problems with highly non-stationary loss surfaces, such as deep RL, because hyper parameters can be tune on the fly.

### 2.4.1 Believability of Agents in Video Games

Recently, it is noticed that the focus is shifting to the design of video games for individuals that can replace human players. An advantage of this type of player-centric design is ideal to enhance gaming experiences for players, irrespective of gender, age or experience [7,52,53]. The literature reveals that player-centered game design is limited to usability and testing, maintenance, player support system and empower players with control. To increase the player's engagement and enjoyment, the perspective of AI researchers is consistent with the perspective of game designers [54] . Advances in AI and game design enable the development of completely new types of games [55]. This requires player modeling to accurately assess personal experience in games. Behavioral modeling of players is almost a necessity when the goal of AI is to entertain the human player and not to defeat the human player.

To achieve this goal, an AI-Controlled agent should be believable. Identifying individual AI traits that lead to believability has many motivations in the field of research. [56] and [57] points out that the human-likeness of AI is indeed a very important reason of fun in games but this is not the only factor. A Human-like controlled agent also contributed in case of believable opponents [33].

To achieve more realism and unpredictability to increase the entertainment for the player, it might be a good approach for agents to imitate human behavior. The computer gaming industry expects computer-controlled game characters to behave less "robotic" and more human-playing styles. Ortega et al. [4] describe human likeness of AI-controlled agents associated with the general principle of believability. A believable agent is typically characterized by human-like peculiarities, such as the ability to learn, make decisions, make mistakes, and adjust their own strategies by responding to opponents' actions [5].

These characters should respond subtly to events in the environment. The ability of agents to rely on artificial intelligence to behave like humans, raises the many future technological trends [1]. It is worth mentioning that hand-coded controllers and AI based trained controllers to play a game using machine learning algorithms, deems to behave non-human like manner unnaturally or mechanically. The similar susceptibility is reported by [58]. So far, success is far from guaranteed when attempting to create human-like controllers, despite even well-playing ones. There are several reasons for substantial growth in the development of intelligent agents for video games play capable of imitating in a human-like manner [3].

Characterized the behavior of a software agent to map its observations into actions from demonstrations over a period [59] to do a task with respect to the surrounding entities [60]. To carry out such behavior require much technical knowledge, domain expertise and many of human testers to test the quality of the game.

### 2.4.2 Machine learning techniques within digital game research

Mostly, the approaches used to develop AI agents are generally traditional i.e. a rule based pre-programmed methods used in counter-strike bots by [61] and finite state machine is implemented in Halo 2 AI game by [62]. Finite state machines, behaviour trees are the most popular ad-hoc behaviour authoring methods for game development, successfully used in games as [63]. These techniques are simple to implement, visualize and debug. However, these are extremely complex to design and are computationally expensive. For non-player characters in -game AI processes of decision-making and control have been a major focus of these methods. until the mid-2000s [64]. Although, handmade behavioural scripts based mostly on AI game systems.

Manual design of AI behaviour, unable to effectively handle increasing complexity and more difficult to create the corresponding data structures and algorithms. To overcome the drawbacks of these techniques, a well-known alternative is to use some variants of machine learning to create AI-controlled agents and adapt their behaviour.

Fruitful, attempts have been made for the development of AI-agents that can imitate human-like behavior[4, 8, 65-67]. To develop autonomous and assisted model players, Ad-hoc authoring, tree search, evolutionary computation, supervised learning, reinforcement learning, and unsupervised learning are most prominent AI techniques which have been used to achieve this goal [10, 68, 69].

### 2.4.3 Direct vs Indirect Imitation

All these attempts attain different degrees of success. This plenty of work could be categorized broadly into direct and indirect imitation [4, 8, 9]. In direct imitation, mostly supervised learning is used by developers to train the agent to follow the same actions that the human player took under the same situation. The designed agent trained on the human game playing traces as input. The selection of input parameters is based on the environment of player character and output being the action of the player at the same state of the game. A major problem is the generalization of unseen situations that lead to an action that is very different from what the player would have taken in the same situation. The main reason is this training aimed at repeating the same decisions of the given player in the same state of play, and not playing well in terms of game goals. Indirect imitation was proposed as an attempt to overcome the generalization problem by using some form of Reinforcement Learning (RL). An optimize reward function used to measure the human-likeness of agent's behavior.

### 2.4.4 Supervised Learning

Supervised learning appears to be a dominant trait in player behaviour modeling and believable agent development. Supervised learning try to derive optimized function which must have the ability to map unseen and new situations to achieve better generalizations Mostly, the approaches used to develop AI agents are generally traditional, i.e., rule-based pre-programmed methods used in counter-strike bots by [61] and the finite state machine is implemented in Halo 2 AI game by [62, 63, 70]. These techniques are simple to implement, visualize and debug. However, these are extremely complex to design and are computationally expensive [64]. Numerous supervised learning algorithms, i.e., artificial neural networks, decision tree learning, regression, k-nearest neighbors, and support vector machines have been used in game AI [71]. It is a fact that there is not a single supervised learning algorithm that works best for all supervised learning problems from the various supervised learning algorithms [72].

#### 2.4.4.1. Neural network approaches to learning in digital games

The neural network approach remains popular to learn from observation and contributions are still in progress, such as the work of [73]. ANN reinforced with compatible algorithms, and a range of neural architectures is generally proven as universal problem solvers for supervised learning that are able to provide a robust model approach to learning discrete, or vector-valued objective functions [74].To predict and model players' actions and intentions, ANNs are of course suitable and implemented in robotics and games to find patterns of behaviour [4, 75]. There is much neural architecture and finding one that suits a specific learning problem is a challenge. Therefore, it is useful to find the optimal neural architecture by evaluating various options.

Various approaches of neural networks have been used in academic game research literature for the game agent controlling generation. In most cases, the multi-layer perceptron (MLP) Network architecture[76] is used, where learning is performed using either the Backpropagation algorithm [77], the Levenberg-Marquardt variant of back propagation [78], or, more often, the use of evolutionary methods. Mostly, ensemble-based network approaches implemented by using MLP networks. self-organizing Maps (SOM) [79] approaches to game agent control were also tried. Human player game traces have been used primarily for modeling players in supervised and unsupervised learning methods.

In a study led by Yannakakis et al. [80], used an MLP network to generate a game agent controller within a multi-agent test bed called FlatLand, where game agents compete to achieve randomly positioned targets and avoid collisions. The performance of the game agents during the experiment was determined using the Levenberg-Marquardt variation of the backpropagation algorithm and evolutionary methods. Training examples for near optimal behaviour of the game agent controller were generated using an artificially designed test bed. Evaluate the best controllers of both backpropagation and evolution learning, the resulting game agent is tested through several simulation processes, with multiple copies of the game agents used in each simulation run. Evaluate the best controllers of both backpropagation and evolution learning, the resulting game agent is tested through a number of simulation processes, with multiple copies of the game agents used in each simulation run [80].

The controller trained with backpropagation was less proficient although required less computational processing than evolutionary methods. From the results it was concluded that backpropagation trained game agent controller proved incompetent with the complex nature of the learning tasks. From the results, it was concluded that the backpropagation-trained game agent controller was unable to handle the complexity of the learning tasks [80].

Artificial datasets are also used to train MLP network. The μ-SIC system was developed in which the MLP network served as the basis for the behaviour of game agents in interactions with each other. MLP network feature vector consists of a set of values of personality, mood, and relationship of the game agent to determine the type of interaction that occurs. However, the authors have suggested that a trained MLP network is suitable for real-time use, but has proven to be time-consuming and therefore not suitable for online learning mechanisms [81].

The research conducted by Geisler et al. [82] used a player modeling approach for the generation of game agent behaviour in the game called first-person shooter. Human player game traces used to train controllers using a single MLP network and an ensemble of MLP. MLP Network feature vector consists of opponent agents and goals. Ensemble of four MPL networks performance as best when trained with boosting techniques. In addition, the results showed that a controller trained on traces of a poorly performing player resulted in a malfunctioning game agent.

The techniques of Bagging and boosting, in conjunction with a player modeling approach, have been used an MLP network was investigated used for controlling a game agent within the racing game Motocross. The controller network generated is compared to the controller's trained evolutionary methods. The human game play traces of different tracks were used as training set for backpropagation algorithm. The Generalization performance of the controller with bagging was not very good compared to a controller trained with boosting or evolutionary methods. When bagging was performed, the training was also inefficient regarding to computer resources used and processing time.

A related study also conducted by Togelius et al. [84] an MLP network that was trained with backpropagation on several player-generated examples originally used to get a player model for a simulated racing car controller. It has been reported that the resulting controller has worked extremely poorly. The work indicated that this was not because to the concept of supervised learning, but to the training which did not contain enough examples of incorrect behaviour and hence the trained controller could not recover from errors.

Both players created training examples and automatically generated training examples were used in the studies of Bryant et al. [38] used both human player game playing data and automatically generated training data. Accordingly, the authors have shown that the improved performance can be automatically attributed to an increased number of training examples. The data generated, the improvement of the behavioural consistency, was mainly due to the Inclusion of examples with symmetric transformations. The authors have pointed out that an improvement in performance has detected with an increase in automatically generated training data. The presence of symmetric transform instances causes behavioural consistency.

#### 2.4.4.2. Evolutionary approaches to learning in digital games

Ponsen et al.[85] presents EA based dynamic scripting technique to generate domain knowledge used by game agents in the real-time strategy game Wargus. This knowledge included in a rule base tactics to represent a set of scripted behaviour for game agents in each evolved chromosome population. A Papulation of chromosome were evolved via tournaments selection and customized genetic operators.

As the developers prepared the evolution offline against the game agents, this leads to successful generation of robust strategies for game agents. This can improve not only the game AI domain knowledge but also the performance of game agent [86]. To evolve a rule-base for game agents, Gallagher et al. [87] used Population-Based Incremental Learning (PBIL) [88], as an alternative to EA to obtain the parameters of a set of probabilistic rules that are used to deliver a game agent control strategy for the Pac-man arcade game.

In the work done by Parker et al. [89] reactive game agent used a canonical GA for the development of a rule base consists of consequences and priorities. Though many behaviour were evolving, the author also claims that with larger rule-based and chromosome length, complex behaviour can also be evolved. Using the same study tasks, Parker continued to research the development of decision and action parameters for a hand-coded, rules-based game agent controller [90].

Demasi et al. [91] conducted experiments by using a co-evolutionary approach to evolve rule-based opponent agents. The experiments were tested on a multi-agent action game. This is the single player game competing against a number of opponent agents. This is the single player game that is played against a number of enemy agents. Co-evolution is done to adapt the opponent's control strategies. The results showed that co-evolution successfully evolved the behaviour of the opposing agents in real time [91]. For the imitation and the prediction of the player's behavior, the regression algorithms proved to be suitable. In general, regression algorithms learn exact numeric ratings from the player experience, but an ordinal scale should be followed instead of the interval scale [92]. However, the known pitfalls provide adequate proof against the use of regression for modeling the player experience.

Classification is best suited for player experience modeling tasks if discrete experience annotations are selected from the list of possibilities and are provided as output targets [93, 94]. By classifying ratings, the transformation process induces some bias against data that seems detrimental and misleading to player experience [92, 95]. Preference learning as an alternative to regression and classification has been commonly applied to the modeling player's aspects. Numerous neural network architectures to predict the emotional and cognitive states of players have explored by Mart'ınez et al.[96] and Yannakakis et al. [97] Similarly, Garbarino et al. proposed in racing games [98]. While reinforcement learning has been pursued with relatively abstract and straightforward games [99]. The primary motivation for using RL for player modeling is being able to capture the relative judgment of a game states that humans are encoded internally while playing a game [100].

Flocking strategies are used to model agents complex behaviour when they interact with each other [101]. Finite state machines, behaviour trees are the most popular ad-hoc behaviour authoring methods for game development. These techniques are simple to implement, visualize and debug. However, these are extremely complex to design and are computationally expensive. For non-player characters in game AI processes of decision making and control have been a major focus of these methods until the mid-2000s [64]. To overcome the drawbacks of these techniques, Fuzzy rules or probabilities used to represent transitions.

The authors have implemented a modular based Behavior Tree (BT) system to imitate the human behavior [102]. They developed an expert-knowledge system using a finite set of behaviors instead of states. Furthermore, the modularity enabled BT to deal with more complex behavior design has been proposed by [103]. BT offers flexibility in design, testing and debugging that makes them dominant in game development. However, BT still have some disadvantages when compared to FSM. Especially, their dynamicity is rather weak since they are representations of static knowledge.

Industrial AI developers in the games have pointed out that the absence of modularity in game behavior has not supported high quality game AI development [104, 105]. A modular and extensible utility-based AI method was used to design control and decision-making in games. For example, helping an agent to decide which option to consider at a given time [106]. Being the core of AI, tree search algorithms are also used effectively in games. Although uniform search algorithms rarely appear successful in games, but surprisingly well in general video game playing such as iterative width search [107].

Variants of A* algorithm have been commonly used in games [108]. However, for two-player adversarial games have been presented for playing games such as Checkers and Chess-playing program [109]. Later on, implementation of Monte Carlo Tree Search (MCTS) in 2007 not only overcome the tree size limitation of Minimax, but also to improve the computational cost [110].

It has been shown that evolutionary planning far surpasses the Monte Carlo tree search in games. Later multiple variants of this technique applied by Wang et al. [111], and Justesen et al. [112] to StarCraft tactics. Other metrics such as player's jumping frequency, playing style, shooting and running are also recorded during game playing. These recorded data were then used to develop player experience models to optimize game levels [113]. Also beyond gameplay, information about player such as head pose

[114] or facial expressions [115] were gathered. After learning these utility values through evolutionary search, it is also possible to produce a clone of a human player. However, it seems that these "clones" have less ability for generalization than the personas defined by the designers [67, 116].

These approaches exhibit some limitations; unable to generalized unseen circumstances [3], dependence on fitness function either during the evolution of human-like agents or evaluation of the human-like AI-agent.

### 2.4.5 Learning from Demonstration (LfD)

The ability of agents to rely on artificial intelligence to behave like humans raises the many future technological trends [1]. It is worth mentioning that hand-coded agents and AI-based trained agents to play a game using machine learning algorithms deems to behave non-human like manner unnaturally or mechanically. The similar susceptibility is reported by [58, 114]. So far, success is far from guaranteed when attempting to create human-like agents, despite even well-playing ones. Characterized the behavior of a software agent to map its observations into actions from demonstrations over a period [59] to do a task with respect to the surrounding entities [60]. To carry out such behavior require much technical knowledge, domain expertise and many of human testers to test the quality of the game.

Hand-code development of such guidelines is often very difficult, and, therefore, machine learning techniques have been used in the development of policies. Also, most of the commercial computer games today have a large search space where AI has to make decisions in real-time, so traditional search-based techniques are not applicable. For this reason, game developers should strive to program specific strategies to play reasonably for every new game.Humans usually do only the task and believe that the observer can know how to imitate the behavior successfully [117].

Manual design of AI behavior, unable to effectively handle increasing complexity and more difficult to create the corresponding data structures and algorithms. To overcome the drawbacks of these techniques, a well-known alternative is to use some variants of machine learning to create AI-controlled agents and adapt their behavior.

A sub-domain of machine learning, LfD has been used to learn such behavior. This helps machines learn complex behaviors that are difficult to program. In games, learning from demonstration has been extensively used to model humanoid players. For the most part, LfD accepts expert behavioral patterns as an entry without explicitly describing the goal to achieve.

Numerous attempts have been made to imitate human behaviour in-car racing games to drive a car in human like fashion using LfD. To replicate human driving [118] used 2D car racing game as a testbed for research to train neural network controllers [9]. An earlier attempt has been made by [11, 12] for the learning part of driving behaviour of NPC cars using human traces. The focus of this research is for how well AI played and not driving behaviour learning. Some form of direct imitation learning has implemented by soft's Forza Motorsport, to estimate the performance of players on each track, instead of real player by using an ad-hoc approach [13]. It is of the case that, reproducing a robust, human behaviour in intelligent agents, mainly two factors can be attributed: (1) Poor Generalization, and (2) unstable in the demonstration.

Another important work done under LfD has recently surfaced in the Case-Based reasoning (CBR) community. [119] presented an imitation learning method that learns how to play RoboSoccer using other team's player traces. The case-based planning methods in context of LfD applied to real-time strategy games by Ontañón and associates [120]. It is worth mentioning that they also highlight the potential economic gain of a case-based reasoning approach, since the behaviour can be demonstrated rather than programmed. A Poker-playing agent developed by [121] using LfD. Conditional entropy based techniques was implemented to improve cases retrieval in LFO-based CBRs by [122].

The major difference between earlier work and CBR-based work is that, due to the lazy machine learning techniques of CBR, there is no need for any form of generalization during learning. Therefore, CBR learning is simply a matter of memorizing new cases and delaying any form of generalization until the problem is solved.

Additionally, the motivation of using Inverse Reinforcement Learning (IRL) for player modeling is to focus on rebuilding the reward function based on optimal behavior. It enables to capture the relative judgment of a game states that humans are encoded internally while playing a game [100]. Player modeling via RL [123]closely related to LfD can further be implemented during the process of game design or as a believable, human-like opponents [124]. RL approach models the player behavior [116] with the help of a Q table. This way of player behavioral modeling had only implemented in first person shooter games [100, 125], education game[126] and adventure games [124, 127]via inverse RL. The IRL paradigm has recently received increased attention for its LfD applications. One of the major drawbacks of using IRL is that there may be different reward functions that match the observed behavior, and need to designed the heuristics for learning bias toward the reward function of interest [100, 128].

Nevertheless, we can conclude that while the LfD has done a considerable amount of work in the last 20 years, it does not have a minimum of formalization or even a terminology agreement. At the end, advanced AI techniques with high potential must be used in games. On the other hand, due to lack of background knowledge and state-of-the art solutions to critical issues, still there exists a gap. To bridge the gap, academics should use and evaluate new algorithms that improve performance or create new phenomena or experiences. After a thorough study of literature, obviously, it seems that some AI game area have space for future implementations of AI methods.

### 2.4.6 Case-based Reasoning in Games

Case-based reasoning technique has been practiced for imitating humans, robots, and software agents. In this section, we will discuss with a special focus on the imitation of humans in games. CBR with combination of reinforcement learning was used to check the action of units on the battlefield that are simulated in real-time military strategy games[129].

This work uses two case bases. To decide an activity to be performed and the new status resulting from the execution of this action, the case referred to as the transitional case base. To determine the expected value for entry into this state, the case is referred to as value-base. Both case bases are initially empty and grow during the game. A new case is constituting if the selected case was not identical enough, is empty, or returns an improper state. However, if the case is very similar to the input, reinforcement learning is used to update the value of the attribute. A longer search time is never required because the case base is set to empty at the beginning of each search. This eliminates the need for case-based pre-processing, and also causes loss of what has been learned.

To learn a player's behavior from a software agent in the domain of the Tetris game, the combination of reinforcement learning and case-based reasoning was also used [130]. To manage the size of the case base, metrics are used to recognize cases that are not often used so they can be deleted. To control the movement of a team of players in a first-person shooter game, case-based reasoning with Qlearning, a method of reinforcement learning was used to map specific locations [131].

In real-time strategy games, case-based reasoning has also been used to retrieve game plans [132]. Cases are generated by recording a human's game play, then expertly commenting on the log file to group each action into a structured plan. Cases are created by using these plans that consists of the game state, a goal, and the plan to achieve that goal given this state of play. Therefore, there is a hierarchical connection between the cases, which represents the complete game plan of the human expert. Floyd et al describes in detail an approach used case-based reasoning to building an autonomous, spatially aware agent in the field of RoboCup football [133]. They use demonstration learning by observing the game decisions of other RoboCup football players.

**Table 2.1**

### 2.4.6.1. Case-based Reasoning in RoboCup

The RoboCup is a collection of competitions used as common platform and aimed at promoting research in artificial intelligence and robotics and the subject of numerous conferences and workshops each year. Case-based reasoning has been explored in the RoboCup domain [119, 133] to enable a software agent to imitate another agent and has been applied to mimic RoboCup Footballer in the RoboCup Simulation League.

Floyd et al describes in detail an approach used case-based reasoning to building an autonomous, spatially aware agent in the field of RoboCup football [133]. They use demonstration learning by observing the game decisions of other RoboCup football players.

**Table 2.1: Summary of Literature using Case-based Reasoning for the Development of Imitation Agents**

| **Case-Based Imitation Agents** | | | |
|---|---|---|---|
| **Reference** | **Methods** | **Goal** | **Environment** |
| Meteb.M. et al. [134] | Case-based reasoning (CBR) | Using CBR approaches to identify the most applicable solutions to behaviour management issues in humanoid soccer robots. | Humanoid Soccer Robots |
| Diana Sofía et al. [135] | Case-based approach | To assess the level of skill of the players in the game of Tetris. Cases are taken from previous tracks of the game which provide time series that explain the progression of certain parameters during the game. | Tetris Game |
| Thiago P. D. Homem et al. [136] | Qualitative Spatial Reasoning theory to model, retrieve and reuse cases by means of spatial relations | This offers a simpler, faster way than using a more conventional metric model to extract a case with better results. | Humanoid Robot Soccer |
| Cynthia L.Johnson et al. [137] | Use a context-based multi-agent system known as Collaborative Context-based | Training a team of autonomous agents in a simulation to mimic the actions of other agents. | Pursuit-Evasion Game bucket Brigade |
| Tesca Fitzgerald et al. [138] | Case based reasoning | Interpretation, in which the robot interprets new skill | Robotic Agents |

| | | | |
|---|---|---|---|
| | Framework | demonstrations as being related to previous observations, and (ii) imitation, in which a robot seeks to use previously learned skills to address new problem scenarios. | |
| Jonathan Rubin et al. [139] | Using a top-down approach to build case-based solutions. | A coherent framework for the development of strong case-based solutions based on expert judgment observation and generalization. | Poker |
| Glen Robertson [140] | Learning by observation approach was used in combination with a domain-independent case-based justification (CBR) paradigm. | The purpose of this work is to show a method for more effectively making better agents in real-time strategy games. | StarCraft |
| Michael W. Floyd et al. [119] | Machine learning methods applied to the case-base in order to understand the significance of various artifacts and to improve overall imitative precision. | Describing case-based imitation technique as an automatic mechanism for observation and imitation of other agents This enables certain simple agents to be imitated with limited human interaction. | RoboCup soccer-playing agent |

Experimental results revealed that simple reactive agents imitated with great success, but more complex agents cannot currently be imitated [133, 141]. When viewed qualitatively, these simple agents appear to behave similarly to the original agents. Various tests for al tested agents also examined the variation of the case representation [141, 142] and feature selection [119, 143] using both manual and automated way to improve the imitative performance using all features. Moreover, during a game these improvements make the distinction difficult of the imitation agent.

Although many cases can be generated automatically, the cases used during execution are randomly selected from the available cases. To optimize the case base time constraints, so far no attempts have been made. As such, no effort has been made to choose the best case for use or reduce redundancy. In addition, while feature selection has been used, there is no attempt to remove features that are not used to reduce storage and computing costs. To solve the problem, the CBR must contain enough variety of cases that can be remembered. And if the number of cases is large enough, this can be problematic for a real robot when resources are scarce.

CBR reasoning ability is associated to the number of cases that can be remembered. The system must contain enough cases so that various input problems can be resolved. As found in [170] the number of stored cases can be quite big (tens of thousands), which can be problematic if such approach is implemented in a real-life robot where resources are scarce.

### 2.4.7 Unsupervised Learning

In contrast to supervised learning, the model derived from observations is without specified target output in unsupervised learning. Previously, unsupervised learning was primarily applied to two main player modeling tasks: clustering behaviour and mining associations among the attributes of the player.

To analyses player behaviour in games, clustering offers a way to reduce the dimensionality of data. This results in some essential features that represent player behaviour regarding player navigation patterns; assets bought, items used, game genres played. A key challenge is to make clusters interpretable that is meaningful involve stakeholders such as designers, testers, and managers. Such an implementation of clustering has been demonstrated in the popular Tomb Raider game [144].

To solve the player modeling problems, such as identification of player type and player behaviour pattern detection, frequent pattern mining also played an import role. These techniques include Apriori, SPADE and GSP algorithm. As Martınez et al. [145] used GSP for the selection of inputs to the player model which are most frequent. Mining frequent patterns that often occur, for example, can be used to determine which game content brought together often or what is the next action after dying in one level [145, 146].

Unsupervised techniques, i.e., clustering has been used for behavioral classification depending on their attributes and association mining for finding frequent patterns of actions to determine player behavior in the game. On the contrary, Reinforcement learning (RL) for player modeling focus on rebuilding reward function on optimal behavior. It enables to capture the relative judgment of game states that humans are encoded internally while playing a game [100]. Player modeling via RL can further be implemented during the process of game design or as a believable, human-like opponent [124]. RL approach models the player behavior [116] with the help of a Q table. This way of player behavioral modeling had been implemented in first person shooter games [100], education games[126] and adventure games [124, 127]via inverse reinforcement learning (IRL). A drawback of using IRL is the different reward functions usage to match observed behavior which needs to design heuristics for learning bias toward the reward function of interest[100, 128, 147, 148].

### 2.4.8 Hybrid Algorithms

Another important class of algorithms that are tested in games are hybrid algorithms. AI approaches can be interwoven in a variety of ways to create modern, complex algorithms that incorporate the strengths of their combined components, often with a decisive effect. In this section, we call the resulting combinations of AI methods as hybrid algorithms. Table 2.2 summarizes literature using hybrid methods for the development of imitation agents.

**Table 2.2: Summary of Literature using Hybrid Learning**

| Reference | Approach | Goal | Advantages | Conclusions |
|---|---|---|---|---|
| Noor Shaker et al. [149] | Active Learning random forest | Random forest method to learn models of player experience | Accelerate the learning process and largely reduce the data needed for training. | Active learning is an efficient approach to effectively learn from data that is less likely that is presented in the literature |
| Liberatore et al. [101] | Genetic algorithms and flocking strategies | To control the Ghost Team in the game Ms. Pac-man | Reduce Complexity | It is tested how well GA-based GHOST controllers work compared to non-evolutionary approaches. FS has best developed with twice the value of fitness. These results support the claim that FSs are a viable solution for intelligent controllers. |
| Emmett Tomai et al. [150] | Learning Behaviour Trees | Game Designer | A wide range of player behaviour can be handle with a single hand designed behaviour tree. | Given a deterministic behavioural tree made by a designer that expresses a typical behaviour, the proposed algorithm is able to observe human players and automatically |

| | | | | |
|---|---|---|---|---|
| | | | | adjust the behavioural tree to explain their choices. |
| Zhao Richard, et al. [151] | Cyclic scheduling model | Game Designer | More reliable behaviour and increase the designer's efficiency, which lowers production costs. | A multi-tiered behaviour architecture that uses cyclic scheduling allows game designers to inform virtual character behaviour more reliably and more efficiently than manual scripts. |
| Shaker Noor et al. [4] | Neuro evolutionary preference learning and automatic feature | Estimating affective and cognitive states | Extremely accurate models of commitment, frustration and challenge reported. | The exhaustive method of search customization suggested behavioural data from the player's gameplay and visual behaviour. For example, evolutionary methods can be used. |
| Gagne, David J et al. [152] | Rule-based intelligent agent | EC may be used to learn organized techniques in the multi-agent distributed method. | EC Successfully implemented | To create ghost teams that are more successful than some manually coded rules, EC can be used to learn rule sets for FRIGHT agents. |
| Dai, Jia-Yue et al. | Evolutionary neural | Evolutionary neural network for | The agent learns well and plays better in team | A ghost can be developed based on a neural |

| | | | | |
|---|---|---|---|---|
| [153] | network | red ghost's chasing behavior | work than the traditional script-driven ghost. | network efficiently and learn well through evolution itself. |
| Slawomir Bojarski et al. [154] | REALM | Rule-based evolutionary computation agent learns to play Mario | The time needed for REALM to measure an action is comparatively limited. This means that we have space to boost the performance of the simulator. | REALM does not attempt to calculate exact locations, also does not look at the enemy future position while achieving competitive levels of performance. |

#### 2.4.8.1. TD Learning with ANN Function Approximators

As mentioned in section (unsupervised learning), RL represents the knowledge in a look-up table that becomes the reason for the depletion of computational resources as the size increases exponentially regarding the action state space. To handle the larger state-of-action representation space, an implementation of ANN based Approximators instead of Q value can be used. There are two algorithms, the TD Gammon algorithm and the inner Q network, that used this solution for learning in backgammon and Atari 2600 arcade games [155].

TD-Gammon used combination of multilayer MLP network and TD($l$) [156]. The feature vector for MLP consist of board current state as an input and probability to win as an output. Instead of states of winning game all other states are rewarded as zero. The MLP was then iteratively trained by playing the game against itself and using the sequence of positions to train the weights by back propagating the error of the temporal difference in its output. Actions were selected on the estimated probability of winning. In this way by adding expert knowledge, TD Gammon Agent has achieved human-level playing performance by integrating game-specific features into the input area.

#### 2.4.8.2. Deep Q Network

An RL based agent called *deep Q network* (DQN) was developed by a team Google's DeepMind [155] to outperform human-level playing performance in Arcade Games (Atari 2600). The feature vector of the DQN consists of game states as input and optimal action values of the corresponding state action as output. DQN is trained to approach the Q-score (actual score of the game) by receiving rewards directly from the game environment. DQN uses an e-greedy policy specifically for its action selection strategy. It is expected that there will be more sophisticated implementations of the deep reinforcement learning.

## 2.5. Case-based Imitation Agent through Case Knowledge Extraction

Knowledge transferring from an expert to a software agent is often a tough application that requires a substantial development time and programming effort [143] . This requires modeling expertise in a way that can be interpreted by a software agent. Even though creating a software agent, it is probably only possible to implement a set of specific tasks in a particular field. Therefore, it requires additional or modified expertise to perform new tasks. Learning through demonstration could be a natural alternative to model expert knowledge for people without such technical skills. Specifically, the burden of training will be shifted from expert to the agent to imitate the expert behaviour when faced with the similar task. This happens with case-based reasoning (CBR), where cases consist of expert performed actions as inputs. Case-based reasoning (CBR) is a reasoning methodology that solves current problems using prior problem instances and their solutions.

Case-based reasoning utilizes the assumption that similar problems have similar solutions, thereby it is well suited to imitate the behaviour of an expert (a human or an agent) when given similar inputs. There may be situations where the cases found the need to be adapted to meet new problems, but this issue is not taken into account in this research because the adoption process depends on domains and problems. Typically, CBR systems need far less knowledge than rule-based systems because they are based on experience. Therefore, it is not necessary to extract a domain model from these cases. In CBR, the ability to systematically relate to the past experiences also called reasoning is based on the size of the case base so that various input problems can be solved.

In case-based reasoning, to determine the most similar case often involves the use of the nearest- neighbour search [1]. It is known that searches of nearer neighbours have a high computational load since the inputs of the agent, must be compared with all available neighbours [157], the stored cases. As the number of neighbours increases, the time to search for the next neighbour also increased. Therefore, imitative agents with an increasing number of cases, it is plausible that there is an increase in the imitative ability of agents. In general, it can be stated that large CBR systems tend towards better solutions than smaller ones. As far as maximizing the imitative efficiency, a larger case-base is necessary, yet the size of the case is limited by cases that can be searched within the real time limit.

However, the diversity of the case base does not only increase by using a sizeable case base. If a new case is the same for an existing case, the agent's ability to imitate is unlikely to improve when added it to the case base. In this scenario, it will be essential to ensure that there are no very similar cases in the case base. To search more diverse case base in a real time limit, one solution is to pre-process cases in the case base.

### 2.5.1 Case Base Maintenance

To give a a boost to the cycle of CBR, Case Base Maintenance (CBM) is very important [158]. As case-based reasoning systems are applied in real-world situations, the problem of case maintenance becomes more important. With the advent of case-based reasoning systems in real-world, the case maintenance problem becomes increasingly critical.

In the past few decades, great attempts have been made to automate this process with various algorithms. The uncontrolled growth of case-bases can lead to serious performance problems because a decrease in retrieval efficiency and cases that are wrong or inconsistent are increasingly difficult to detect. The accomplishment of a Case-Based Reasoning (CBR) system relies heavily on three performance criteria and characteristics that affect the cases in the case base. There are three main criteria for the performance of CBR systems: problem-solving, quality, efficiency, and completeness. The quality of problem-solving for a CBR system refers to the average quality of the proposed solution that can be determined by the accuracy and adaptation effort required.

Accuracy is the percentage of problems that can be solved correctly. The adaptation effort required is the cost of modifying the proposed solution from the case taken to solve the problem. The CBR system must be determined that the solution presented can be used to solve new issues as a necessary qualification. The effectiveness of case retrieval will be affected by the existence of redundant cases, that is, an incorrect case may be taken even if the correct case is in the case base. Therefore, accuracy will be reduced, and the required adaptation effort will increase. Second, the efficiency of problem-solving is substantial, which can be defined as the average time to solve a problem. As the storage of cases increases, the speed of the case retrieval is slowed down. Concerning a real-time domain, there are limited cases that can be contained in a case base. This is an issue when more cases are available than those that can be processed in real time limits. To meet real-time obligations, a subset of cases must be selected for use as a case base. However, the selected cases should contain as much information as possible.

This area of research is known as maintenance of the case base and aims to refine the case base used by a CBR system [159]. There are three groups of maintenance operations: level of knowledge, representation level and level of implementation. In this study, we are concerned with maintenance processes at the knowledge level that directly alters the data contained in the case base [158].

### 2.5.2 Knowledge Level Maintenance

To solve the problem of a CBR system, the knowledge contained in a case base is very pivotal. Therefore, there is a trade-off between the retrieval performance and the number of cases. Initial work in maintenance level knowledge checks the policy for removing cases from the basic case [160]. Coverage and reachability were examined for each case in the case base. Case coverage is a collection of target cases that can be used to solve. Reachability is a collection of target cases in a case base that can be used to provide solutions. In a case base, not all cases are equal. Based on their contribution to competence, they are classified into four classes, i.e., a pivotal, auxiliary, support or spanning case. These are also true for the performance criteria. Pivotal cases are outliers and cannot be reached by cases other than themselves because they are too isolated to be resolved by another case. An auxiliary case can be reached by at least one other case having the same coverage that, make it redundant and hence not affecting the competence. Spanning cases, links regions of pivotal cases and do not affect competency.

### 2.5.3 Feature Reduction

In case-based reasoning, the most similar cases are considered as the solution to an input problem after comparison to cases stored. Therefore, for every problem provided, the case-based reasoning system must perform many comparisons between the problem and cases in the case base. These comparisons, whether of similarity or dissimilarity measures, can give rise to computation time required by the CBR system. The computational time required can be reduced by removing non-informative features in a case; called feature-reduction. As a by-product, the time needed for CBR to solve a problem will also be reduced because each comparison of cases will be carried out in a shorter time and it can also improve the performance of CBR system. The reduction in dimensionality was performed by feature selection methods in related work on FR [161].

To reduce the dimensionality of the data for a case-based reasoning system, a widely used technique k-nearest neighbour search algorithm or a variant of the nearest-neighbour search was used[162]. Because the neighboring k-algorithm has been known to be sensitive to redundant features [157] especially the case-based imitation system [119, 141, 143], this problem is critical to consider during the pre-processing process.

### 2.5.4 Case Selection

In case-based reasoning, the current situation is equated compared to the situation faced earlier to find the right solution for the current situation. To find the appropriate solution in case-based reasoning, the current problem is compared with the earlier experienced problem. The performance of the CBR system can relate to the number of problems that have previously occurred in the case base. This allows CBR to locate similar cases and the resultant solution to the current problem. However, in real-time environments, the case database must be searched within a real-time limit, which limits the number of cases in the case database. Several cases can be replaced with a single case, in which case it is replaced to maximize the case-based information. This would produce a reduced case base size but would ideally carry similar information as previously obtained. This procedure is called prototyping. Prototyping will allow the information in the case base to remain the same, but it will reduce the time needed to find a case.

#### 2.5.4.1. Similar Cases

During the collection of training data, there is a possibility that some or much of the collected data is very similar. There may be subtle differences in these similar data items that result in objects associated to different classes. However, these items often contain almost the same information, and it is less advantageous to keep all items than keep one item and discarding the rest.

## 2.6. Neuroscience and AI

During recent times, prompt growth has been carried out in the allied arenas of neuroscience and artificial intelligence. A long and entwined history exists between neuroscience and artificial intelligence (AI) fields. Recently, the related fields of neuroscience and artificial intelligence have achieved rapid growth. At the beginning of the computer era, work on AI was inseparable from neuroscience and psychology. Many of the early explorer were straddling these two areas, as collaborations between these disciplines 'being proved very productive. However, interactions have become less common lately because the complexity of the two topics has increased significantly and the boundaries of scientific disciplines are getting stronger. The growing importance of neuroscience is critical to generating and accelerating ideas in AI research [31]. To gain insight into innumerable important facets of general intelligence on a higher level, the study of animal cognition and its neural implementation also plays an important role.

The development of AI has two advantages in scrutinizing biological intelligence. Firstly, for applying new algorithms and architectures, neuroscience is a rich source of stimulation as biological computation was considered critical to support a cognitive function. This will be an excellent test field for integration into artificial systems. Second, neuroscience can be used as a test environment to validate existing AI techniques. As an example, if an algorithm does not somewhat reach the required or expected level of performance, but we find that this is central to the functioning of the brain, we can assume that increased engineering efforts aimed at artificial systems are likely to pay-off. Indubitably, from a practical viewpoint when setting up an AI system, we do not have to indiscriminately force compliance with biological believability. What interests us is a system for understanding the neuroscience level of the brain, to be precise the algorithm, the architecture, and the functions. While concentrating on computing and algorithmic levels, we get insight that can be transferred to the normal mechanisms of brain work, while leaving *silico* in place of intelligent machines to accommodate unique opportunities and challenges.

It is expected that artificial intelligence will significantly impact clinical practice in the foreseeable future. Even though the use of artificial intelligence in medicine is exaggerated, it cannot be denied that artificial intelligence will have a major impact on the medical field soon. Several different types of brain activity measurements and signal recordings such as positron emission tomography (PET), electroencephalography (EEG), Magnetoencephalography (MEG), and functional near infrared (FNIR) have been studied. But because of the hemodynamic response, electrical signals are fast compared to blood flow. However, good spatial resolution of fMRI distinguishes it from other available methods. It is good for inferring cognitive brain state from fMRI data.

The focus of current advances in human neuroimaging technology has been documented only to detecting the content of a person's conscious experience based on non-invasive measures. .Functional Magnetic Resonance Image (fMRI) can detect brain activity to collect data on human brain activity [22, 163]. Real-time monitoring of the functioning of the human brain is possible due to Neuro-dynamic functional brain imaging techniques.

The human brain can process various kind of impulses that activate multiple organs in the body. The functional connections of the brain regions enabled the brain to accomplish these tasks [12, 164]. The fMRI is considered to be a powerful experimental technique for analyzing stimulus-based activation in the brain. The triggering of systematically cross-linked activated brain regions has established itself as a new trend in fMRI imaging through the analysis of multi-voxel patterns. The key observation of these studies is that brain regions are activated for specific mental activities.

In order to find brain regions, their specific cognitive processes and stimulus-triggered activities, several experiments were designed using functional brain imaging [16-18]. Since the image (fMRI) contains thousands of voxels, the analysis of such a data volume leads to an over-fitting of the classification algorithm [20], high level of noise, the curse of dimensionality and small number of samples[21, 165] in the field of radiology. Furthermore, redundant features enlarge the search space and may significantly affect the performance of many learning algorithms [166]. Due to the lack of tools, previous attempts to investigate this collaborative mode of functioning of the human brain have been hampered. Recent advances in instrumentation and data analysis techniques have enabled a complete brain examination while performing specific tasks [13]. The high-resolution fMRI can access specific areas of the brain that are activated when a task is performed.

To accurately assess neural activation, the most popular techniques of Artificial intelligence technologies i.e. Machine learning, convolution neural networks, was applied to decipher brain patterns for medical image analysis. For a more accurate classification, a state-of-the-art ensemble/hybrid technique was used [22, 23].

The neural network (NN) is a type of machine learning has been applied to a wide range of fMRI problems and to develop diagnostic models. To identify the complex interactions between input data, ANN distinguishes the mechanisms of time series and results. This makes it possible to recognize the hidden relationships that are normally invisible when using traditional statistical approaches. So ANN seems to be a promising tool for clinical decision making and have been used in several areas, such as Alzheimer's disease [24, 167] , noise reduction [168], cognitive state classification [17], fMRI pattern classification[25, 169], face recognition [26] word prediction from fMRI data[27] just to mention a few .

The idea of profound knowledge emerges through the study [163]. Formally, activations of neurons are needed to fine tune the weights to a regular neural network (NN) so that NNs meet expectations. However, the NN training process at different stages can lead to complex computational problem. Backpropagation has been an effective gradient-descent algorithm since 1980, playing an important role in NNs [170]. Controlled learning method has implemented to train the ANNs. The performance of backpropagation is compromised, when used to test statistics. Although the training accuracy is high because the base is on the local gradient data, the algorithm is often stuck in local optima with very low fitness. The same is also reported in the literature when applied to certain problems [171].

In addition, If the training size is not large enough, NN will have difficulty dealing with the outliers [172]. Therefore, there is a greater need to find the relevant Artificial Intelligence algorithms that are used in the field of cognitive neuroscience. Also the hybrid algorithm domain needed to be explored. The ensemble/hybrid techniques have recently gained more attention in computational neuroscience due to their exceptional advantages in processing with complex data structures [22, 28].

Therefore, hybrid techniques have been proposed as tools to improve the classification performance. They aim to combine the results of different classification members trained for the same task, which is one of the most popular fields in pattern recognition. This integration improves overall accuracy compared to any single classifier [29]. One of the most significant used hybrid intelligent system is the neuro-fuzzy combination. These systems use fuzzy logic to model the environment, having non-linear vagueness on data [30], interconnected neural network processing elements and information connections [17].

We hope that, for the future, closer alliance between neuroscience and AI researchers, as well as the identification of a common language between the two, will enable a righteous cycle in which research will be accelerated by shared theoretical insights and joint empirical progress [11]. For algorithmic construction, refining intelligence by comparison with the human brain could provide intuition to understand the secrets of the mind, such as dreams, the essence of creativity, and even the consciousness of one day.

## 2.7. Challenges and Requirements

For modeling and analysis of human behavior, data collections to create corpora are a prerequisite. Moreover, it is very difficult to perform experiments with efficient and reproducible results. Following are the major issues with respect to data:

- Need of a framework for annotating complete and easily interoperable data and analysis
- Right now, the biggest obstacle is the lack of standard, sufficient and accessible information publicly to researchers.

The annotation for the modeling, analysis, and synthesis of social behaviour remains a labor-intensive and time-consuming task. In fact, data is enriching usually manually with descriptive and semantic information. Recent advances in sensor technologies have enable to automatically collect behaviour data with features of interest. This leads to build collection of rich data sets and can be encoding manual greater than before. Furthermore, the modeling and analysis of behaviour data hampered by the lack of complete annotation system or scheme of classification for speech, gestures and other features.

Technologies to treat human behaviour should try to accommodate very specific aspects of the contexts where they are used Instead of trying to be generic. In contrast, it should always be kept in mind that effective approach to a given situation or context may not work in others.

It is found that Model-based approaches contain reasoning systematically in support of an idea and understanding of the selection of models that do not have a model-free approach. But not succeed to include relevant features due to absence of comprehension of model builders. On contrary model–free approach has benefit of automatically identify the related features. An extensive range of functions of the behaviour can be extracted and measured.

Therefore, model-free approaches seem best appropriate along domain-specific knowledge, feature selection and extraction needed to achieve meaning models.

- To develop approaches that produce better results with as less data as possible. We therefore need a comparative study of different algorithms and techniques to know which technique can produce results with a comparatively smaller dataset so that model can be learning in smaller time.
- We need generalized techniques i.e. one technique should be able to extract data from different kinds of platforms.
- Behavioural data analysis still suffers from a scarcity of knowledge. Moreover, different algorithms have different strengths and weaknesses we need to find out which algorithms work best on what kind of dimension.

At the end of industrial game AI, sophisticated AI techniques with high potential must be used in their games. On the other hand, academic game AI has less ability to deal with realistic problems that occur during game production to bridge the gap, academics should use and evaluate new algorithms that improve performance or create new phenomena or experiences. AI developers evaluate software architectures and reliable algorithmic modifications that support a specific game design. After thorough investigation of literature, obviously, the margin for future applications of AI methods appears to be quite large under certain AI game areas.

## 2.8. Chapter Summary

The panoramic view of game AI with respect to most popular AI methods and core AI areas identified that research has focused primarily on Ad-hoc authoring, tree search, evolutionary computation, supervised learning, reinforcement learning, and unsupervised learning and performs to be dominant in both player experience and behavior modeling[10, 68, 69].

Supervised learning is very useful in the game AI, and performs to be dominant to develop autonomous and assisted model players, Mostly, the approaches used to develop AI agents are generally traditional , i.e., rule-based pre-programmed methods used in counter-strike bots by [61] and the finite state machine is implemented in Halo 2 AI game by [62, 63, 70], A* and FSM and planning algorithms for playing games such as Checkers and Chess-playing [173]. These techniques are simple to implement, visualize and debug. However, these are extremely complex to design and are computationally expensive [64].

Neural network and evolutionary computation are the leading method for modeling players, content generation, playing to win and believable agent design research. Moreover, behavior authoring is only useful for playing games. For modeling player's neural networks seem to be appropriate, but they need a comprehensive set of training and validation data. Because traditional neural networks were limited to offline learning, an alternative ensemble and modular networks proved to be a promising approach to real-time use. It's pretty obvious through our literature review that neural networks and evolutionary methods, when implemented in a hybrid way, become predominant.

This type of hybridization for the generation of game agents has been successful because of the optimization method of evolutionary techniques. However, with the use of evolutionary and hybrid approaches, some success has been achieved. The stochastic nature of digital games is likely to be a cause of fitness assessment, therefore, requires larger game data. Approaches that were pre-initialized populations have proven to be a robust and competent game-agent controller. This creates a basis for the operation of online learning techniques.

Mostly neural networks are used by developers to train the agent to follow the same actions that the human player took under the same situation. The designed agent trained on the human game playing traces as input. The selection of input parameters is based on the environment of player character and output being the action of the player at the same state of the game. A major problem is the generalization of unseen situations that lead to an action that is very different from what the player would have taken in the same situation. The main reason is this training aimed at repeating the same decisions of the given player in the same state of play, and not playing well in terms of game goals [81, 83, 84].

Although evolutionary techniques can provide robust optimization methods, canonical implementations are at the forefront usually suffers from a number of optimization problems, i.e. on slow convergence rates, on right selection of a right chromosome scheme and on a fitness function for learning [42].

However, success has also been achieved with the use of reinforcement learning, there are also several problems with the use of RL techniques in complex gaming environments. An appropriate state action abstraction associated with a value function representation is needed [174]. However, to improve the performance of game controllers with RL, the inclusion of hierarchical approaches and prior knowledge can help in this regard [175]. In the same way, if pre-trained game controllers are used, the resulting agent may improve [67, 176].

In conclusion, it has been found that there is still a considerable room for the implementation of innovative machine learning techniques in the domain of game AI agent development. Some limitations of machine learning techniques are also highlighted during learning tasks, especially in more complex game environments. Furthermore, a deeper understanding of the nature of the digital game environment and problems associated with the control of the learning game agent have materialized. Nonetheless, the machine learning application has caused a number of common problems with the digital games, from the selected learning algorithm or game genre used.

We can consider such limitations as opportunities that can bring industry and academia closer to continuing to develop the AI in the game. The developers from industry and researchers from academia concluded that there is a gap between the two. Such as lack of background knowledge and state-of-the-art solutions to critical issues. Due to the slow development on both ends, this gap still exits. To bridge the gap academia should use and evaluate new algorithms that improve performance or create new phenomena

or experiences. After a thorough study of literature, obviously, some AI games appear to offer room for future AI techniques to be implemented.

# Chapter 3: Imitation of Human-like Behaviour through Hybrid Approaches

## 3.1. Overview

The prime focus of the work presented in this chapter is on providing a framework for the development of AI-agents, which can accurately exhibit the individual player behaviour that is considered realistic in a game environment. To achieve this goal, this study developed AI-agents that are trained on winning and losing game patterns of individual player recording one's game traces using machine learning techniques (Objective 1). To achieve the reproducibility in the results, an automated process of the corpus has also been developed. The model of these two patterns also concludes what moves let the player lose the game, and that move takes him to victory.

The novelty of this work is to develop the framework to derive a measure of imitation of human behaviour from the likelihood ratios using precise AI techniques. i.e., variants of artificial neural networks and Neuro-Fuzzy hybrid system. This study focused on two key benefits of neural networks and fuzzy systems, which excel at handling vague and a priori knowledge. There are a number of neural architectures grouped under the ANNs umbrella and compatible hybrid learning algorithms. However, certain architectures may function better for certain learning problems than others. Therefore, it is a real challenge to find the best architecture and training method.

For the development of such an agent, we use human play traces which imitate demonstrated behaviour. There are typically two types of neural networks mentioned in the literature, namely feed-forward and recurring networks. The later ones are superior to former in computing. Although the number of ANN architectural options is overwhelming, choices for representative neural architectures from these two groups need to be narrowed down. This study utilized four well known network architectures of ANNs including feed forward, recurrent, extreme learning machines, and regressions are used to simulate the human player behavior.

Furthermore, we used Adaptive Neuro-Fuzzy Inference System (ANFIS) with grid partitioning, subtractive clustering and fuzzy c-means clustering for training to find the ideal ANFIS. ANFIS is a soft computation system powered by data that reflects a neural network approach to solving problems of functional approximation that are used to solve complex and non-linear problems. An integrated statistical approach to historical evidence, the ANFIS process, is used to evaluate the parameters of the expert knowledge-based fuzzy system to take account of the self-adaptability and learning capability of the systems with the aid of fuzzy-if-then rules. The use of this ANFIS hybrid framework tends to complement the weakness of the systems. The Takagi-Sugeno (TS) fuzzy inference model type was chosen because a global function approximation can be achieved from a set of local linear equations.

The hybrid learning algorithm, when used as a learning algorithm in an adaptive network has a better ability to accelerate convergence and avoid the occurrence of trapped in local minima. This is because the hybrid-learning algorithm, when used as a learning algorithm in an adaptive network, has advantages over the original method of Backpropagation. After studying the literature thoroughly, it has found that it is still necessary to explore the application of selected architectures from ANN and ANFIS systems for the development of human-like agents. This study focused on two key benefits of neural networks and fuzzy systems, which excel at handling vague and a priori knowledge.

To evaluate the believability of the obtained AI-gents, we also compare the traces generated by agents with the actual human player by using two automatic statistical methods Mann-Whitney U- test and Cosine similarity in this study. Both methods validate the human-likeness of our game agent controllers worked perfectly with high average similarity scores 95%. All simulations are implemented in MATLAB and Python.

The proposed architecture represents AI behaviour by demonstrating it to the system. In case, the agent behaves incorrectly in each situation; the game developers do not merely have to keep the bug in the program and fix it, they can demonstrate the right action in each situation. Then this information is integrated into the training set so the system can behave better in the future. It is worth pointing out that, AI-agent can also be shaped into any desired behaviour using players' data. According to our best of knowledge, no study has been published that implements and compares the behavioural imitation techniques used by this research. This motivates the work presented in this research, which we will explore in depth on these grounds.

## 3.2. Baseline Techniques

To build a computational model, two learning paradigms were explored. Namely artificial neural networks (ANNs) and adaptive neuro-fuzzy inference system (ANFIS).

### 3.2.1 Supervised Learning

As ANNs have many standard architectural variations, eight different standard architectures of ANNs have been selected for study to simulate the player behaviour. Feed-forward Multi Perceptron (FF)[177], Extreme learning machines(ELM)[178], Focused time delay neural network(FFTD) [179], Design time series distributed delay neural networks(DTDNN)[179-181] for feed-foreword networks category, layer recurrent network(LRN), Nonlinear autoregressive Model (NARX) for recurrent networks[182], Generalized regression neural networks (GRNN) [183] for regression networks category. The purpose is to find the best fit of the model after comparison for application of imitating human behaviour.

### 3.2.2 Hybrid Learning (Adaptive Neuro-Fuzzy Inference System (ANFIS)

A typical hybrid intelligent system, such as the Adaptive neuro-fuzzy inference system (ANFIS) in neuro-fuzzy combination, was developed initially by Jang [184] and has since been used with regularity. The neural network used to tune the parameters of a Fuzzy inference system consists of a set of fuzzy rules [185]. These rules must be equal to membership functions and cannot share the same output as membership functions. Because ANFIS architecture is based on a first-order Takagi-Sugeno model, the fuzzy rules are as follows. Figure 3.1 illustrates the five-layer inference and defuzzification mechanism of first order Sugeno model.

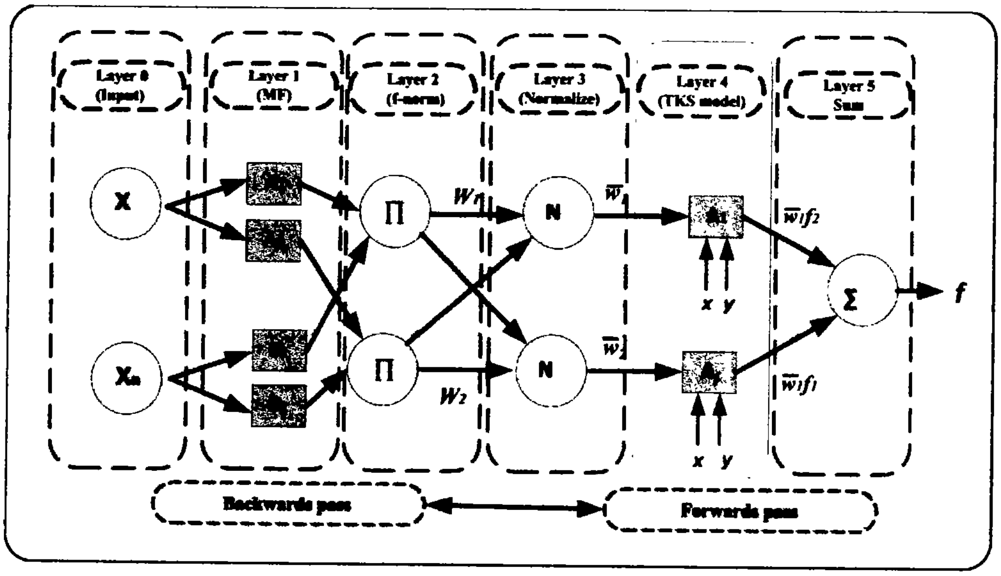



**Figure 3.1: ANFIS five-layer architecture**

$\mathbf{R_k}$: **IF** ($x_1$ *is* $A_{j1}$) *AND*........ *Moreover,* ($x_1$ *is* $A_{jp}$) *THEN* (3.1)

$(f_g = a_{j0} + a_{j1}x_1 + \ldots + a_{jn}x_n)$;

where:

- $x_i$ is the input variable
- R is the number of rules, i.e., $R_k$ is the $k^{th}$ fuzzy rule

$fg$ is the outputs; $a_{ji}$ are the design parameters, $P$= [0, n]

## 3.3. Proposed Framework

This section sheds a light on the functionality and architecture of the proposed scheme as shown in Figure 3.2 in detail. The methodology comprises of the following major tasks: (1) Generation of data for the model (2) Builds the model (3) apply the model to the environments.

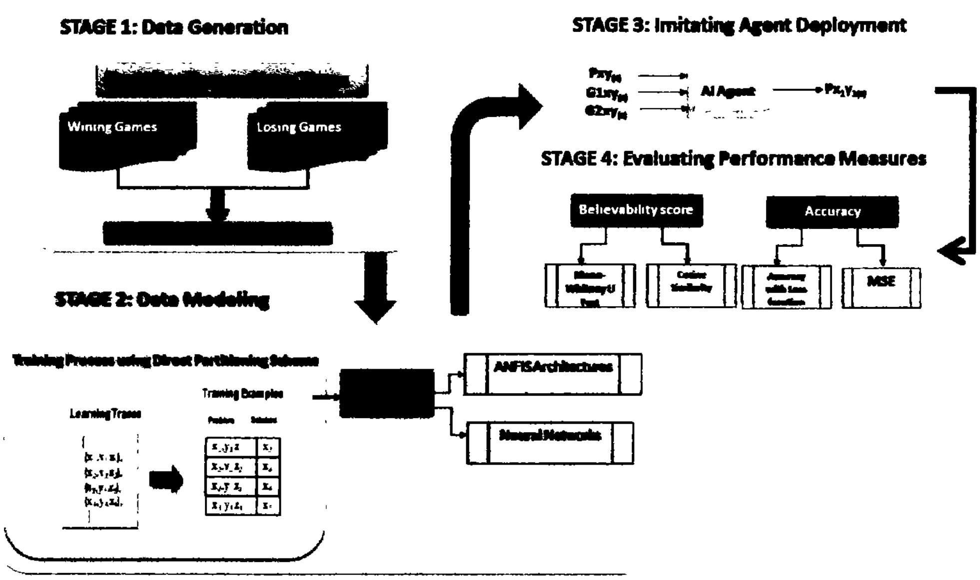



**Figure 3.2: Proposed Model for Imitation Human Player.**

Algorithm for Imitation of human Player is given below,

**ALGORITHM: Proposed framework for Imitation Human Player**

| | |
|---|---|
| **Input:** | A Collection of current positions of three agents, Pac-man, Ghost1 and Ghost2 |
| **Output:** | Next executed position as outputs($Px_c + 1, Py_c + 1$) |
| **Step:1:** | Generate and Collect traces of game agents |
| **Step 2:** | Generate Abstract Traces as:<br>$R = [\{P(x_0,y_0),\ldots, P(x_t,y_t)\}, \{G1(x_0,y_0),\ldots, G1(x_t,y_t)\}, \{G2(x_0,y_0),\ldots, G2(x_t,y_t)\}]$<br>R is a sequence of traces, where<br>$(x0, y0)$ is the initial position of agents and<br>$(xt; yt)$ is the position of agents at a given time t<br>Each agent moves in the following quadrants. Initial Pacman Position: $P(x_0, y_0)$<br>Where $i$ =1,2, 3,... ... ... ... ... ..... 9<br>$j$ =1,2, 3,... ... ... ... ... ... ... ...18 |
| **Step 3:** | Direct Partitioning scheme for traces of agent's movements, with 50:50, 60:40, 70:30, 80-20 training-testing split<br>$(Pt;At,Gt) = P(x_c, y_c; x_c + 1, y_c + 1), (G1(x_c,y_c), G2(x_c,y_c))$<br>Where Pt and Gt are sequence of traces of Pacman and Ghost Agents at a given time t and the actions At the Pac- |

man very next executed position at time interpreted as output, and each (Pt ; At ,Gt) $\subset$ R.

**Step 4:** Generating inference system by tuning training parameters with ANFIS hybrid learning system until the specified error target is achieved as:

*Rk* : IF x is $A_i$ AND y is $B_i$, THEN $f(i)=p_i x + q_i x + r_i$

Where:

- x and are y are the input variables.
- $A_i$ and $B_i$ are Fuzzy Sets.
- $f(i)$ are the outputs.
- $p_i$, $q_i$, *and* $r_i$ are the design parameters, P=[0, n]

**Step 5:** Evaluating Performance measures

a. Calculate Minimum MSE error based on step 3

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 \qquad (1)$$

**Step 6:** Estimate Prediction of Pacman agent position ($Px_i$, $Py_i$) using STEP 5

**Step 7:** Estimate Loss Function as:
until the specified error target is achieved as:

$$L = \left|P(x_i, y_i) - P(x_i, y_j)\right| \qquad (2)$$

IF L=0 THEN
Predicted action is the same with lineal combination of the same (x, y) coordinates and direction.
IF L=1 THEN
Difference of one-step in the same direction.
IF L=2 THEN
Difference of two steps in the same direction and so on.

**Step 8:** Execute the model

### 3.3.1 Data Generation

To measure the performance of the selected machine learning approaches studied in this work requires traces of human players from game sessions to imitate human play. A module for retrieving player's data for generating a corpus has been developed while playing games. This was achieved by modifying the Pac-man project so that event logs of player and ghosts are created every time there is a game being played. Python 2.7 has been used for game development and is not based on any external packages. For configuration and development, accessor methods have been modified to access the state data of all three agents of the Pac-man game environment.

Following are the application domains used for this research. Case study Pac-man used for generating human game play traces and deployment of AI-agents and Case study Battles Ground (PUBG) was used for validating the predictive results of proposed framework.

#### 3.3.1.1. Application Domains

Real-time human behavioural data is needed to depict particular person's behaviour/style of play. The testbed simulation game based environment used for the study as a benchmark is the version of Pac-man projects and Battle Grounds (PUBG)

**Pac-man Simulation**

Some interest from AI researchers has recently been paid to the Pac-man game. With the Pac-man game, some artificial intelligence research was conducted. To depict a particular person's behaviour/ style of play, we need real-time human behaviour data. The test bed simulation game-based environment used for the study as the benchmark is a version of Pac-man projects. These projects are developed for UC Berkeley's AI introductory course CS188. The original source code is available on the web (http://ai.berkeley.edu/project_overview.html). We have developed a module to automatically retrieve the player's data by modifying the Pac-man project. The game is coded in Python 2.7 and does not depend on other external packages. Pac-man has to gather all the pills from a maze while avoiding two ghosts chasing each ghost guided by a distinguished strategy and moving at its own speed. Pac-man should avoid ghosts for losing his life if touched by ghosts. Special power pills scared ghosts for a limited period. Pac-man can eat them and during times of vulnerable ghosts.

A ghost, eaten by Pac-man disappears from the maze and reappears after a certain time

in a non-scared state. To win the game Pac-man takes all the pills before the ghost. Pac-man can move left, right, down and up with arrow keys. The game has only one level. The primary aim of the Pac-man agent is to eat as many as pills before reaching at the end of the game as quickly as possible. Human player's data gathered from the same single level of the game to evaluate the controller performance. The selected environment is deterministic in nature. Figure 3.3 shows the general layout of the Pac-man game interface. The Pac-man game layout has a simple design with one difficulty level. To model the player's behaviour and to evaluate the performance, this representation used to apply artificial neural networks and neuro-fuzzy logic and case-based reasoning method.

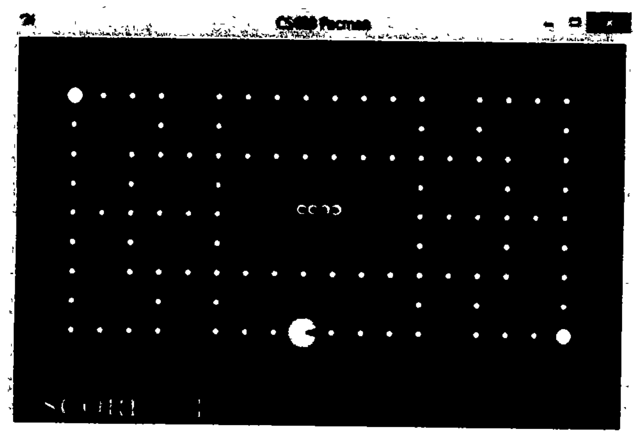


**Figure 3.3: Capture image from Pac-man game.**

### Environment Representation

Figure 3.4 illustrates the maze, design for the Pac-man project based on the idea of the board spilt into tiles to understand the Pac-man ghost behaviour. "Tile" refers to 8 x 8 pixels square on the screen. This gives us a total size of 18 x 9 tiles. The grey shades tiles are not accessible to Pac-man or the ghosts as illustrated in Figure 3.4. The impact of the tiles reflected as; it is considered that a ghost captured Pac-man when he occupies the same tile as his. Pac-man agent navigates his maze world, to reach a particular place and efficiently gather food. Four primitive actions (↑, ↓, ←, →) available for Pac-man virtual player.

The search algorithms for formulating and controlling the behaviour of the antagonists (Ghosts) in the game is implemented by DFS, BFS, UCS, and A*. To model the behaviour of the player correctly, we needed traces of the winning as well as the losing game data. The pattern of these traces leads us to conclude what made the losing lose the game moves and what moves that lead to victory. Thus, the Environment representation is composite by a total of three elements, Pac-man, Ghost Agents and their tile positions used as input for approaches used.

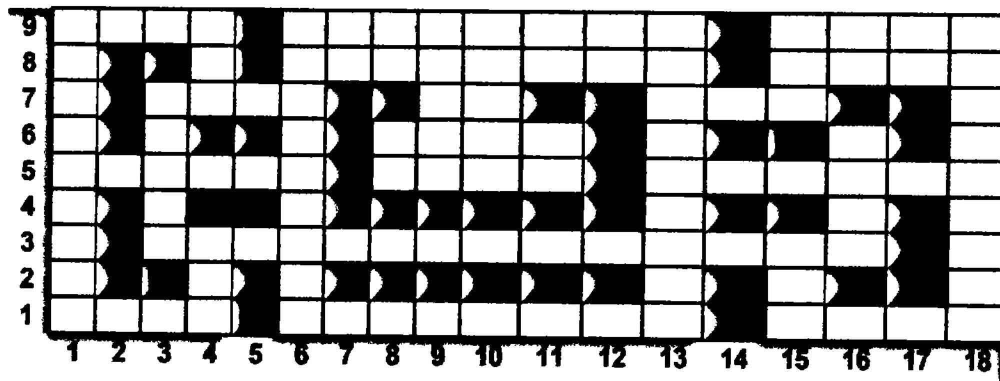



**Figure 3.4: Equivalent Layout in (X, Y) Coordinates of Hurdles and Valid Movement as a grid of 18 * 9 tiles**

**Battle Grounds (PUBG)**

Player Unknown's Battle Grounds (PUBGs) is a Battle Royale competitive multiplayer game released by PUBG Company. The PUBG game was motivated by the Japanese film "Battle Royale," published in 2000. Throughout the video, up to 100 players jump into the island and attempt to destroy others while escaping killed.

The accessible safe space in the playing region of the game reduces over time which causes the remaining players to compete in closer locations. The final player or team who eliminates each other will be able to win the round, for which experience, and bonus points are awarded, who provides points towards the number of killers in each round. The accessible safe space in the playing region of the game reduces over time which causes the remaining players to compete in closer locations. For PUBG, any match begins with up to 100 players (matchId) Teams will be put in players (groupId) at the conclusion of the game (winPlacePerc) depending on how many players already reside until removed. Throughout the game, players will obtain different weapons, live away-but-not-out teammates, drive vehicles, dive, jump, fire, and all of the outcomes – such as sinking too far or running over and killing themselves.

The dataset of PUBG game for research is taken form kaggle and available on the web https://www.kaggle.com/c/pubg-finish-placement-prediction/data. The accessible safe space in the playing region of the game reduces over time which causes the remaining players to compete in closer locations. The final player or team who eliminates each other will be able to win the round, for which experience, and bonus points are awarded, who provides points towards the number of killers in each round. The data derives from tournaments of all kinds: solo, duos, squads, and custom.

**Generate abstract traces**

By recording the game event logs, the trace recording module generates an initial, unprocessed case base. This log has execution traces, which has information related to each game object state, the time at which the object state change and the game result, lose or succeeded. Once the game is finished, an abstract trace is formulated from the original game traces as.

$R = [\{P(x_0,y_0),\ldots, P(xt,yt)\}\quad c, \{G1(x_0,y_0),\ldots, G1(xt,yt)\},\{G2(x_0,y_0),\ldots, G2(xt,yt)\}]$

R is a sequence of traces, where
$(x_0, y_0)$ is the initial position of agents and
(xt; yt) is the position of agents at a
given time t
Each agent moves in the following quadrants. Initial Pacman Position: $P(x_0,y_0)$
Where i =1,2,3,..........................9
j =1,2,3,..........................18

The abstracted trace contains critical pieces of information: those needed to determine the Pac-man player's next move. Key pieces of data consist of the positions of players and ghosts in the form (x, y) coordinate concerning time. An abstract trace 'R' is a sequence of entries, where each entry (Xt; Yt) = $P(x_c, y_c; x_{c1}, y_{c1}), G1(x_c, y_c), G2(x_c, y_c)$ contains the state Xt at a given time t, and the actions Yt the Pac-man very next executed position at time t interpreted as output. The game state Xt captures. Since the trace can help localize past events, execution traces provides a considerable advantage in modeling the behaviour patterns of individual human players as well as a population of human players.

### 3.3.2 Data Modeling

Once the data were generated, two methods with ANNs: Supervised learning and Hybrid Learning (Adaptive Neuro-Fuzzy Inference System) have been used and compared to simulate human behaviour as mentioned in Section 3.2. We want to build a system that learns from the x and y coordinates of all game characters. This learning leads the agent to predict the next move of the Pac-man agent imitating player playing patterns and initially to replace the player input through keyboard.

#### 3.3.2.1. Direct Partitioning Scheme

The key challenge for all machine learning techniques is how to achieve a generalized error, also called testing error, as low as possible. The factors that determine how well a machine-learning algorithm works are its ability to minimize training error and the gap between training and testing errors. Direct Partitioning scheme for traces of agent's movements, with 50:50, 60:40, 70:30, 80:20 training-testing split were used as:

$$(Pt;At,Gt) = P(x_c, y_c; x_{c+1}, y_{c+1}), (G1(x_c, y_c), G2(x_c, y_c))$$

Where Pt and Gt are sequence of traces of Pacman and Ghost Agents at a given time t and the actions at the Pac-man very next executed position at time interpreted as output, and each

$(Pt; At, Gt) \subset R$.

#### 3.3.2.2. Artificial neural network (ANN) Training and Learning in Environment

To find the best neural network model for the selected genre of video game, eight standard neural network architecture variants are used, including feed-forward, repetitive, extreme learning machines and regressions to simulate human gambler behaviour. All ANN variants are trained throughout the epochs until the minimum error for each division was reached. The error is calculated as the Mean Square Error (MSE) between the actual and desired output.

#### 3.3.2.3. Adaptive Neuro Fuzzy Inference System (ANFIS) Training and Learning in Environment

Generating inference system by tuning training parameters with Adaptive Neuro Fuzzy

Inference System (ANFIS) hybrid learning system until the specified error target is achieved as:

Rk : IF x is Ai AND y is Bi, THEN f(i)=pix + qix+ri

Where:

- x and are y are the input variables.
- Ai and Bi are Fuzzy Sets.
- f(i) are the outputs.
- pi, qi, and ri are the design parameters, P=[0, n]

To train ANFIS, the first step was to implement 'anfis.m' function. The 'anfis.m' function receives the dataset in matrix form, where each column represents the number of inputs desired, and the last column represents the output. The desired output in comparison to the evaluated system is performed with the function 'evalfs.m'. In addition, to create an initial FIS training data matrix, the functions 'Genfis1.m', 'Genfis2.m', 'Genfis3.m' were implemented. The Genfis1 model uses grid partitioning to produce single-output membership function on the data. Genfis1 initiates with fixed membership function numbers "3*3*3" for all trails. Every rule produced by Genfis1 has one output membership function, which is of the "linear" type by default. The Genfis2 structure used Fuzzy subtractive clustering (SC) to generate a FIS with a single output to model the behaviour. The system then continued to train with various training data sets and radius values to determine the best ANFIS system. A cluster's range of influence, specifically radii = [0.5- 1.0], was used. The best configuration results from the radius between 0.8 and 1.0. To model the data behaviour, the Genfis3 structure used fuzzy c-means clustering (FCM) to generate an FIS. It then used the same model to evaluate the lowest MSE values. The ANFIS training process begins with a varying number of training instances and the shape of their membership functions until the minimized error reached. In this work, Neural Network (NN) and Takagi-Sugeno (TS) fuzzy inference systems based ANFIS model was developed to adapt learning content formats for our game environment. Once the minimum error was reached, the model results have been checked before being deployed in the environment to obtain quality.

### 3.3.3 Imitated Agent Deployment

The agent developed trained on current positions of three agents, Pac-man, Ghost1 and

Ghost 2 as input gathered from human players and provide next executed positions as output. For the principal case study, this research use Python 2.7 as development tools and MATLAB for training and testing the architectures. A middle-tier configured to implement trained architectures to pass the next Pac-man move to Pac- man game. The "Pymatlab" package allows Python users to interact and communicate with MATLAB from Python. The basic purpose of this package is to route Python data to MATLAB function for execution. After processing, you retrieve data to python, which allows you to process the data with built-in MATLAB's functions, toolboxes, or MATLAB-scripts. For configuration and development and accessor methods has been modified to access the state data of all three agents of Pac-man game environment. Middle Tier Function received (x, y) coordinates of Pac-man agent from the trained network and pass it to Pac-man Game to run. A hurdle and wall control function has also developed for the Pac-man agent.

### 3.3.4 Evaluating Performance measures

To evaluate the prediction of proposed framework, following measures have been used in this study.

#### 3.3.4.1. Calculate Minimum MSE error

Among the various standard measures to represent an error, e.g., Mean Squared Error (MSE) and Accuracy were used in this work. The error is calculated as the mean square error (MSE) between the actual and desired output. To compare the model results and select the best one, MSE recorded against each run between the actual and the desired output.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 \qquad (3.2)$$

Then estimate prediction of Pac-man agent position ($Pxi$, $Py_j$) has performed.

#### 3.3.5.2 Estimate Accuracy with Loss Function

In this study, a precision metric with multiple loss functions was used to evaluate the prediction performance as:

Estimate Loss Function until the specified error target is achieved as:

$$L = \left| P(x_i, y_i) - \hat{P}(x_i, y_j) \right| \qquad (3.3)$$

IF L=0 THEN

Predicted action is an exactly the same with lineal combination of the same (x, y) coordinates and direction.

IF L=1 THEN

Difference of one-step in the same direction.

IF L=2 THEN

Difference of two steps in the same direction and so on.

We present the average loss for all the entries of the traces in the test set. We also used two evaluation metrics to evaluate the human-like play of the agents: Mann Whitney-U Test and Cosine Similarity Metric.

### 3.3.5 Evaluation of Agents of Playing Style Identification

To evaluate the believability of the obtained AI agents, a statistical analysis of the accuracy was performed. Then, the traces left by the agent and player were compared to reproduce the observed behaviour. Two automated methods namely the Mann-Whitney U test and the cosine similarity were applied rather than human assessment.

#### 3.3.5.1. Mann-Whitney U Test

The Mann Whitney U test is a popular non-parametric test for comparing the results between the two independent groups. Conversely, the null and two-sided research hypotheses for non- parametric tests can be expressed as follows:

H0: The two populations are equal

H1: The two populations are not equal.

This test is often done as a two-tailed test and, therefore, the research hypothesis shows that the population is not the same as opposing direction determination. The test statistic for the Mann Whitney U Test is denoted U and is the smaller of U1 and U2, See Eq.

(3.4) & (3.5) below.

$$U1 = n1n2 + \frac{n1(n1+1)}{2} - R1 \qquad (3.4)$$

$$U2 = n1n2 + \frac{n2(n2+1)}{2} - R2 \qquad (3.5)$$

### 3.3.5.2 Cosine Similarity Metric

The general principle of evaluation is to calculate the vectors that characterize human behaviour and agents. These vectors represent as signatures because it is designed to represent behaviour. By measuring the distance between an agent's signature and a human, it is possible to decide whether his or her behaviour appears as human or not. Cosine similarity (CS) between two vectors $\overline{V}_h$ and $\overline{V}_c$ is defined as (See Eq. (3.6))

$$CS(\overline{V}_h, \overline{V}_c) = \frac{\overline{V}_h - \overline{V}_c}{|\overline{V}_h||\overline{V}_c|} \qquad (3.6)$$

Where $\overline{V}_h$= (xi, yi) denote a set of actual Pac-man positions by an individual human player and $\overline{V}_c$= (xi, yi) denote a set of Pac-man positions by the AI controller. The function with a pair of input vectors$\overline{V}_h$, $\overline{V}_c$ indicates whether $\overline{V}_h$ and $\overline{V}_c$ match or not.

## 3.4. Experimental Design of Imitation of Human-like Behaviour

In these experiments, we designed different agents based on techniques and compare them to find the most precise and accurate agents that were developed specifically for the purpose of playing humanly. The aim of this study is how to derive a measure of imitation of human behaviour from the likelihood ratios between the models and which model gives best results.

### 3.4.1 Demonstrated Dataset

To measure the performance of the selected machine learning approaches, neural architectural types, and Adaptive Neuro-Fuzzy Inference System (ANFIS) studied in this work requires traces of human player from game sessions in order to imitate human play.

A module was developed to record the events log files of player's data. The module recorded the data with 100% accuracy and there is no missing value. Since the selected environment is deterministic in nature, the data generated are of the stationary type. As the data was captured by a developed module and the analysis is done later by proposed imitating learning framework. So, this study is offline. The log files generated stores the positions of players and ghosts in the form (x, y) coordinates with respect to time with labels as shown in Table 3.1. The AI-agents designed trained on six inputs ($Px_c$, $Py_c$, $G1x_1$, $G1y_1$, $G2x_2$, $G2y_2$) i.e. Pac-man's current position, along with two corresponding ghost positions of the grid representation and the very next move of Pac-man is interpreted as outputs ($Pxc_1$, $Pyc_2$) are used for the learning purpose.

**Table 3.1: Log Data Generated with labels of Pac-man and two Ghosts game agents**

| Pac-man Current Position | | Ghost 1 Position | | Ghost 2 Position | | Pac-man Next Position | |
|---|---|---|---|---|---|---|---|
| $Px_c$ | $Py_c$ | $G1x_1$ | $G1y_1$ | $G2x_2$ | $G2y_2$ | $Pxc_1$ | $Pyc_2$ |
| 9 | 1 | 8 | 5 | 11 | 5 | 10 | 1 |
| 10 | 1 | 9 | 5 | 10 | 5 | 11 | 1 |
| 11 | 1 | 9 | 6 | 9 | 5 | 12 | 1 |
| 12 | 1 | 10 | 6 | 8 | 5 | 13 | 1 |
| 13 | 1 | 10 | 5 | 9 | 5 | 13 | 1 |
| - | - | - | - | - | - | - | - |

Volunteer players have been selected from undergraduate students at University of Gujrat. To this end, data were collected with different levels of experience of the Pac-man game as shown Table 3.2. Keep in mind that different levels of experience and playing styles of players allow them to play the same level in different ways.

Therefore, imitation learning performance should differ from player to player. One could argue that data of ten players are very small sample. However, each player played 30 games, hence each imitation-based model trained on winning and loss game patterns separately and it is a non-trivial exercise.

**Table 3.2: Player's Traces of winning and losing game play**

| Players | Sample Size | |
|---|---|---|
| | Win | Loss |
| Player 1 | 2294 | 1125 |
| Player 2 | 2336 | 1235 |
| Player 3 | 2526 | 1224 |
| Player 4 | 2616 | 1178 |
| Player 5 | 2459 | 1309 |
| Player 6 | 2332 | 1142 |
| Player 7 | 1942 | 1296 |
| Player 8 | 1443 | 994 |
| Player 9 | 1972 | 3768 |
| Player 10 | 2366 | 1019 |
| **Total** | **22286** | **14290** |

To perform all the experiments, this research used two common datasets of the same kind reflecting winning and losing behaviour. The traces of losing player behavior is almost half the traces of winning players. The reason is that if a player has lost the game, they will score fewer points. To find the best model, a uniform sampling technique was employed for training and testing for both datasets to get the lowest generalized error and to reduce the configuration space. In order to train and validate the model 50:50, 60:40, 70:30, 80:20 training/testing spilt were used.

## 3.5. Results and Discussion

To illustrate the main objective of this study, we describe the results of agents trained to human game play traces; evaluate its performance and their human-likeness in playing games in this section. All the experiments are performed on two common datasets of the same type but reflecting winning and losing behaviour to imitate the human game play as explained in dataset section.

### 3.5.1 ANN Results

The first contribution of our experimental work is to find the best neural network model for the selected genre of the video game after comparing eight standard architectural variations of neural networks. For the performance evaluation, a sample of 600 tracks (in heuristics) were selected from the data sets of ten players. Experiment 1 proceeded with varying training/testing instances ratios and neuron value of winning and losing games play traces. The neuron value, usually between 1 and 15. The Figure 3.5 presents the best configuration of each neural nets option with least MSE value.

The results also revealed that from all the neural networks applied, the dominant neural net with considerably least error is generalized regression neural network (GRNN). The Generalized regression neural network behaves similarly for both types of game-play traces from all other selected architectural variations as shown in Figure 3.5. The results indicate that GRNN has an empirical advantage over all other selected architectural variations. This is because GRNN has many theoretical advantages over other ANNs. In contrast to standard feed forward networks, GRNN solves the convex optimization problem by always converging to a global solution and not being trapped by a local minimum, which depends on the minimum samples and has robust internal over-fitting mechanism[186].

It was found that generalized error mostly obtained between the epochs ranges of 1 to 3 for all training/testing ratios used for the experiment. This is particularly evident in the case of GRNN, since it forms very reasonable regression surfaces based on only a few samples. The structural parameters are not determined iteratively but directly from examples and can generalize immediately when structure learns. It is also worth noting that the error remains nearly constant for all training/testing ratios used for the experiment, as shown in Figure 3.6. While the training and generalization error generally varies with the size of the training set.

This regression and function approximation approximation-based architecture includes polynomials in its hypothesis, not just linear functions to increase the model's capacity. The capacity of the model refers to is its ability to adapt to various functions[187].

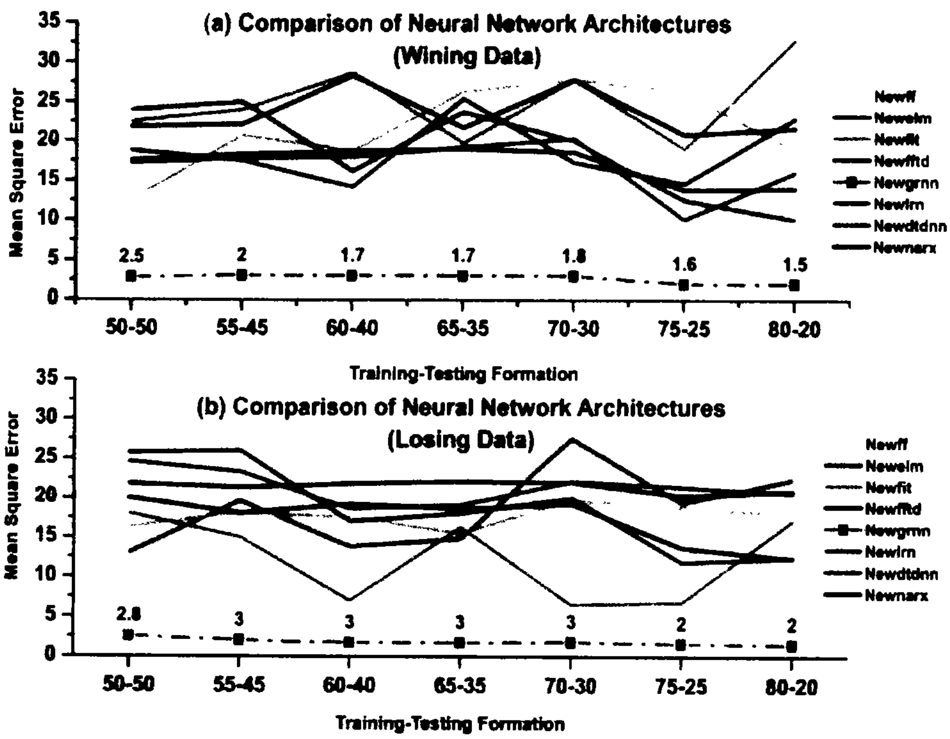



**Figure 3.5: Least error of Neural Network Architecture of Winning and Losing Game Play traces.**

grnn= Generalized regression ; ff= Feed Forward; lrn= Lay-ered recurrent ; dtdnn= Design time series distributed delay; fit= fitting neural network; narx= Nonlinear autoregressive network; fftd= Focused time delay neural network.

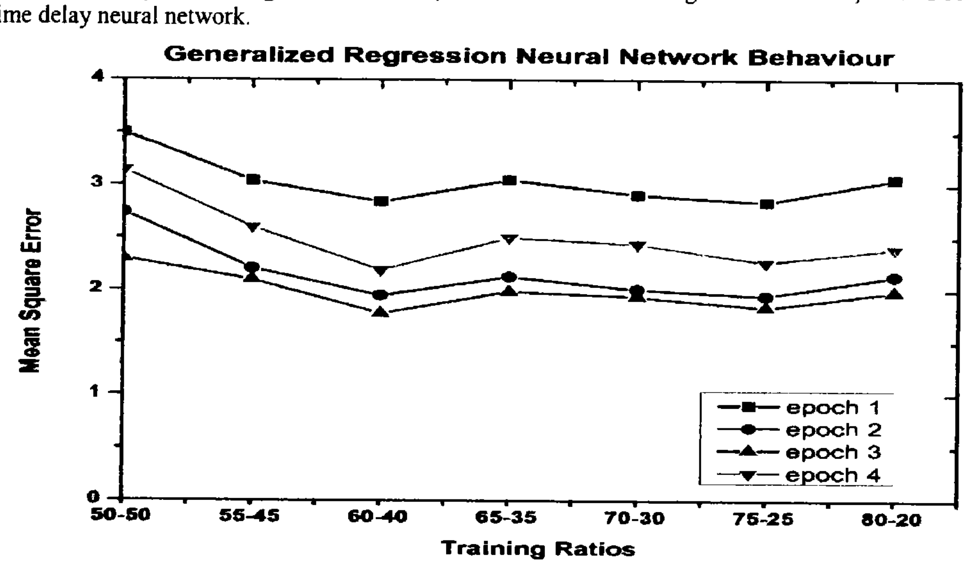



**Figure 3.6: Error trend of Generalized Regression Neural Network**

It is worth mentioning here that all neural networks behave arbitrarily when deployed into agents. The agents based on these ANNs were stuck in an unreasonable cycle and proved incompetent players.

Also, often choose inaccessible tiles, but jump to random locations where the AI-agent immediately loses the game. Hence, behave very contrary to the human-like play. On the other hand, the generalized regression neural network performs acceptably without entering a cyclical state. This allows you to explore the level without being stuck soon, when adding some fitness heuristics.

### 3.5.2 ANFIS Results

Furthermore, to find the ideal Adaptive Neuro-Fuzzy Inference System (ANFIS), Grid Partitioning, Subtractive Clustering, and Fuzzy C-means Clustering (FCM) for training were used. We implement Backpropagation and hybrid-learning algorithm to tune the Sugeno fuzzy inference systems using neuro-adaptive learning techniques. For this purpose, four discerning membership functions i.e., Triangular (Trimf), generalized bell (Gbellmf) and Gaussian (Gaussmf) were used to the input/output data.

However, triangular-shaped (trimf) membership function with a hybrid-learning algorithm performs most effectively with a minimum average error during training and testing data sets as shown in Figure 3.7 for both winning and losing data types. The hybrid learning algorithm, when used as a learning algorithm in an adaptive network has a better ability to accelerate convergence and avoid the occurrence of trapped in local minima. This is because the hybrid-learning algorithm, when used as a learning algorithm in an adaptive network, has advantages over the original method of Backpropagation. It was also found that Pac-man motion prediction did not vary largely between the other membership function (MFs) and was only marginally poorer than the 'trimf' values. Therefore, membership function type 'trimf' was selected for Grid Partitioning to generate a single output FIS from training data to compare with Subtractive Clustering, and Fuzzy C-means Clustering. Here, the validity results are calculated as the mean square error (MSE) between the actual and desired output. The results in Table 3.3 illustrate category wise MSE of players using three methods Grid partitioning, Subtractive Clustering and Fuzzy C-mean of winning and losing games data sets. The minimum MSE were achieved by Subtractive Clustering.

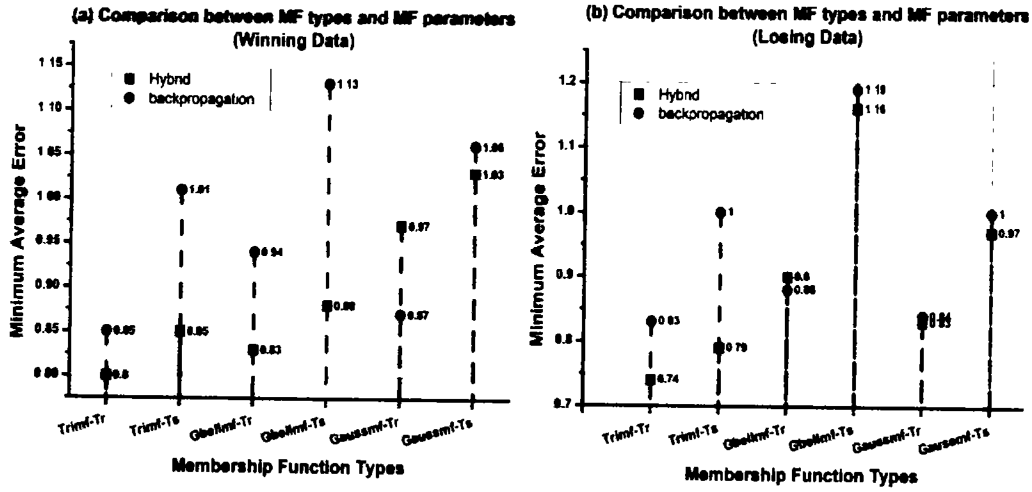



**Figure 3.7: Comparison between Membership Function types with MF Parameters for Winning and Losing Data**

**Table 3.3: Accuracy for winning and losing game play traces**

| | Accuracy | | | | Accuracy | | | | Average Accuracy | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Win | | | | Loss | | | | Win + Loss | | | |
| | L-0 | L-1 | L-2 | L-0+L-1 | L-0 | L-1 | L-2 | L-0+L-1 | L-0 | L-1 | L-2 | L-0+L-1 |
| **Genfis 1** | 0.50 | 0.30 | 0.06 | 0.80 | 0.47 | 0.28 | 0.09 | 0.75 | 0.45 | 0.28 | 0.075 | 78.0 |
| **Genfis 2** | 0.62 | 0.35 | 0.03 | 0.97 | 0.66 | 0.28 | 0.02 | 0.94 | 0.64 | 0.31 | 0.025 | 95.5 |
| **Genfis 3** | 0.58 | 0.33 | 0.10 | 0.91 | 0.57 | 0.29 | 0.06 | 0.86 | 0.56 | 0.31 | 0.08 | 88.5 |
| **GRNN** | 0.52 | 0.34 | 0.08 | 0.86 | 0.57 | 0.24 | 0.05 | 0.81 | 0.59 | 0.29 | 0.065 | 83.5 |

Genfis 1= Grid partitioning; Genfis2= Subtractive Clustering; Genfis3= Fuzzy c-means clustering; GRNN=Generalized regression neural network

Certain architectures may perform better than others for certain learning problems. So it is very important to find the answer to the question: "While imitating human behaviour, which training method produces the best prediction results? This work used accuracy with loss functions to measure the prediction ability of implemented models. The results in Table 4 illustrate category wise average accuracy of players with accuracy at level-0 (L-0), L-1, and L-2 using three methods (Grid partitioning; Subtractive clustering; fuzzy c-means clustering) and Generalized regression neural network of winning and losing games data sets.

Although some architecture specified accuracy to Level-4, it is usually very nominal. Therefore, it is not important to mention here. The average accuracy for both wins and loss player traces of the GRNN at L-0 is approximately 54%. Using Grid partitioning; Subtractive clustering; fuzzy c-means clustering, the average accuracy for both win and loss datasets at L-0 is approximately 48%, 64% and 58% respectively. All the models produced very poor results with four primitive actions available for virtual players.

At this, stage all the models unable to play in a minimally acceptable way when deployed into agents. However, when we combine the accuracy with Level-0 and Level-1 to be considered as single accuracy before applying it into the controller then it increased to 95%, which is significantly high. By examining the results in Table 3.3, the accuracy levels statistics exhibit a general trend for both player behaviour, indicates that the average accuracy value of Subtractive clustering at L-0 and L-1 is higher, i.e. Subtractive clustering performance is better overall and fuzzy c-means clustering follows closely. The results trend shows that the two unsupervised methods (fuzzy c-means clustering and subtractive clustering) when using to generate a FIS outperformed in terms of accuracy score by a margin as shown in Table 3.3.

Several researchers compared different AI methods to learn human behavior and used a crowd-sourced version of the Turing Test to measure the similarity of the style of play [4, 188, 189].Indirect methods, and i.e. Neuroevolution and dynamic scripting, while providing the best results in training, do not appear promising to external observers[4]. Miranda et al. used the backpropagation, Neuroevolution with a fitness function, which was expanded to include parameters at the game level. The accuracy obtained by all methods is approximately 60%. A fitness function was used to measure the similarity in play style. The agents were incapable of playing in a minimally acceptable way[189]. Miranda et al. [9] who tried to find features to distinguish between human and non-human, uncovered some interesting points about it. Edwards et al. [190] suggested mimicking the Imitations Latent Policies from Observation (ILPO) framework to train an agent to mimic behaviors from observations of expert states with very little environmental interactions. Khalifa et al. [191] created agents to play not only well but also in human-like manner for the GVG framework. Experiments show that the perceived performance of the modified MCTS agent is quantitatively similar to that of human gamers. The Turing test shows that the agent has a human-like style of play in some games but not in others.

### 3.5.3 Results Validation

The aim of this experiment is to show that the proposed ANFIS-based hybrid modeling technique is more general and can be applied to many other classification or prediction applications. It will be an interesting investigation that the current methodology will be appropriate. To perform this experiment, this research used two type of datasets. Relatively, large datasets are required for a fair comparison of the techniques. We aggregate the same data that was used to reflect winning and losing behavior to imitate individual human play in previous experiment. The dataset was collected of 30 games (15 winning games and 15 loss games) of twenty-six (26) players. The second is a benchmark dataset collected from the Kaggle machine learning repository.

This data consists of anonymized player data worth over 65,000 games, with 25 attributes. The data is used to predict the final placement based on the final stats in the game and the initial player ratings. Table 3.4 presents descriptive statistics for all of the datasets used. This experiment proceeded with two overall best-performed methods i.e. Subtractive Clustering (SC) and Fuzzy C-Means Clustering (FCM) to compare with other case study.

**Table 3.4: Aggregated Player's game play Data**

| | Win | Loss | Win and Loss | PubG |
|---|---|---|---|---|
| **Total Instances** | 62549 | 39472 | 102014 | 50000 |

The results in Table 3.5 illustrate category wise average accuracy at L-0, L-1, and L-2 on two type of datasets. These results achieved also confirmed that the proposed methods attain very high accuracy in different domain. The promising results of this study support the viewpoint that proposed methodologies can be used to gain insight into human data in any paradigm even though the current work involves a collection of data obtained from game-based test beds. The proposed models were also compared to mean square error (MSE) and root mean square error (RMSE) to evaluate the measure of performance, as shown in Table 3.6. Another experiment was performed.

Another experiment was performed results, by comparing them with other prominent machine learning techniques. This research implements Random Forest, an ensemble-based decision learning algorithm, and XgBoost from the Gradient Boosting Paradigm. The comparative accuracy values displayed in Table 3.7. From the presented results Genfis2 and Genfis3 dominate other implemented machine learning algorithms.

**Table 3.5: Comparison with other state of the art dataset**

| | Genfis2 | | | | Genfis3 | | | |
|---|---|---|---|---|---|---|---|---|
| **Data** | **L0** | **L1** | **L2** | **L0+L1+L2** | **L0** | **L1** | **L2** | **L0+L1+L2** |
| **Pacman-Win** | 0.30 | 0.49 | 0.17 | 0.98 | 0.34 | 0.47 | 0.15 | 0.97 |
| **Pacman-Lost** | 0.53 | 0.16 | 0.11 | 0.81 | 0.30 | 0.39 | 0.24 | 0.93 |
| **Pacman-Win&Lost** | 0.89 | 0.10 | 0.002 | 0.99 | 0.30 | 0.59 | 0.07 | 0.97 |
| **PubG** | 0.99 | 0.002 | 0 | 0.99 | 0.99 | 0.005 | 0 | 0.99 |

Genfis2= Subtractive clustering; Genfis3= fuzzy c-means clustering

**Table 3.6: Comparison with Proposed Hybrid Machine Learning Techniques**

| | Genfis2 | | Genfis3 | |
|---|---|---|---|---|
| **Datasets** | **MSE** | **RMSE** | **MSE** | **RMSE** |
| **Win** | 0.27 | 0.52 | 1.34 | 1.15 |
| **Lost** | 1.48 | 1.21 | 2.06 | 1.43 |

| | | | | |
|---|---|---|---|---|
| **WinLost** | 0.13 | 0.36 | 1.03 | 1.01 |
| **PubG** | 0.01 | 0.10 | 0.01 | 0.13 |

The comparative accuracy values displayed in Table 3.7. From the presented results Genfis2 and Genfis3 dominate other implemented machine learning algorithms. Therefore, our proposed hybrid methods are superior to the implemented machine learning algorithms.

**Table 3.7: Comparison with other AI/ Machine Learning Techniques**

| | **Accuracy** | | | |
|---|---|---|---|---|
| **Datasets** | **Genfis2** | **Genfis3** | **RF** | **Xgboost** |
| **Pacman-Win** | 0.98 | 0.97 | 81.36 | 94.37 |
| **Pacman-Lost** | 0.81 | 0.93 | 70.79 | 96.56 |
| **Pacman-WinLost** | 0.99 | 0.97 | 70.08 | 98.00 |
| **PubG** | 0.99 | 0.99 | 97.17 | 97.37 |

Genfis2= Subtractive clustering; Genfis3= fuzzy c-means clustering; RF=Random Forest; Xgboost= Extreme Gradient Boosting.

## 3.5.4 Evaluation of Agents of Playing Style Identification

An experiment 3 performed to evaluate the believability of the obtained AI agents, the traces left by the agent and player were compared. This study used automated methods rather than human assessment. Because they judge the game AI subjectively, are often inaccurate and are time-consuming because of the involvement of many people to assess agents. The size of human testers to get adequate results of the evaluation by human judgment is however a great challenge[58, 192]. The limits of human abilities are also a major downside in human assessment i.e., Turing test. To achieve this, statistical method Mann-Whitney U test and Cosine similarity methods were used. Human perception, however, is the ultimate judge of the resulting quality of game character believability[58]. Figure 3.8 illustrates the comparison of two best ANFIS mechanisms generated to imitate them with traces of two players.

### 3.5.4.1. Mann-Whitney U Test

This part of the study extrapolates the Mann-Whitney U Test to the believability of the obtained agents. To find the answer to the question: "Is there a significant difference between the traces of human players and the agents?" traces of human players and the agents were compared using the Mann Whitney U test, a non-parametric statistical technique. Mann Whitney U test was also applied by[4, 9], but to compare the ten players and all three AI-agents in both win and losing game races.

The average difference between players and AI-agent is not more than 0.01, which is minimal. It seems difficult to separate human players from AI-agents. The comparison results also reinforce the results of the Turing crowdsource test to find the significance between each observation made during the statistical Mann-Whiney U test for measuring similarity on each pair of agents. An examination of the findings in Table 3.8 reveals the results of the Mann -Whitney U test of all ten players for the win and losing traces of human player and three best agents (Grid partitioning; Subtractive clustering; fuzzy c-means clustering, Generalized regression neural network)) showed no statistical difference. All values of p obtained are >0.05 with a margin. The close Mean rank scores of the human players and agents indicate that the two populations are the same in a statistical sense as shown as shown in Figure 3.9. The results obtained indicate that the traces generated by agents are identical to the traces of human players.

### 3.5.4.2. Cosine Similarity Analysis

The initial purpose of this experiment is to measure the overall orientation between human game play traces and AI trained agents. Second, to verify the believability results obtained from Mann-Whitney U Test. A cosine measure is the best choice to judge the orientation rather than magnitude. The individual player game logs were used to measure the behavioural similarities between traces of human-run game characters and compared with the game traces of trained AI-agents on the same player data. Cosine similarity is used primarily in high-dimensional positive spaces, where the results are always within the range of [-1 to 1]. -1 means the opposite, 1 means the same, with 0 normally indicates independence. Table 3.9 illustrates the similarity of player style calculated by using a cosine similarity value for vectors of individual player actions and

AI trained agents. In general, we can see high similarity scores 99%, between the traces of all players and AI-agent.

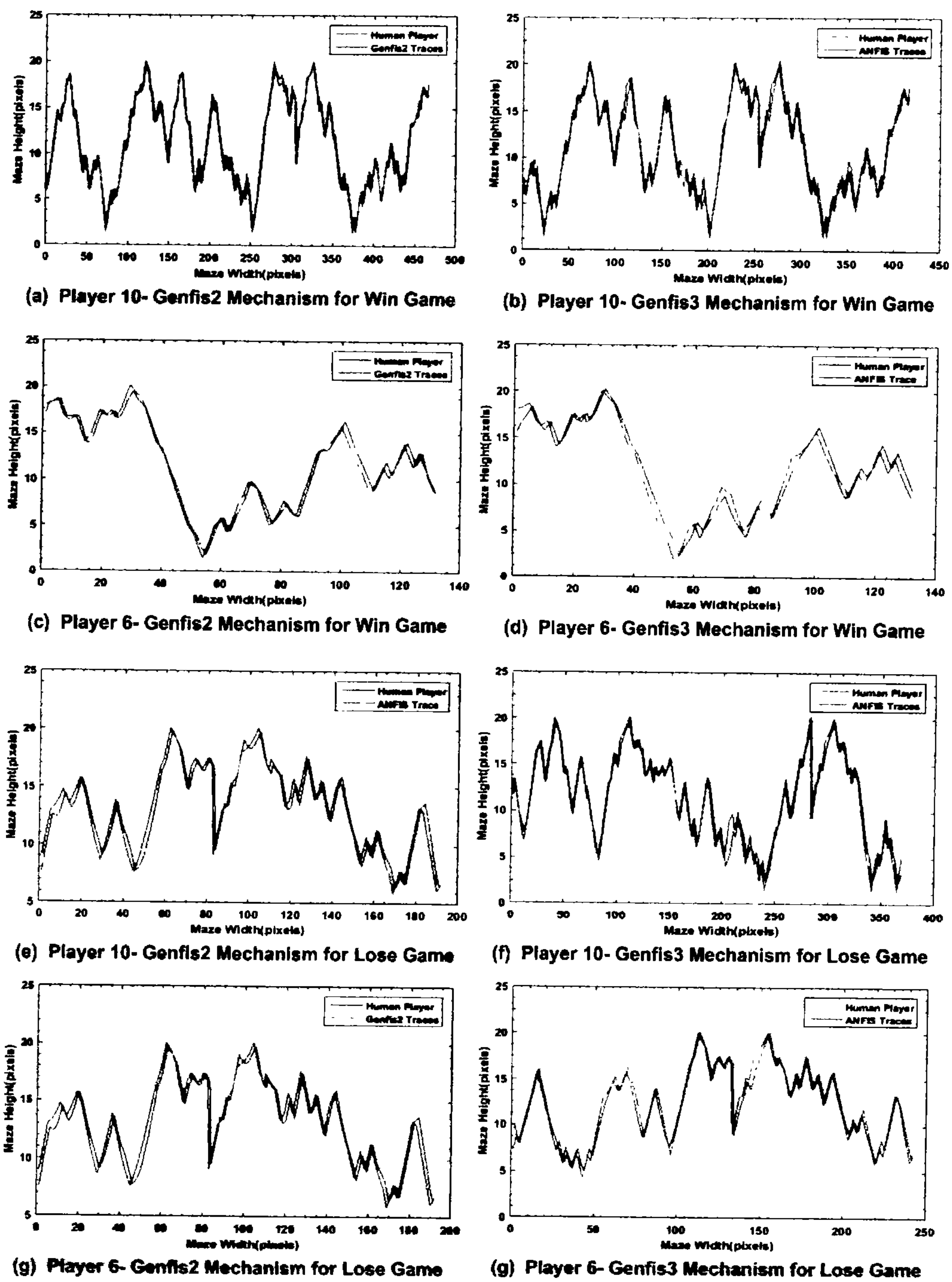



**Figure 3.8: Game play traces from two human players (Player No 6 and 10) and traces generated by *Genfis2* and *Genfis3* architectures in their attempt to imitate those players.**

**Table 3.8: The statistical values obtained from applying the Mann-Whitney Test on pair of Player and AI agents.**

| Players | *Genfis2* | | *Genfis3* | | *GRNN* | |
|---|---|---|---|---|---|---|
| | Win | Loss | Win | Loss | Win | Loss |
| | *P-Value* | *P-Value* | *P-Value* | *P-Value* | *P-Value* | *P-Value* |
| **Player 1** | 0.94 | 0.94 | 0.97 | 1.00 | 0.38 | 0.38 |
| **Player 2** | 0.86 | 0.72 | 0.77 | 0.95 | 0.77 | 0.40 |
| **Player 3** | 0.91 | 0.97 | 0.92 | 0.65 | 0.83 | 0.82 |
| **Player 4** | 0.96 | 0.96 | 0.64 | 0.95 | 0.75 | 0.96 |
| **Player 5** | 1.00 | 0.83 | 0.65 | 0.93 | 0.80 | 0.54 |
| **Player 6** | 0.96 | 0.71 | 0.82 | 0.93 | 0.87 | 0.23 |
| **Player 7** | 0.98 | 0.96 | 0.93 | 0.42 | 0.78 | 0.93 |
| **Player 8** | 0.71 | 0.97 | 0.92 | 0.76 | 0.87 | 0.70 |
| **Player 9** | 0.95 | 0.97 | 0.93 | 0.84 | 0.33 | 0.86 |
| **Player 10** | 0.98 | 0.76 | 0.98 | 0.98 | 0.05 | 0.67 |
| **Averages** | 0.92 | 0.88 | 0.85 | 0.84 | 0.64 | 0.65 |
| **All Averages** | **0.90** | | **0.84** | | **0.65** | |

Genfis2= Subtractive Clustering; Genfis3= Fuzzy c-means clustering;
GRNN=Generalized regression neural network;

**Table 3.9: The Cosine Similarity Test values obtained on pair of Player and AI Agents.**

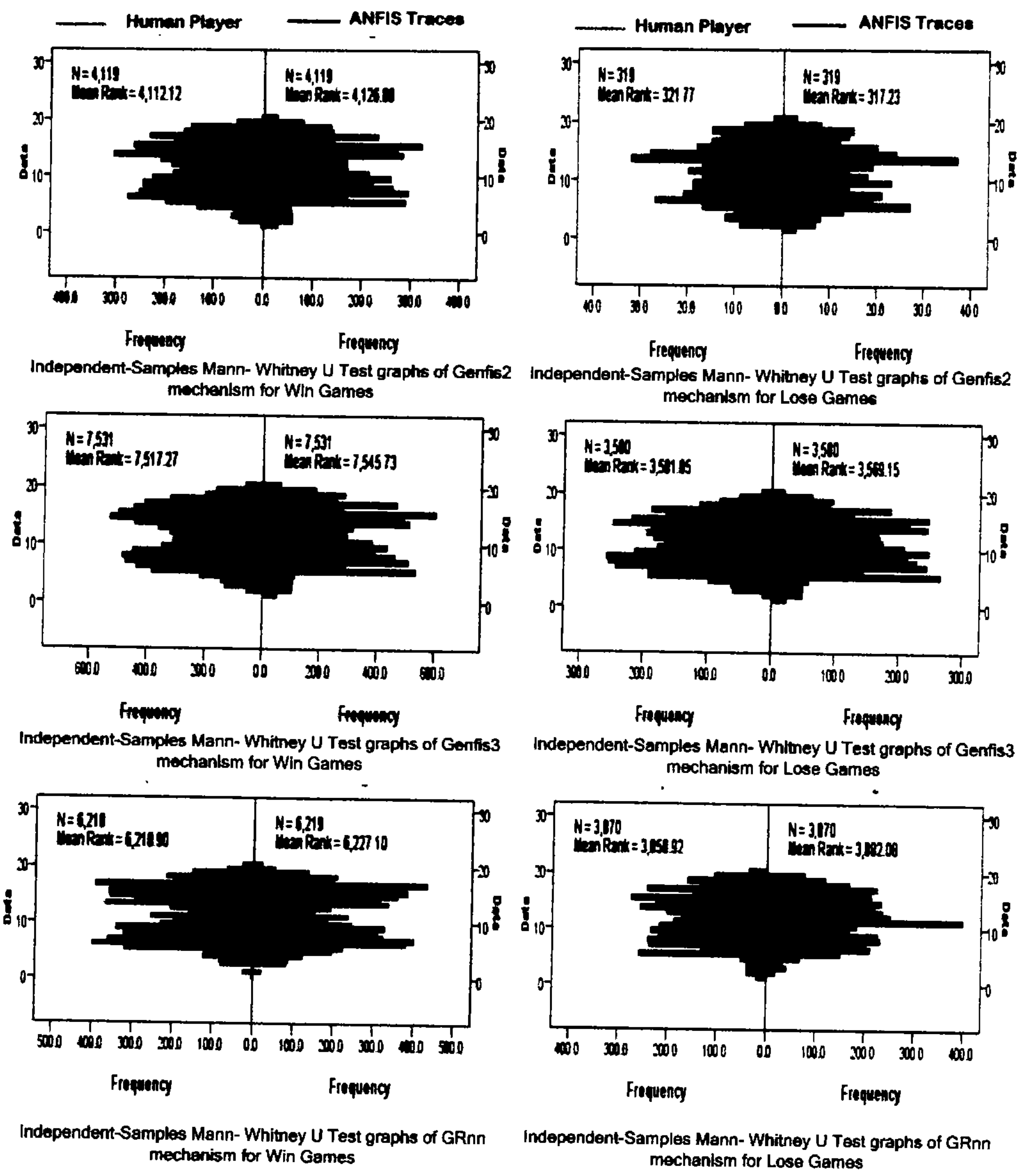
| Players | *Genfis2* | | *Genfis3* | | *GRNN* | |
|---|---|---|---|---|---|---|
| | Win | Loss | Win | Loss | Win | Loss |
| **Player 1** | 0.998 | 0.996 | 0.995 | 0.997 | 0.995 | 0.945 |
| **Player 2** | 0.998 | 0.999 | 0.998 | 0.996 | 0.996 | 0.998 |
| **Player 3** | 0.997 | 0.996 | 0.988 | 0.996 | 0.999 | 0.999 |
| **Player 4** | 0.997 | 1.000 | 0.996 | 0.998 | 0.993 | 1.000 |
| **Player 5** | 0.998 | 0.997 | 0.996 | 0.997 | 0.999 | 0.999 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Player 6** | 0.998 | 0.997 | 0.996 | 0.997 | 0.999 | 0.995 |
| **Player 7** | 0.998 | 0.997 | 0.997 | 0.995 | 0.999 | 0.995 |
| **Player 8** | 0.996 | 0.999 | 0.995 | 0.998 | 0.995 | 0.997 |
| **Player 9** | 0.998 | 0.999 | 0.998 | 0.997 | 0.998 | 0.999 |
| **Player 10** | 0.998 | 0.996 | 0.997 | 0.996 | 0.996 | 0.999 |
| **Averages** | 0.998 | 0.998 | 0.996 | 0.997 | 0.997 | 0.992 |
| **All Averages** | **0.998** | | **0.996** | | **0.995** | |

Genfis2= Subtractive Clustering; Genfis3= Fuzzy c-means clustering; GRNN=Generalized regression neural network;

## 3.6. Chapter Summary

This research immersed in the endowment of hybrid artificial intelligence-based approaches in the domain of imitation of human-like behavior to provide a viable solution to synthesize a more precise prediction model. Towards these goals, we implemented a believability framework for simulating believable AI agents. In building this framework, recorded data of winning and losing gameplay of individual human players used to train AI agents. In building this framework, recorded data of winning and losing gameplay of individual human players used to train AI agents. This study used the Pac-man game as a testbed for the exhibition of human-like behaviour. Eight representative standard network architectural variations of ANNs and hybrid intelligent system, i.e., adaptive neuro-fuzzy inference system (ANFIS) was used to simulate the Human player behaviour. This work has also been significantly expanded to include hybrid systems i.e., Adaptive neuro-fuzzy inference system (ANFIS) for human behaviour modeling issues in digital games. This research applied these methods for the first time to imitate human-like behaviour. On the other hand, to find the ideal ANFIS system, we trained the system with Grid partitioning, Subtractive clustering, and Fuzzy C-Means clustering methods. Using these methods, we have evaluated the similarity of best-performing agents. GRNN (Generalized regression neural network) surpasses eight neural network architectures by showing the best results with the smallest variance




Independent-Samples Mann- Whitney U Test graphs of Genfis2 mechanism for Win Games

Independent-Samples Mann- Whitney U Test graphs of Genfis2 mechanism for Lose Games

Independent-Samples Mann- Whitney U Test graphs of Genfis3 mechanism for Win Games

Independent-Samples Mann- Whitney U Test graphs of Genfis3 mechanism for Lose Games

Independent-Samples Mann- Whitney U Test graphs of GRnn mechanism for Win Games

Independent-Samples Mann- Whitney U Test graphs of GRnn mechanism for Lose Games

**Figure 3.9: Comparison of Independent-Sample Mann-Whitney U Test Graphs from human players and traces generated by *Genfis2, Genfis3* and *GRNN* architectures in their attempt to imitate those players.**

and average accuracy of 88% for both winning and losing behaviour. The increasing number of training and testing demonstrations does not affect GRNN performance, so it remains consistent. In general, subtractive clustering performance is overall best with an average accuracy of 94% along with Fuzzy C-Means followed closely with an average accuracy of 87%. In other words, we can say that two-hybrid methods outperformed the direct imitation method regarding the accuracy score. All the agents are trained performed the best. In general, subtractive clustering declared the most human-like AI agent with both automatic methods that have 90% similarity scores with Mann-Whitney U Test and 99% with Cosine Similarity Analysis. The results of the studies suggest that the model identified the best by the proposed framework do contribute to making virtual agents being perceived as more believable. Ultimately, these techniques need to be verified in a complex and immersive virtual world with higher dimensions. We believe that the proposed method can be further exploited in the field of agent-agent coordination, advanced methods for building more believable AI-agents, in a complex and immersive virtual world, hybrid recommendation systems and artificial intelligence-based expert systems. Moreover, the promising results could help for the development of human-like agents by testing the playability of generated levels in games like Super Mario Bros [191].

# Chapter 4: Performance of Case-based Imitation Agent through Case Knowledge Extraction

## 4.1. Overview

In this chapter we examine the impact of the pre-processing in the transition of raw cases without compromising the imitative performance of the imitation agent. In particular, this research examines a variety of algorithms and pre-processing techniques to determine which algorithm performs better than the others and is most beneficial for the case-based imitation system. This study is using the philosophies of rough set-based and FP-based case selection approaches. Knowledge transferring from an expert to a software agent is often a tough application that requires a substantial development time and programming effort [143] . This requires modeling expertise in a way that can be interpreted by a software agent. Even though creating a software agent, it is probably only possible to implement a set of specific tasks in a particular field.

Therefore, it requires additional or modified expertise to perform new tasks. Learning through demonstration could be a natural alternative to model expert knowledge for people without such technical skills. Specifically, the burden of training will be shifted from expert to the agent to imitate the expert behaviour when faced with the similar task. This happens with case-based reasoning (CBR), where cases consist of expert performed actions as inputs. Case-based reasoning (CBR) is a reasoning methodology that solves current problems using prior problem instances and their solutions.

To solve the problem of a CBR system, the knowledge contained in a case base is very pivotal. Therefore, there is a trade-off between the retrieval performance and the number of cases. Initial work in maintenance level knowledge checks the policy for removing cases from the basic case [160]. In case-based reasoning, the most similar cases are considered as the solution to an input problem after comparison to cases stored. Therefore, for every problem provided, the case-based reasoning system has to perform many comparisons between the problem and cases in the case base. These comparisons, whether of similarity or dissimilarity measures, can give rise to computation time required by the CBR system. The computational time required can be reduced by removing non-informative features in a case; called feature-reduction.

In case-based reasoning, the current situation is equated compared to the situation faced earlier to find the right solution for the current situation. To find the appropriate solution in case-based reasoning, the current problem is compared with the earlier experienced problem. The performance of the CBR system can relate to the number of problems that have previously occurred in the case base. This allows CBR to locate similar cases and the resultant solution to the current problem. However, in real-time environments, the case database must be searched within a real-time limit, which limits the number of cases in the case database. Several cases can be replaced with a single case, in which case it is replaced to maximize the case-based information.

During the collection of training data, there is a possibility that some or much of the collected data is very similar. There may be subtle differences in these similar data items that result in objects associated to different classes. However, these items often contain almost the same information, and it is less advantageous to keep all items than keep one item and discarding the rest.

## 4.2. Objectives and Research Questions

The key goal of this research is the pre-processing of a case-base attempted by an agent to imitate winning and losing game movements of the population of human players in the Pac-man game by recording one's game traces. The purpose is to reduce the time required to search the case base without affecting the diversity with minimal cases. This leads to the research question as following:

How the imitation of the case-based agent in the simulation game environment can be improved by pre-processing the case base?

Many methods have the potential to be used for this investigation. However, the feasibility of pre-processing will be limited to several methods. The following issues will be examined to answer the above mentioned research question:

1. **Feature Removal:** Observe the impact of execution time by removing the less relevant features of the case base without compromising the imitative performance of the imitation agent?

2. **Case Selection:** Observe the impact of replacing similar sets of cases, with the single case using different methods without compromising the imitative performance of the imitation agent?

## 4.3. Problem Definitions

This section contains an overview of the basic concepts used to understand Pawlak's rough set and Frequent Pattern theory along with some corresponding definitions.

### Terminologies:

*Information System (IS):* is a system that arranges data on a specific topic. Information is presented in a flat table, with each row representing an object and each column representing an attribute.

Often, $A$ indicates a finite set of attributes (features) representing the object information.

*Decision System (DS):* is an information system represents knowledge about the model contains decision attributes for specific objects

### Definition 1

Let S = $(U, A, f)$ be an information system, where U is a non-empty finite set of Objects $\{x_1, x_2, x_3 \ldots\ldots x_n\}$ called universe; and A is a finite nonempty set of attributes $\{a_1, a_2, a_3 \ldots\ldots a_m\}$; $f_a$ : U→ $V_a$ is an information function that maps an object of U to exactly one value in $V_a$ for a∈A, where $V_a$ is called the value set of a. A particular Information System called decision table DS= $(U, A \cup \{d\}, f)$, where $d \notin A$ is a distinguished attribute called *decision.* The elements of $A$ are called *conditions.*

### Definition 2: (Indiscernibility Relation).

#### Definition: (Lower and Upper approximations)

For a subset of objects **X** ⊆ **U**, Pawlak [193] define a pair of lower and upper approximations as follows:

$$\underline{IND(B)}(X) = \{x \in U : [x]_B \subseteq X\} \quad (4.1)$$

$$\overline{IND(B)}(X) = \{xU : [x]_B \cap X \neq \emptyset\} \tag{4.2}$$

## Definition 3

Let CB= {$C_1$, $C_2$, $C_3$………, Cn} is a given case base containing $n$ Cases where Ci (i ∈ [1…n]) is a cases which contains a set of traces in r= {r1,r2,r3………rn} where

r= [P ($X_0$; $Y_0$), …., P ($X_T$; $Y_T$), G1($X_0$; $Y_0$), …., G1($X_T$; $Y_T$), G2($X_0$; $Y_0$), …., G2($X_T$; $Y_T$)] is a sequence of entries, where each entry ($X_t$; $Y_t$) = {P ($x_c$, yc; $x_{c1}$, $yc_1$), G1($x_c$, yc), G2($x_c$, $y_c$)} contains the state Xt at a given time t, and the actions Yt the Pac-man very next executed position at time t interpreted as output. The support of a pattern R, which is a set of traces, is the number of traces containing R in CB. R is a frequent trace pattern if the support of R is not less than a predefined minimum threshold support $\partial$. For a given case base CB with minimum support threshold $\partial$, find the complete set of frequent patterns L(N) and compare with ordered case base $CB_R$ is called the frequent pattern case base problem. All L(N) pattern with support < minSuport are considered as distinct or pivotal case.

## 4.4. Baseline Techniques

This research has selected rough set-based and novel FP-based case selection approaches as base line techniques.

### 4.4.1 Rough Set Theory

Rough-set is another common approach [193, 194], and has proven to be very useful in many different areas [195, 196]. Rough sets allow to identify the most informative features in a supervised manner and then select them by the reduction calculation. The methods of case selection can affect the quality of the selected case knowledge base after reducing features. The quality of the case base knowledge can be influenced by the reduced case volume. In Computer Science Zdzislaw Pawlak [193, 194] proposed a new theory called rough set theory (RST) to study classification analysis and to synthesize concept approximation from the uncertain or inexact data in the artificial application. The primary motive of rough set theory establishes theoretical models of indiscernibility and discernibility notations to describe indistinguishable objects. The entire modeling process consists of a sequence of multiple sub-steps, which all require various concepts, i.e. lower and upper approximations for adjustments. Rough-Set is based on the philosophy of those observations, which cannot be distinguished by specific information. The indiscernibility of objects is due to imperfect information. The indistinguishability (equivalence relation) induces approximation spaces from equivalence classes of indistinguishable objects. This form of knowledge representation, which is provided in rough sets, is free from redundancies that are needed in real applications. Rough-set algorithms and models can be used to support decision-making optimization, dissimilarity analysis, rule generation, and symbolic learning.

### 4.4.2 Frequent Patterns

For the last two decades, discovering frequent itemsets remains a silent feature to find the association and correlation among mass data [197]. Most primary and early sequential pattern mining algorithms adopt an Apriori property which is based on an anti-monotone heuristic [198]. According to this property, any sub-pattern of a frequent pattern must be frequent. Although the seminal work using the Apriori heuristic achieves a significant gain in reducing candidate sets, it still suffers from the following limitations. 1) It is expensive to process many candidates' sets because too many passes through the database are required. 2) It is cumbersome to repeatedly scan the database and examine many pattern matching candidates, particularly for the mining of long patterns and become a cause of increased I/O load.

An FP-tree-based method called FP-Growth presented in [199] proved to be an excellent solution to the above issues. FP-Growth adopts a pattern fragment growth method and therefore scans the database only twice to evade excessive generation of large candidate sets. It compresses the database directly into a frequent pattern tree, rather than using a candidate set, and association rules are generated using the FP tree. Because of its popularity, and many implementations, we have decided to base our work on the original FP growth algorithm. Frequent patterns are generated through the steps of FP-Tree generation, Conditional Pattern Bases and Conditional FP-Trees in the FP-Growth algorithm. The formation of frequent item sets stores it in a database done by using DBMS (Database Management System). The database used is SQL Server 2012. This research is concerned with the problem of feature and case reduction through the stages of generation frequent patterns.

A novel approach is proposed by developing an ordered case base using L (1) frequent patterns in alphabetic preferences. Moreover, then compare L(N) frequent patterns with ordered case bases that match L(N) item sets. The matched cases replaced with a single case and then updated the original case base. This would result in a reduced case base size, but preferably holding similar information as previously obtained. We have applied this approach with the hope that predictive accuracy will be improved or even maintained.

### 4.4.3 Case Knowledge Extraction

The case knowledge process is summarized in Figure 4.1. For removing redundancy, reducing case base size while preserving competency, we consider the following tasks. 1) Learning similarity measures 2) feature and case reduction. Primarily in case selection methods, the calculation of case coverage sets, reachability sets, and the search for k nearest neighbours is based on a similarity calculation using all the features involved. By training models using the weight of features, or using domain knowledge, the importance of features can be determined in advance. However, each method has its advantages and disadvantages, which present challenges for FR and CR. For determining the feature weights in advance while using the domain knowledge, the necessary domain knowledge is either obtained by questioning experts or extracting from the cases. The first approach is labor intensive and the second one increases the training load. Likewise, when neural networks and decision trees models are used to learn the feature weights, the training load is not trivial. To retrieve similar cases from the case base, for the unseen cases, the output in the form of a trained neural network or set of rules is not appropriate.

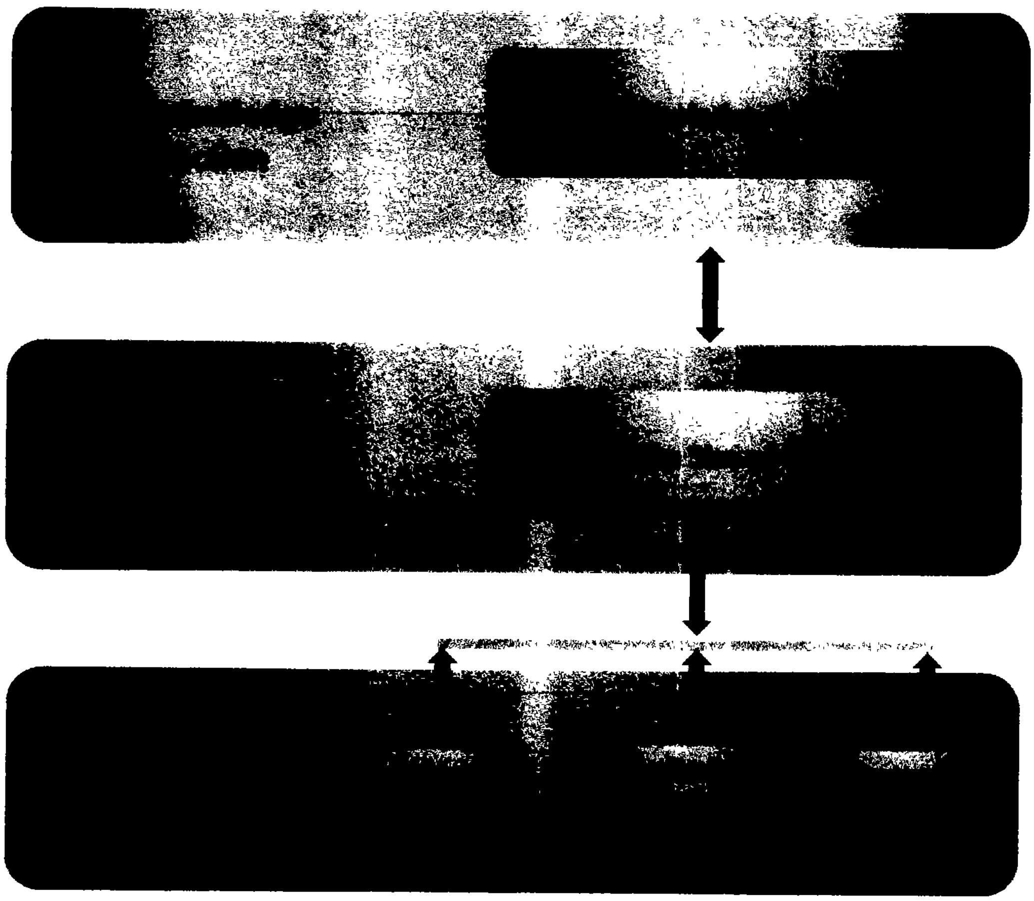


**Figure 4.1: The methodology of case Knowledge extraction**

An approach based on rough sets theory to solve these issues has been applied for FR and CS. The importance of functionality is to count with the generation of reducts. The features in reduct are considered the most important with higher accuracy and others are considered irrelevant. Domain knowledge is not required for the computation of the reduct, and the computational complexity is linear in respect of the number of attributes and cases.

Using an indiscernible class for FR and CS, the case representation is always the same as the original case base. The representation of knowledge in this form for unseen case retrieval is more practical and more comfortable to learn. Since only the reduction features are involved the runtime and case selection computations are thus reduced. Three main benefits are achieving for CBR classifiers using this approach: 1) classification accuracy can maintain or even improve, 2) storage requirements are reduced, and 3) the response time of the classification decision can be reduced when an unseen case occurs. In this method, we apply CS and FR interactively, determining the best approximate reduct. The interaction of FR and CS is reflected in the identification with the highest accuracy. The best set of sub-features is defined as approximate reduct R * with the highest accuracy. By definitions 1 and 2, the FR algorithm based on rough sets in our expanded approach is described in Algorithm 1.

## 4.5. Proposed Framework

The research methodology adopted for this research has been widely used in literature testing the effects of different algorithms on the case-based reasoning process [3, 45] and case-based agent imitation field [188, 200]. This section sheds light on the functionality and architecture of the proposed scheme.

The methodology includes the following steps with details of our application:

1. *Develop a Case base:* An initial case base is generated automatically by observing human player while the Pac-man game is playing (See Algorithm 1).
2. *Pre-processed Case base*: Two proposed pre-processing techniques Rough Sets Similarity Based Learning and Frequent pattern-based Case Knowledge Extraction are then applied to the unprocessed case base (See Algorithm 2& Algorithm 3). This creates a pre-processed case base. Pre-processed case bases are generated for each of the pre-processing algorithms being examined. This methodology step is the focus of this research.
3. *Observe the behaviour of the imitation case-based system after pre-processing:* Compare the results of pre-processed cases bases and original unprocessed cases base by using various metrics (See Section 4.7).

**ALGORITHM 1. Case-Based Imitation Framework**

**Input:** For a Given case-base CB containing a collection of n cases, case $C_p$ representing a new problem,
SM representing similarity measure, SA solution algorithm, and cardinality of nearest
neighbourhood |NN|=Nn.

**Output:** Predicted solution $sol_P$ of $C_p$ Accuracy x in the form next predicted the position
$(Px_{c1}, Py_{c2})$
*For each Cj* $\in$ *CB*
$sim_j$=findSimilarity ($C_p$, $C_j$, SM)
$w_j = sim_j$
End For
$NN$ = extractNeighbour($N_n$)
$sol_P$=aggregate ($NN$,$SA$,w)

**ALGORITHM 2. Rough Sets Similarity Based Learning**

**Input:** For a Given case-base CB containing a collection of n cases and A Features
Information System S= (U, A)

**Output** Reduced Case Base

**Method**

1. *Initialize* P= Ø , *Accr* = Ø

*P will store reduced case base after FR; Accr will store resultant classification accuracies using these reduced case bases.*

2. Compute indiscernible matrix for each feature 1→A
$M(S)=(C_{ij})$
3. Reduce M using indiscernible class
*For a given feature, choose one case from each equivalent class/ indiscernible class.*
*and populate the reduced CB in the following way.*
$P \leftarrow P \cup \prod_R(U)$
d - a number of non-empty fields $C_1, C_2 \ldots C_d$
R- build families of sets $R_0, R_1, \ldots, R_d$
begin

R0= Ø

For $i$ = 1 to $d$

Implement Algorithm 1 for test *case classification using* $\prod_R(U)$*;* *(Output the current accuracy, a)*
*Accr* ← *Accr* ∪{y}.
*Find y**
*y* = max {y € Accr}; and find the corresponding R*.*

end
*Output* the reduced final case base corresponding to R*, denoted by $CB^{*} = \cup\prod_{R^{*}}(U)$.

**ALGORITHM 3. Frequent pattern-based Case Knowledge Extraction**

| | |
|---|---|
| **Input:** | Take unprocessed Case base CB of *n* cases, minimum support Threshold |
| **Output:** | L(N) Frequent items set |
| | |
| ***Method:*** | |
| **Step 1:** | Collect the *L(1)* set of frequent item of size 1 sets and their support |
| **Step 2:** | Sort F1 in support order as a prefix<br>Develop an ordered Case base $CB_R$ using *L(1)* in the alphabetic preference |
| **Step 3:** | Find set of frequent items of size *N* from the ordered Case base $CB_R$, Where the N=Total number of Features |
| **Step 4:** | Compare *L(N)* frequent itemsets with ordered case base $CB_R$ developed in Step 3 to find the cases that match *L(N)* item sets. |
| **Step 5:** | Remove redundant cases from the case base formed at step 4. |
| **Step 6:** | Update the original case base CB. |

### 4.5.1 Case-Based Model for Imitation Behavior

Before CBR is executed, we need to choose a measure of similarity (SM). The final solution found with a set of nearest neighbors NN having some cardinality $N_n$ and a solution algorithm SA. Every case $C_j$ is compared with current $C_p$ using SM in CB. To find the final solution $sol_p$ of $C_p$, $sim_j$ similarity is measured to treat as weight $w_j$ of $C_j$. These decisions may differ from one area to another. The nearest neighbor set NN is extracted based on the calculated similarities, and NN is aggregated using SA and w to find $sol_p$. The case study is presented as numerical attributes between 1 and 18. If $C_p$ is an entirely new experience and its closest match in CB does not exist, an adjustment mechanism can be used to revise the solution. To preserve the vector $<C_p, solp_i>$ as a new CB experience, an automatic decision mechanism is necessary. This decision depends on the new knowledge obtained from this new experience.

#### 4.5.1.1. Case Representation

A case C=<p, O> denotes as a problem p and outcomes O represented as a linear vector of attributes. A case represents a discrete-time snapshot of three agents' (Pac-man, Ghost1, and Ghost2) view in the form of their current position extracted from capture state of monitor resource. In this case, the performed action will be the agent's next position. The whole game is therefore represented as a series of cases C= $\{c_1, c_2, c_3, \ldots c_n\}$. Here each case 'C' is presented as an aggregation of all agents (Pac-man, Ghost1, and Ghost2) type, position, and subsequent action.

A problem 'p' is consisting of problem description and outcomes are the solution. The description of the problem consists of six features Pacman current position ($Px_c$, $Py_c$) and Ghost1 ($G1x_1$, $G1y_1$) and Ghost2 ($G2x_1$, $G2y_1$) corresponding positions. In this case, outcomes will be the Pacman agent's next position to predict. A vector is used to represent the collection of n parameters as $i^{th}$ case of the case-base as follows [45]:

$$\mathbf{Ci=\sum Pxy= (Pxy_1, Pxy_2, \ldots, Pxy_n)} \qquad (4.3)$$

Where $\sum$ represents a collection of parameters. A sample case base CB containing z cases is represented as follows:

$$\mathbf{A=\sum Cm= (C_1, C_2, \ldots\ldots\ldots, C_z)} \qquad (4.4)$$

#### 4.5.1.2. Case Similarity

We developed a CBR system using nearest neighbor retrieval algorithm [7] to test the collection of measurements of similarities in our research. The fundamental purpose of this CBR system is to predict the behaviour of test trace $R^{test}$ by learning from a collection of training traces R = $\{R_1, \ldots\ldots Rn\}$. When an entry **(xt; yt)** € $R^{test}$ is given in the test trace, the CBR system finds the most similar state to xt from training traces and predict the very next move of the human player. Then we determine the prediction error against yt. In this work, seven commonly used similarity functions for retrieval were used to find the nearest neighbors of the current case. A comparison of these similarity measures based on accuracy against a variable number of nearest neighbors. The similarity functions implemented are given in Table 4.1.

**Table 4.1: Similarity Functions Used.**

| Similarity measure name | Similarity measure definition |
|---|---|
| **Euclidean distance** | $dij = \sqrt{\sum_{k=1}^{m} (wk(P_{ik} - P_{jk}))^2}$ |
| **Hamming similarity** | $sim_{ij} = \frac{matches_{k=1}^{m}(P_{ik,}P_{jk})}{m}$ |
| **Manhattan distance** | $d_{ij} = \sum_{k=1}^{m} w_k \lvert P_{ik} - P_{jk} \rvert$ |
| **Canberra distance** | $d_{ij} = \sum_{k=1}^{m} \frac{\lvert P_{ik} - P_{jk} \rvert}{P_{ik} + P_{jk}}$ |
| **Bray-Curtis distance** | $d_{ij} = \frac{\sum_{k=1}^{m} \lvert P_{ik} - P_{jk} \rvert}{\sum_{k=1}^{m} P_{ik} + P_{jk}}$ |
| **Squared Chord distance** | $d_{ij} = \sum_{k=1}^{m} (\sqrt{P_{ik}} - \sqrt{P_{jk}})2$ |
| **Element-wise binary operation** | $d_{ij} = \sqrt{\sum_{i=1}^{n} (xi - ai)^2} < \sqrt{\sum_{i=1}^{n} (xi - bi)^2}$ |

### 4.5.1.3. The cardinality of nearest neighborhood

The solution algorithm used next neighbor Set (NN) as input to develop the solution with a varying cardinality of NN from 1 to the size of the case-base. The purpose of this analysis is to find a similarity measure that requires the least cardinality of NN to get the best results. The purpose of this analysis is to find a similarity measure that requires the least cardinality of NN to get the best results.

## 4.6. Experimental Design of Case-based Imitation through Case Knowledge Extraction

In these experiments, we designed software agents based on a technique that was developed specifically for playing humanly. The primary aim of this study is how the software agent's imitation ability can improve with pre-processing the case base it uses? Moreover, to derive a measure of imitation of human behaviour from the likelihood ratios between the models and which model gives best results.

### 4.6.1 Data Collection

To measure the performance of the selected machine learning approach, studied in this work requires traces of human player from game sessions to imitate human play. Traces of human gameplay are needed to measure the humanness of controller by comparing it to the human player.

Dataset was used for the experiments consists of the data of 30 games (15 winning games and 15 loss games) to correctly model winning and loss behavior of twenty-six (26) players, to measure the overall performance of CBR proposed. Dataset used to investigate the normal playing style of a population of players. Table 4.2 presents descriptive statistics for both the dataset.

**Table 4.2: Player's Traces of Winning and Losing Game Play Dataset 2.**

| **Players** | **Sample Size** | |
|---|---|---|
| | **Win** | **Loss** |
| **26** | 40566 | 22366 |

The model of these two patterns of players will conclude what moves let the players lose the game and that move takes them to victory. Each imitation-based model trained on winning and loss game patterns separately and it is a non-trivial exercise.

### 4.6.2 Data Analysis

All observed cases in Section Data Collection for the agents are used to create a single case base, also called a complete case base. The complete case base was then used to create test and training cases. The training and test case bases are created by random selection of cases from the complete sample case bases.

- **Training case base:** Different experiments may use cases bases of different sizes. For experiments that require an N-size training case base, the training case base will contain the N cases selected randomly from the base of the full sample case base for the agent under test.
- **Testing case base:** Each of these test cases is used as an input for a case-based reasoning system and an output action from a case-based reasoning system compared to the actions in the test base. This comparison is used to calculate various metrics that measure imitation system performance.

N-fold cross-validation, Leave-N-out validation, and so forth were not used because we use algorithms to reduce redundancy as a pre-processing for CBR. For example, rough set theory and Frequent Pattern theory presented in Section 4.4.1 and 4.4.2 attempt to identify pivotal cases and combine into a single case. Therefore, after merging, we would assume that every remaining case is different from all other remaining cases. Now if we subsampled the case base into training and testing, we would not expect a test case to have a similar case in the training set, so the similarity measures used would only find dissimilar neighbours.

### 4.6.3 Performance Metrics

The same metrics as described in Section 3.3.4 has used in this work to represent an error, accuracy, and believability of playing style of developed agent.

## 4.7. Results and Discussion

The experiments in this chapter will examine a variety of algorithms and pre-processing techniques to determine which algorithm performs better than the others and is most beneficial for the case-based imitation system.

The following comparisons will be performed:

H1: To find the appropriate similarity measures with the best fidelity for the selected genre of a video game after comparing seven standard similarity measures for the retrieval of nearest neighbors and which one is more computationally efficient concerning time?

H2: To determine the improvement of the pre-processing technique over a non-preprocessed case base, compare each pre-processing algorithm with the use of no preprocessing.

H3: What is the impact of pre-processing methods on agents' ability to imitate?

To illustrate the main aim of this study, we describe the results of agents trained to human gameplay traces; evaluate its performance and their human-likeness in playing games in this section. A common dataset used for all experiments performed but, reflecting winning and losing behavior to imitate the human gameplay as explained in the dataset section. The first contribution of our experimental work is to find the best similarity measure for the selected genre of a video game after comparing seven standard similarity measures. Accuracy and mean square error (MSE) selected as the measure of performance of the proposed method for this case study. The results of this empirical study were analyzed over the following dimensions. Table 4.3 presents the performance of different similarity measure with the best accuracy.

**Table 4.3: Highest Accuracy Achieved with Different Similarity Measures.**

| *S.no.* | *Similarity Measure* | *Accuracy* |
|---|---|---|
| 1 | Euclidean distance | **82%** |
| 2 | Hamming similarity | 59% |
| 3 | Manhattan distance | 57% |
| 4 | Canberra distance | 52% |
| 5 | Bray-Curtis distance | 19% |

| | | |
|---|---|---|
| 6 | Squared Chord distance | 40% |
| 7 | Element-wise Binary Operation | **82%** |

The empirical study shows that two similarity measures among the selected ones are similarly reasonable, as shown in Table 4.3. They are Euclidean distance and Element-wise Binary Operation distance. Further, to look at the computational load, we have demonstrated a speed test to compare both methods. The element-wise binary operation was implemented by 'bsxfun' function and Euclidean function by 'norm' function with Matlab. bsxfun is one of the multithreaded Matlab functions. The results are presented in the Figure 4.2. As it can be depicted from the graph that bsxfun is about 18% faster than norm function in winning and losing game play traces. The difference becomes stronger as the arrays grow larger.

With bsxfun, the element-wise binary operation is applied to matrix based on applying an arbitrary operator to all pairs of elements of two vectors, and not explicitly working on two matrices obtained by replicating the singleton-expansion enabled vectors.

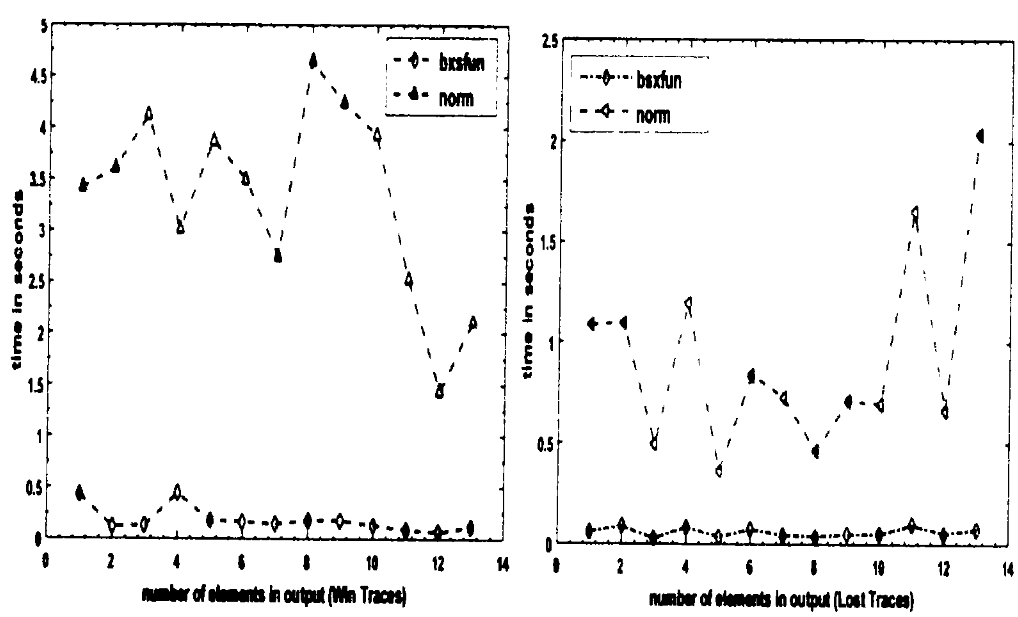



**Figure 4.2: A speed comparison between Element-wise Binary Operation (bsxfun) and Euclidean distance (norm) Similarity Measures.**

**Experiment 2** was conducted to test and compare our proposed algorithms, rough set and FP- based techniques to reduce the dimensionality and find the redundant cases in the case base. We use three main indices: predictive accuracy, reduced features, and reduced case to prove their effectiveness.

We evaluated the performance of our approaches using accuracy with a number of loss functions. L-0 accuracy evaluates how many times the model generates a lineal combination of the same (x, y) coordinates and direction as the human from the training set. Besides computing L-0, the accuracy metric L-1 computes a second accuracy value the ratio between the numbers of times the model chose (x, y) coordinates with a difference of one step in the same direction. Same is the case with accuracy with L-2. The reduced storage means the percentage cases selected for final case base after applying proposed algorithms. The case base that contains the reduced feature set and the selected cases represent the final reduced case base.

The training and testing dataset is randomly selected 30% as test data; and 70% as training data. 28400 cases initially used as training and 12156 cases as testing of winning games. For losing games 15659 cases for training and 6710 cases as testing. This distribution of cases remains the same for all subsequent experiments.

Table 4.4 and Table 4.5 illustrates the comparison of rough set and FP-based proposed approaches to identify the redundant cases regarding storage reduction for winning and losing games cases base. The losing game case base has 15,659 initial training cases. When the data was processed using the Rough Set algorithm, the number of cases was reduced to 12876 (17.77%) with six features. The number of cases was significantly reduced to 12487 (20.25%) of the number cases in the initial training cases (15659) when processed using FP- based algorithm. The same trend was provoked in Table 4.6 for winning game data set.

Table 4.5 and Table 4.7 demonstrate the improved accuracy when using proposed algorithms for winning and losing game data. Here, P0 represents the original accuracy with the entire data set, while P(RS) denotes the accuracy after applying Algorithm 2 and P(FP) denotes accuracy after applying Algorithm 3. Table 4.6 shows that accuracy is improved when applying RS using a reduced data set for 6Features-3Features of winning games dataset. For 1Feature 2Feature, accuracy decreases slightly due to features reduction. After applying FP, the results in Table 4.6 depicts that the accuracy using reduced data set improves for all reduced feature set 6Features-1Features.

**Table 4.4: Algorithms Comparison for reduced storage- 40556 Initial Cases of Winning games.**

| Features | Initial Training Cases | Reduced Instances (RS) | Reduced Instances (RS)% | Reduced Instances (FP) | Reduced Instances (FP)% |
|---|---|---|---|---|---|
| 1 Feature | 28400 | 100 | 99.6 | 86 | 99.69 |
| 2 Feature | 28400 | 252 | 99.11 | 92 | 99.67 |
| 3 Feature | 28400 | 3088 | 89.12 | 2207 | 92.22 |
| 4 Feature | 28400 | 11309 | 60.17 | 7616 | 73.18 |
| 5 Feature | 28400 | 21414 | 24.59 | 17118 | 39.72 |
| 6 Feature | 28400 | 22701 | 20.06 | 20380 | 28.23 |

**Table 4.5: Algorithms Comparison for reduced storage- 22366 Initial Cases of Losing games.**

| Features | Initial Training Cases | Reduced Instances (RS) | Reduced Instances (RS)% | Reduced Instances (FP) | Reduced Instances (FP)% |
|---|---|---|---|---|---|
| 1 Feature | 15659 | 167 | 98.93 | 129 | 99.17 |
| 2 Feature | 15659 | 405 | 97.41 | 315 | 97.98 |
| 3 Feature | 15659 | 3650 | 76.69 | 3051 | 80.51 |

| | | | | | |
|---|---|---|---|---|---|
| 4 Feature | 15659 | 8742 | 44.17 | 7681 | 50.94 |
| 5 Feature | 15659 | 12442 | 20.54 | 11554 | 26.21 |
| 6 Feature | 15659 | 12876 | 17.77 | 12487 | 20.25 |

**Table 4.6: Algorithms Comparison for improved accuracy- 40566 Initial Cases of Winning games.**

| Features | P0(RS) % | P(RS)% | P0(FP)% | P(FP)% |
|---|---|---|---|---|
| 1 Feature | 16.17 | 18.00 | 6.42 | 20.09 |
| 2 Feature | 72.66 | 77.00 | 84.33 | 83.46 |
| 3 Feature | 78.70 | **82.66** | 83.00 | **83.96** |
| 4 Feature | 78.08 | 79.98 | 82.52 | 82.86 |
| 5 Feature | **79.41** | 81.78 | 80.32 | 80.79 |
| 6 Feature | 72.39 | 75.68 | 80.68 | 82.97 |

**Table 4.7: Algorithms Comparison for improved accuracy- 22366 Initial Cases of Lose games**

| Features | P0(RS) % | P(RS)% | P0(FP)% | P(FP)% |
|---|---|---|---|---|
| 1 Feature | 20.35 | 17.07 | 19.90 | 19.90 |
| 2 Feature | 66.45 | 62.34 | 75.55 | 75.96 |
| 3 Feature | 79.78 | **83.00** | 76.26 | **83.90** |
| 4 Feature | 78.00 | 77.67 | 77.66 | 77.93 |
| 5 Feature | 80.31 | 80.53 | 74.05 | 74.85 |

| | | | | |
|---|---|---|---|---|
| 6 Feature | 68.05 | 71.48 | 70.15 | 70.75 |

The highest accuracy of around 83 percent achieves by both approaches when the feature set is reduced to 3 features. Since the accuracy reaches its maximum when the feature set reduced to 3, this could be reported as the final reduced case base.

In case of Lose gameplay traces, Table 4.6 shows that accuracy is improved when applying RS using a reduced data set for 6Features-2Features. For 1Feature, accuracy decreases slightly due to features reduction. After applying FP, the results in Table 4.7 depicts that the accuracy using reduced data set improves for all reduced feature set 6Features-1Features. The highest accuracy of 84 percent achieved by both approaches when the feature set is reduced to 3 features. Since the accuracy reaches its maximum when the feature set reduced to 3, this could be reported as the final reduced case base.

***Summary:*** *In* general, the results mentioned above lead to the following observations: 1) the two approaches proposed have been able to reduce cases substantially. 2) The prediction accuracy is improved or even preserve when applying both approaches. The FP growth approach remains consistent to achieve greater accuracy for all reduced feature sets and with reduced cases. The accuracy achieved by both approaches is nearly the same for selecting the final reduced case base. FP-based algorithm has a more powerful capability to reduce redundant cases than rough set-based Feature reduction approach. The accuracy reaches most of its maximum when the feature set reduced to 3 by both approaches.

### 4.7.1 Evaluation of Agents of Playing Style Identification

H2: To determine the improvement of the pre-processing technique over a non-preprocessed case base, compare each pre-processing algorithm with the use of no preprocessing.

H3: What is the impact of pre-processing methods on agents' ability to imitate?

An experiment 3 performed to determine the impact of pre-processing methods on agents' ability to imitate. And to evaluate the believability of the obtained AI agents, the traces left by the agent and players were compared. We used automated methods rather than human assessment. Because they judge the game AI subjectively, are often inaccurate and are time-consuming because of the involvement of many people to assess agents. The size of human testers to get adequate results of the evaluation by human judgment is, however, a great challenge[58, 192]. The limits of human abilities are also a major downside in human assessment, i.e. Turing test. To achieve this, statistical method Mann-Whitney U test and Cosine similarity methods were used. Human perception, however, is the ultimate judge of the resulting quality of game character believability [58].

#### 4.7.1.1. Mann-Whitney U Test

This part of the study extrapolates the Mann-Whitney U Test to the believability of the obtained agents. To find the answer to the question: "Is there a significant difference between the traces of human players and the agents before and after pre-processing techniques?", Traces of human players and the agents were compared using a non-parametric statistical technique Mann Whitney U test. Mann Whitney U test was also applied by[4], but to compare the results of Turing crowdsource test to find the significance between each pair of agents. Table 4.8 represents the findings of the results of Mann Whitney U test for the win and losing traces of human player. All values of p obtained are >0.05 with a margin. The imitative agents before and after pre-processing showed the significant statistical difference and hence improved imitative agent's ability. The close Mean rank scores of the human players and agents indicate that the two populations are the same in statistical sense as shown in Table 4.8. Based on the results obtained, it could be stated that traces generated by agents are the same to human player traces.

Among all comparisons, the Mann-Whitney test statistics include that all players trace distributions are the same across all AI-agents play styles in both categories. The results in Table 4.8 reveals a notable increase of 25% in the statistical values of the Mann-Whitney test when using cases after preprocessing by both FR and FP based agents; however, there is not a significant difference between FP and FR, *i.e., only 2%.* We can conclude from these results that the FP-based agent is the most human-like a Pac-man agent, getting higher averages of the p-value scores 90% both in wins and losing game traces.

**Table 4.8: The Statistical Values of the Mann-Whitney Test on Pair of Player and Agents.**

| | **P(FR)** | **P(FP)** | **P(FR)** | **P(FP)** |
|---|---|---|---|---|
| | Win | Win | Loss | Loss |
| | *P-Value* | *P-Value* | *P-Value* | *P-Value* |
| **Before Processing** | 0.676 | 0.671 | 0.676 | 0.671 |
| **After Processing** | 0.92 | 0.93 | 0.87 | 0.90 |

Our experiment reveals that the Pac-man agent in the Pac-man game has similar playing styles when using automatic judging methods.

## 4.8. Chapter Summary

In this experiment, we examine the impact of the pre-processing in the transition of raw cases without compromising the imitative performance of the imitation agent. The experiments in this chapter have shown that pre-processing of cases increases the imitative performance of the imitation agent. We describe rough set-based and FP-based case selection approaches. In the rough set approach, the concept of equivalence classes has used to generate indiscernibility classes for each feature 1 to N. Then by removing the objects that cannot be distinguished from each other based on the available attributes the final discernible class formed from all other objects. A novel FP-based approach is proposed by developing an ordered case base using L (1) frequent patterns in alphabetic preferences. And then compare L(N) frequent patterns with ordered case bases that match L(N) item sets. The matched cases replaced with a single case and then updated the original case base. The results of the experiments illustrate that the FP based algorithm for case selection is the most significant algorithm with the highest accuracy and least storage requirements. Higher prediction accuracy and less storage space requirement were gained with each algorithm when compared to using the original case base.

# Chapter 5: Diagnostic Accuracy in Dependent Personality Disorders on fMRI Data

## 5.1. Overview

This chapter explores the cognitive state classification of decision-making and patterns of activation from the prefrontal cortex of human brain of the brain regions through functional magnetic resonance imaging (fMRI) data. This research implements the same framework proposed and implemented for imitation human-like player behaviour (See Section 3.2). The purpose is to validate the proposed architectures in both areas i.e., for Imitating human-like behaviour using Game data and to investigating the neural basis of rewardless-related human decision making using functional magnetic resonance Imaging (fMRI) data. The goal of our study is to help patients in making rewardless-related decisions by performing the respective task in a particular state. Towards this goal, we compared different variants of artificial neural networks (ANNs) to the adaptive neuro-fuzzy inference system (ANFIS) for useful analysis of brain activities.

## 5.2. Dependent Personality Disorders

Recently, the advancements in neuroscience studies captured the attention to find the activation pattern and functional connectivity of the brain regions through functional magnetic resonance imaging (fMRI) data. The human brain can process various kind of impulses which cause activation and functioning of multiple organs in the body. The functional connections of the brain regions enabled the brain to accomplish these tasks[12, 164]. The functional Magnetic Resonance Imaging (fMRI is considered as a powerful experimental technique for analyzing stimulus-based activation in the brain. The triggering of systematically cross-linked activated brain regions, has itself established a new trend in fMRI imaging through the analysis of multi-voxel patterns. The key observation of these studies is the brain regions are activated for specific mental activities. Due to the lack of tools, previous attempts to investigate the collaborative functioning of the human brain have been hampered. The advancements in instrumentation and in analysis techniques, have enabled a complete examination of brain while performing specific tasks. [13]. The high-resolution fMRI can access specific areas of the brain that are activated when a task is performed.

To perform such an fMRI analysis, Lueck et al. [201] and Friston et al. [14] developed an SPM package which maps the activated brain areas of the fMRI data. Neuroimaging determines the relationship between the responses of a particular task to the predefined associative areas. This is possible because, during the execution of a particular task tandem activations in the specific brain regions are performed [15]. While performing a particular task, it is necessary to isolate the resolution of brain regions from induced activations in brain regions.

To find the most activated brain regions, their specific cognitive processes and stimulus-triggered activities, several experiments were designed by using functional brain imaging data [16-18]. Identification of signals from high-resolution fMRI data allows us to classify them into pre-defined areas of the brain, such as the visual area or decision area. Belilovsky et al. [19] proposed a Voxel wise autoregressive study for fMRI to locate activations. Since the image (fMRI) contains thousands of voxels, the analysis of such a data-volume leads to an over-fitting of the classification algorithm[20], high level of noise and curse of dimensionality in the field of radiology [21, 165]. To accurately assess neural activation, the most popular techniques of Artificial intelligence technologies, i.e., machine learning and convolution neural networks were applied to decipher brain patterns for medical image analysis. For a more accurate classification, state-of-the-art ensemble techniques were employed [22, 23].

The artificial neural network (ANN) has been applied over fMRI complications and to develop diagnostic models. To identify the complex interactions among input data, ANN distinguishes the mechanisms of time series and results. This makes possible to recognize the hidden relationships which are usually invisible during traditional statistical analysis. ANN seems to be a promising tool for clinical decision making and have been used in several areas, such as Alzheimer's disease [24, 167] , cognitive state classification [17], fMRI pattern classification [25, 169], face recognition [26] and word prediction from fMRI data [27].

Fayyaz et al. [163] have used Multilayer Perceptron (MLP) neural network to evaluate training accuracy based on 40 voxels while classifying fMRI patterns for making reward-less decisions. Due to limited data, the algorithm is often stuck on local optima with very low fitness. The performance of backpropagation and test statistics is also compromised due to local gradient data [163]. When the training size is not large enough, ANN will have to face the problem of overfitting and reflect biases [172]. Therefore, there is a great need to use large data to avoid the problem of overfitting and to find the appropriate Artificial Intelligence algorithms that can be used in the field of cognitive neuroscience. The hybrid techniques have recently gained more attention in computational neuroscience due to their exceptional advantages in processing with complex data structures [22, 28]. The applicability of the Adaptive neuro fuzzy inference system (ANFIS) in the domain of cognitive neuroscience is yet to be explored as it is suitable for managing nonlinear and fuzzy problems. We have focused on two key benefits of neural networks and fuzzy systems; (i) they are good at handling vague data, (ii) a priori knowledge.

Keeping in view these precincts, this study has been carried out where we have applied eight standard architectural variations of ANNs including feedforward, recurrent, extreme learning machines, and regression. We have used Adaptive Neuro-Fuzzy Inference System (ANFIS) with Grid partitioning, Subtractive clustering and Fuzzy c-means clustering for training to find the ideal ANFIS on fMRI data. These techniques have gained more attention in computational neuroscience due to their exceptional advantages in processing with complex data structures [22, 28]. ANN and ANFIS architectural genres have compared to identify the spatiotemporal activation patterns and classifying decision-making voxels from the prefrontal cortex of the brain region.

For this study, we have extended the approach used by Fayyaz et al. [163] to a larger architecture. Moreover, data size and multiple statistical validation methods are used to further refine the performance and worth of the technique.

## 5.3. Baseline Techniques

The same baseline techniques have been used to solve this problem as described in Section 3.2.

## 5.4. Proposed Framework

This section sheds light on the functionality and architecture of the proposed scheme as shown in Figure 5.1 in detail. The methodology comprises of the following primary tasks: (1) Data for the model (2) Pre-processing (3) builds the model.

### 5.4.1 Data Description

The required fMRI data is publicly available at ("http://cnl.web.arizona.edu/spm.htm"). fMRI data of two major Brodmann areas (BAs) 10 and 47 with a total of 1200 voxels are collected. The three conditions: (i) hard-to-learn (ii) easy-to-learn and (iii) familiar birds as control condition, were visualized as a subject of fMRI experiments to the respondent. Bird images were presented to the participants and instructed to make a judgment among each type silently which criteria mentioned above. There were four blocks for control birds and two blocks each for easy and hard learned birds. The data presented four bird image stimuli each of 5 seconds and a total of 80 scans with a 2-second time repetition (TRs) were created. For this study, GE 1.5 Tesla signa 5x software is used to obtain the Echo Planar Image (EPI) data[163, 202]. The expectation is that the visual and prefrontal cortex areas of the brain are significantly active when the subject observed and responded to the bird's image.

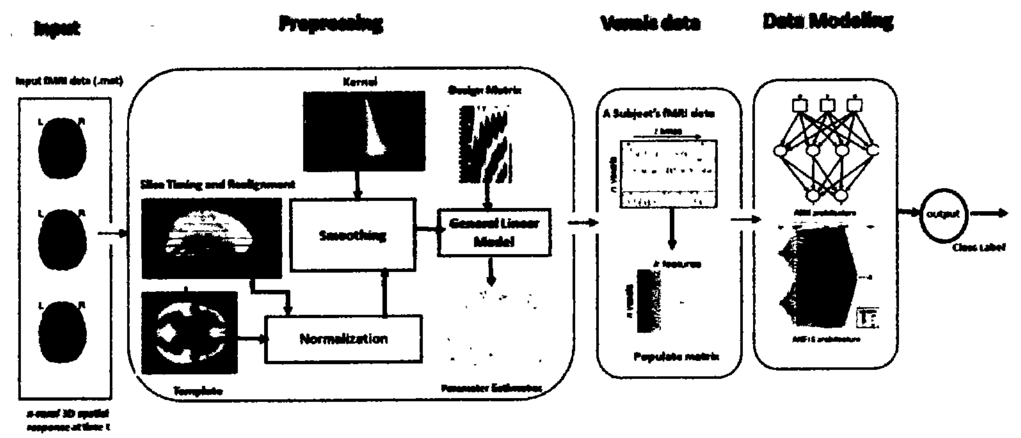



**Figure 5.1: System Architecture and Overview.**

### 5.4.2 Data Pre-Processing

Simulated fMRI data has been pre-processed[203] and analyzed by using the latest SPM package (SPM12; "http:// www.fil.ion.ucl.ac.uk/spm/"). It is recommended to perform pre-processing steps before initializing GLM analysis of human brain fMRI data to avoid errors. After pre-processing, fMRI activation was statistically analyzed in many steps by using the Gaussian kernel to reduce the noise, realigned for correction of head motion, the Montreal Neurological Institute (MNI) template for image normalization, and the slice timing to correct image differences. To eliminate the noise, a cut-off frequency of the 0.2 Hz and filter of the respiratory rate (0 - 0.2 Hz) was used. A standard low-pass frequency filter was used to remove the noise, such as respiratory and cardiac effects during GLM analysis [12, 163, 202]. After the compilation of the pre-processing steps, an analysis was performed to detect voxel activation in the fMRI brain data. The steps for analysis are model specification, estimation, and results.

All 80 images of block design are used for model specification. All three visual conditions specification were provided for this experiment. Also, specified the scan duration and the origin of these visual conditions (using the block design). In our experiment, TR was 2 seconds for fMRI data. Then, this research performed the analysis and specified the design pattern in the output. In the next step, we used the file "SPM.mat" to estimate the betas of each condition. In the resulting step, a t-test based contrast was performed to compare the beta images. We can establish multiple contrasts depending on the objectives. The procedure to describe the contrasts is that overall sum should be zero. For example, 1 indicates the most active condition, -1 indicates the least active condition and 0 indicates the ignorance of a condition. For the classification between easy and hard birds we choose the contrast value C [1 -2 1], which shows that easy and hard were more active than the control conditions. The threshold value has been determined and obtained brain activation during scanning. We chose RoI (Brodmann areas 10 and 47) to correctly identify voxels of decision making using the ANN and hybrid technique[163].

### 5.4.3 Data Modeling

The same architectures as described in Section 3.3.2.1, 3.3.2.2 and 3.3.2.3 were used in this work for data modeling.

#### 5.4.3.1. ANN Training and Learning in Environment

The key challenge for all ANN architectures is to keep a generalized error, also known as a test error, as small as possible. The factors that determine how well a machine learning algorithm works are its ability to make the training error small and to minimize the gap between training and testing errors. To train and validate the ANN, 50:50,55:45,60:40,65:35,70:30,75:25,80:20 training/testing ratios was used to get the lowest generalized error. The ANN was trained during epochs until achieved the minimum error against each division. The error is calculated as a mean squared error (MSE) between the actual and the desired output. In this work, weighted 1000 heuristics are arranged in search to find the best configuration of each architecture. Some other adjustable parameters are also set up along with their allowable or fixed values and are given in summarized form in Table 5.1.

**Table 5.1: Network Parameters.**

| No. | Architecture parameter | Value |
|---|---|---|
| 1 | No. of weights per net network configuration | 1-1000 |
| 2 | Hidden Layers | 1 |
| 3 | Training goal (MSE) | 0.01/fold |
| 4 | Training trails with weight re-initialization | 15 |
| 5 | Input vector used voxels of Brodmann areas 10 and 47 | 1200 |
| 6 | Output vector | [0,1] |

## 5.5. Experimental Design of Improved Diagnostic Accuracy in Dependent Personality Disorders

### 5.5.1 Performance Metrics

Among the various standard measures to represent an error, e.g. Mean Squared Error (MSE), Accuracy, Specificity and Sensitivity has used in this work. Following performance evaluation measures have used.

$$MSE = \frac{1}{n}\sum_{I=1}^{n}(Yi - \hat{Y}i)^2 \qquad (5.1)$$

$$Sensitivity \; \frac{number\ of\ true\ positives}{number\ of\ true\ positives\ +\ number\ of\ false\ negatives} \qquad (5.2)$$

$$Sepecificty = \frac{number\ of\ true\ negatives}{number\ of\ true\ negatives +\ number\ of\ false\ positives} \qquad (5.3)$$

$$Accuracy = \frac{number\ of\ true\ positives + number\ of\ true\ negative}{number\ of\ condition\ positives +\ number\ of\ condition\ negatives} \qquad (5.4)$$

To estimate the importance of each voxel, the Leave-one-out sequential method has constructed for this purpose as shown in Figure 5.2. The method is repeated N times (N is the total number of voxels, i.e. 1200) and in every iteration, each $i^{th}$ voxel (one voxel) is left out one time, and the remaining n-i voxels are fed to the classifier for training and testing. Then calculate the accuracy of remaining n-i voxels. The significance of each voxel is calculated by the difference between accuracy estimate with and without of the voxel. The high difference, the higher is the rank of the voxel. Although it is slightly conservative and computationally expensive estimate of the performance of each voxel, it is almost unbiased.

## 5.6. Results and Discussion

Voxels of Brodmann areas 10 and 47 were used as an independent set variable to analyze the selected techniques. As compared to Fayyaz et al. [163] a more substantial number of voxels were selected, i.e., 1200 (See Table 5.1) for reliable results. Different statistical validation methods, i.e., K fold cross validation (10 and 20-fold), bootstrapping and 70:30 training/testing were used to estimate the skills of machine learning models comprehensively. The first contribution of our experimental work is to find the best neural network model after comparing eight standard architectural variations of neural networks. The neuron value is usually between 1 and 15. Figure 5.3 presents the best configuration of each neural nets option with minimal mean square error (MSE) value. The results revealed that from all the applied variants of neural networks, the dominant neural network with considerably least error is 'Generalized regression neural network'.

This is because GRNN has many theoretical advantages over other ANNs. In contrast to standard feedforward networks, GRNN solves the convex optimization problem by always converging to a global solution and not being trapped by a local minimum which depends on the minimum samples and has a robust internal over-fitting mechanism [183].

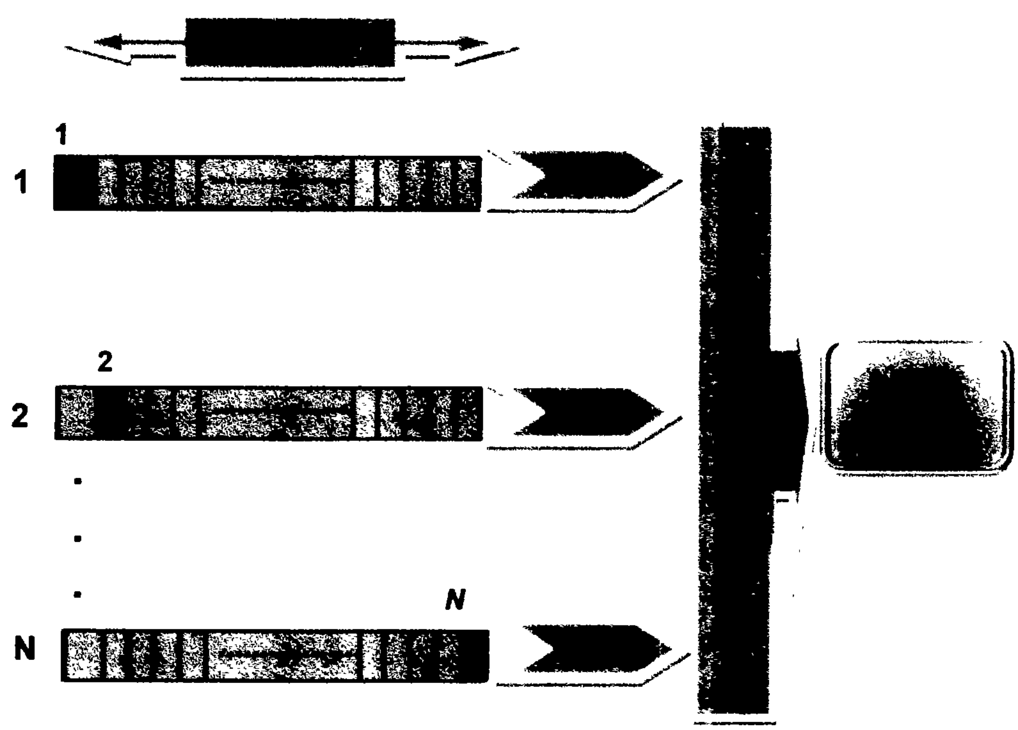



**Figure 5.2: Schematic view of leave-one-out sequential (LOOS) algorithm. Samples coloured as yellow are the training and testing sets and the red sample is the one that leaved out. PM: performance metric.**

Table 5.2 represents the high percentages of the accuracy of dominant ANN architecture, i.e., GRNN during training and testing. Table 5.3 represents the accuracy percentage of predictions of the model during testing with minimum MSE. GRNN accuracy is dominant in all other models with an accuracy of 95%. Remaining eight model's accuracy fall in the range of 65%.

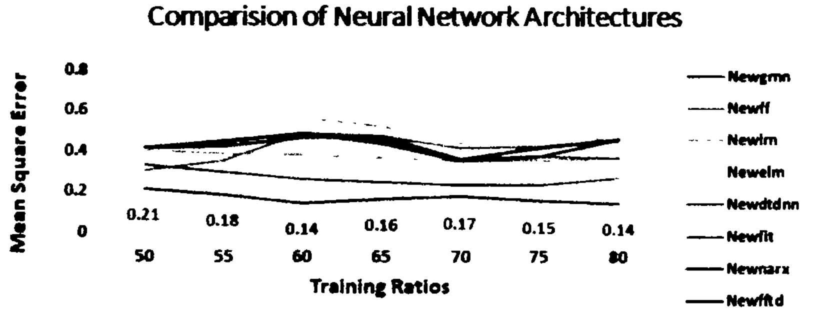



**Figure 5.3: Least Mean Square Error of Neural Network Architectures for the voxels of Brodmann areas 10 and 47 corresponding to easy and hard birds stimuli.**

GRNN= Generalized regression neural network; FF=Feedforward; LRN= Local response normalization; NARX= nonlinear autoregressive network with exogenous inputs; DTNN= Design time series distributed delay neural network; FFDT= Focused time delay neural network; ELM=Extreme learning machine; FIT=Function Fitting neural network

**Table 5.2: Percentage of incorrect predictions of Generalized Regression Neural Network model during training and testing**

| | | Sensitivity | Specificity | Accuracy |
|---|---|---|---|---|
| **Training** | Percent incorrect predictions | 100% | 100% | 100% |
| **Testing** | Percent incorrect predictions | 94% | 95% | 95% |

**Experiment 2** was conducted to find the best-suited model among the three, i.e. Genfis1, Genfis2, Genfis3 of the ANFIS for voxel classification. The results in Table 5.4 illustrate category wise average accuracy using three hybrid methods (Genfis1, Genfis2, Genfis3) and ANN method GRNN of fMRI data set.

For the model precision and classification of two visual conditions, a widely used performance measure ROC curve constructed for each learning algorithm.

Table 5.4 the general trend seen throughout all evaluation metrics is that the best performing Genfis3 (fuzzy c-means clustering) shows higher average accuracy than the Genfis2 (subtractive clustering) and Genfis1 (grid partitioning approach) respectively, for hybrid learning. For the model precision and classification of two visual conditions, a widely used performance measure ROC curve was constructed for each learning algorithm.

**Table 5.3: Accuracy of easy/hard Classification Artificial Neural Network Models during Testing**

| Neural Network Models | Testing error | Accuracy |
|---|---|---|
| **GRNN** | 0.001 | 95% |
| **FF** | 0.33 | 65% |
| **LRN** | 0.22 | 63% |
| **NARX** | 0.38 | 66% |
| **DTNN** | 0.24 | 64% |
| **FFDT** | 0.30 | 62% |
| **ELM** | 0.31 | 64% |
| **FIT** | 0.29 | 63% |

**Table 5.4: Results of All Models using all validations during Test.**

| | 70:30 | | | 10-Fold | | | 20-Fold | | | Bootstrapping | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Models** | **Avg Sensitivity %** | **Avg Specificity%** | **Avg Accuracy %** | **Avg Sensitivity %** | **Avg Specificity%** | **Avg Accuracy %** | **Avg Sensitivity %** | **Avg Specificity%** | **Avg Accuracy %** | **Avg Sensitivity %** | **Avg Specificity%** | **Avg Accuracy %** |
| **GRNN** | 94 | 95 | 95 | 99 | 98 | 90 | 86 | 90 | 87 | 78 | 79 | 79 |
| **Genfis1** | 94 | 96 | 94 | 96 | 98 | 90 | 88 | 92 | 88 | 82 | 84 | 85 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Genfis2** | 96 | 98 | 97 | 90 | 92 | 95 | 89 | 90 | 90 | 87 | 92 | 90 |
| **Genfis3** | 99 | 98 | 99 | 91 | 98 | 97 | 91 | 96 | 93 | 89 | 94 | 92 |

GRNN=Generalized regression neural network ; Genfis 1= Grid partitioning; Genfis2= Subtractive Clustering; Genfis3= Fuzzy c-means clustering;

To estimate the importance of each voxel, the Leave-one-out sequential method has constructed for this purpose. The method is repeated N times (N is the total number of voxels, i.e. 1200) and in every iteration, each $i^{th}$ voxel (one voxel) is left out one time, and the remaining n-i voxels are fed to the classifier for training and testing. Then calculate the accuracy of remaining n-i voxels. The significance of each voxel is calculated by the difference between accuracy estimate with and without of the voxel.

Figure 5.5 illustrates the importance of dominant voxels graphically with a significance of over 70%. We used MRIcron and MATLAB to determine the location of activated voxels with a visual representation of Brodmann areas 10 and 47 while they decided on the easy and hard situation in a reward-less task [163]. Figure 5.6 represents the axial view of the most dominated voxels, and Figure 5.7 represents the exact position of the decision-making voxels indicated by cross-hairs in the brain. To show the visual locations in 3D, we used a cut-out brain image as shown in Figure 5.8. It is also exciting to mention that 8 out of 22 identified voxels are common, as reported by Fayyaz et al.,[163]. It also validates the accuracy of our results. Common voxels referred to as bold in

Table 5.5 and yellow color in Figure 5.7 and Figure 5.8.

People with a dependent personality disorder (DPD) have a hard time making everyday decisions between two conditions without receiving excessive advice and reassurance from others. A dependent personality disorder is diagnosed in 0.5 to 0.6 percent of the total population and is most often diagnosed in adulthood. To diagnose a dependent personality disorder, there is no blood or genetic tests are available and are usually diagnosed by a trained psychologist. Several people with personality disorders usually seek treatment only when the disease significantly affects their lives.

Our findings are beneficial not only to diagnose early but also the treatment of DPD. A mental health expert can also determine whether the symptoms meet the criteria needed to diagnose the personality disorder by knowing the exact location of voxels. This can lead to finding the cause that is still unknown. As much as the voxels are activated, more the person will make his decisions quickly. In this study, voxels of the whole brain considered as population. Brodmann area 10 and 47 which are part of frontal and visual-frontal lobe [12] were selected as ROIs.

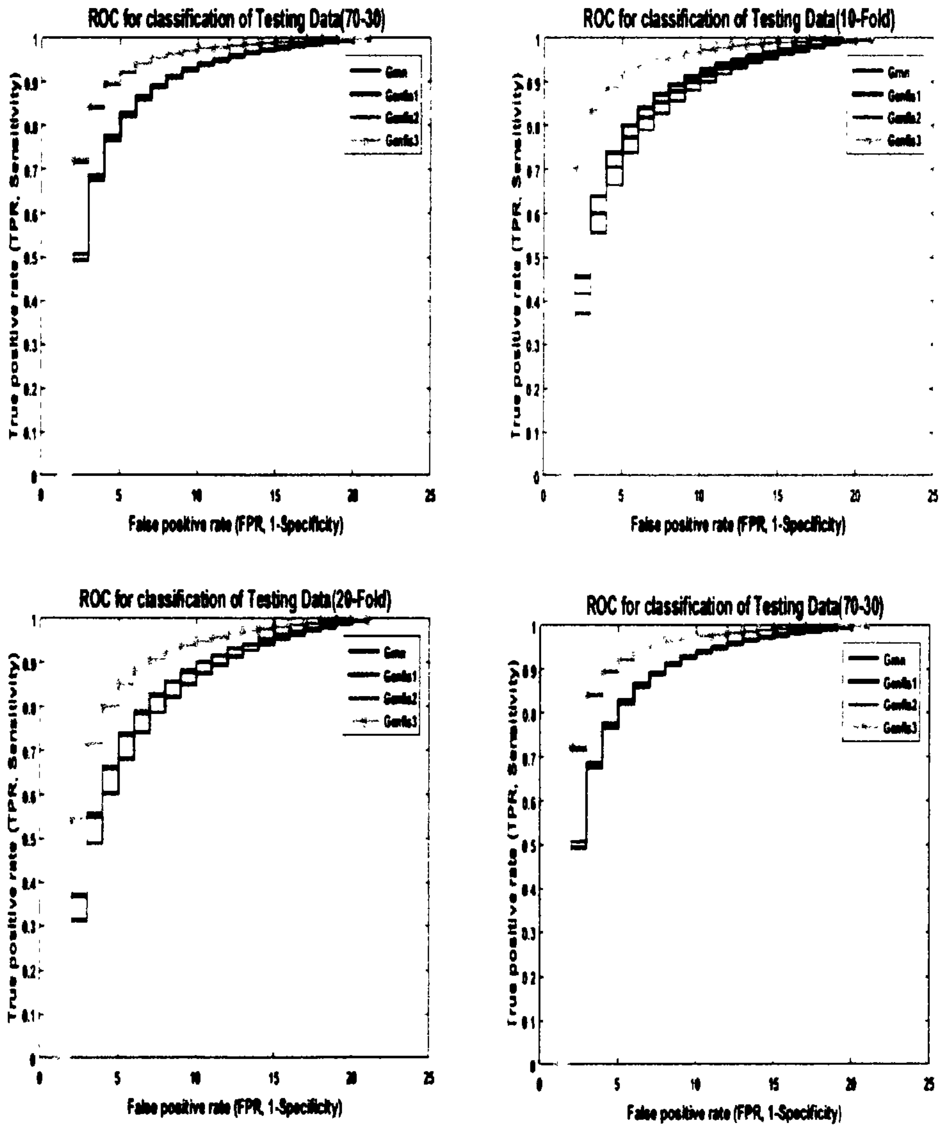



**Figure 5.4: Analysis for area under receiver operating characteristics curve by Generalized regression neural network (GRNN), Grid partitioning (Genfis1),**

**Subtractive clustering (Genfis2), Fuzzy C-Means Clustering (Genfis3) of all validation schemes.**

**Table 5.5: The locations of most dominated voxels with a significance of over 70%.**

| S.no. | Importance (%) | Location of Voxels in MNI Coordinates | | |
|---|---|---|---|---|
| | | X | Y | Z |
| **1** | 100 | **-28** | **28** | **-20** |
| **2** | 99 | -10 | 36 | -8 |
| **3** | 93 | -14 | 50 | -2 |
| **4** | 91 | **-30** | **30** | **-16** |
| **5** | 85 | **-28** | **28** | **-18** |
| **6** | 84 | **-16** | **50** | **-4** |
| **7** | 83 | -40 | 22 | 2 |
| **8** | 82 | -32 | 26 | 4 |
| **9** | 81 | -12 | 52 | 0 |
| **10** | 80 | **-38** | **24** | **2** |
| **11** | 79 | -46 | 20 | 2 |
| **12** | 75 | -54 | 18 | -2 |
| **13** | 75 | -48 | 36 | -2 |
| **14** | 75 | -10 | 52 | -4 |

| | | | | |
|---|---|---|---|---|
| **15** | 74 | -18 | 46 | -2 |
| **16** | 73 | **-36** | **22** | **0** |
| **17** | 71 | **-16** | **48** | **-4** |
| **18** | 71 | -52 | 20 | 2 |
| **19** | 71 | -54 | 26 | -6 |
| **20** | 71 | -14 | 52 | -4 |
| **21** | 70 | -56 | 28 | -2 |
| **22** | 70 | **-30** | **32** | **-18** |

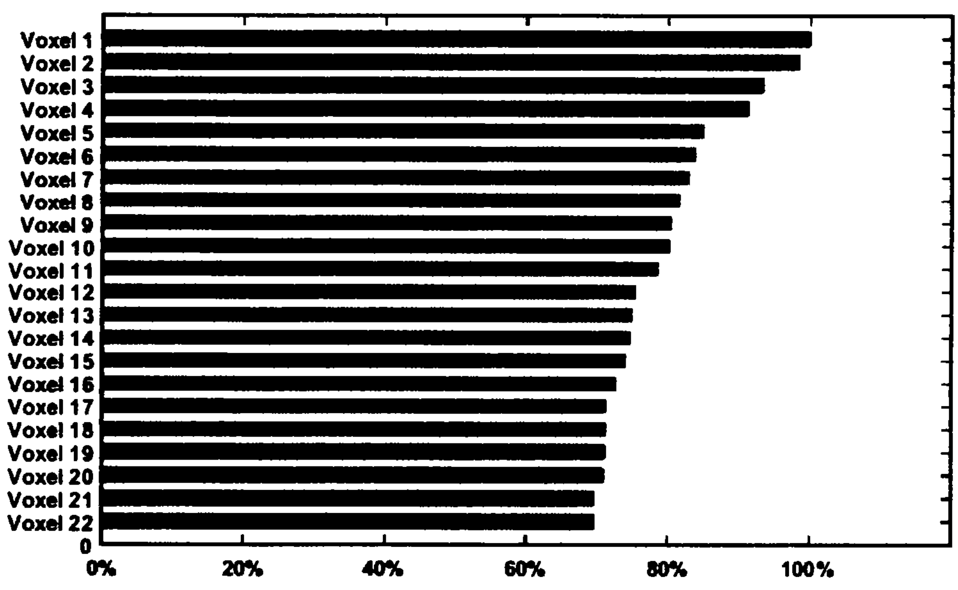



**Figure 5.5: Bar graph representing (%) importance of dominated voxels of two decisions (easy and hard birds).**

It has already reported in the literature that BAs 10 and 47 had a strong causal relationship with BA17[12]. The clusters of voxels of these selected BAs are used as independent variables.

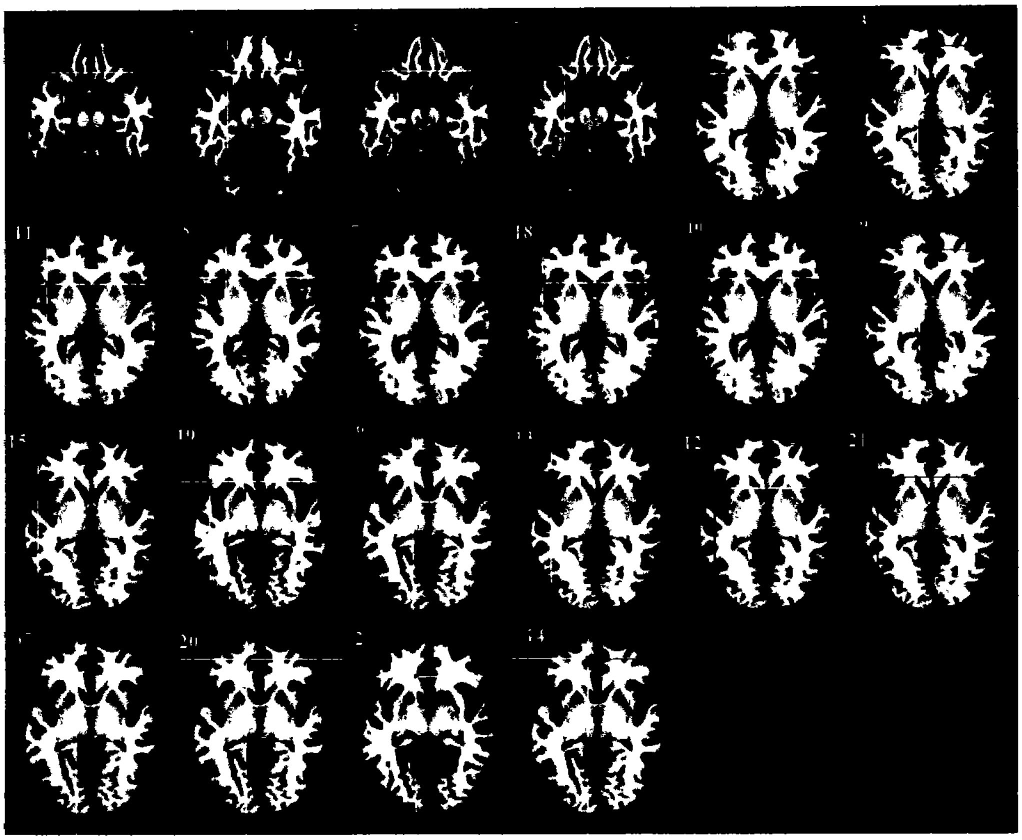


**Figure 5.6: The axial view of slices of decision-making voxels and crosshairs are representing their locations in brain.**

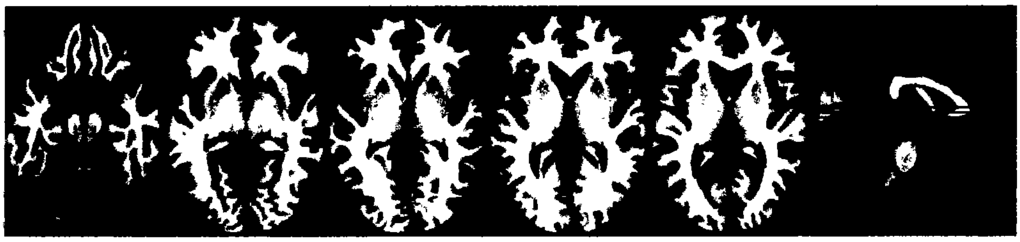


**Figure 5.7: The multi-slices axial view of most dominant voxels during decision making.**

**Figure 5.8: Render view displays location of dominated voxels in the prefrontal cortex.**

## 5.7. Chapter Summary

In this chapter, we reported a comparative study of the effectiveness of machine learning techniques in a hybrid way to diagnose dependent personality disorder in reward-less decision-making using fMRI data. This research focuses on the use of ANNs and hybrid adaptive learning algorithms for identification of features and classification of human brain voxels for decision making. We conclude that this was the activation associated with reward less decision making when given to visual tasks as they appear in the data. Because human brain function involves a very complex causal network, a little disturbance can lead to the development of acute pathologies. In such a case, a comparison of neural activation with a predetermined brain area for specific tasks may help to improve the diagnosis of such neurological pathologies. If the activation that is associated is well compared, the patient may free from disorders that affect the decision task. Medical treatment is suggested when abnormal activations in these specific brain areas are observed against these visual tasks. There are some encouraging results from this study. Particularly, Fuzzy C-means clustering that achieve promising higher accuracy of 95%. Among the eight different ANN standard architectures selected, only Generalized Regression neural networks (GRNN) trained model, achieved higher compatible accuracy 90% with the smallest variance. In general, all learning algorithms proved promising in obtaining accuracy with not an enormous difference for this application. To the best of our knowledge, only one publication has been published that focuses on reward-less relative decision making using a neural multilayer perceptron network with minimal training data size [163].

# Chapter 6: Conclusions and Future Work

## 6.1. Conclusions

This thesis study was inspired to address issues of imitating human-like behaviour and behavioural neuroscience to provide viable solutions in order to synthesize a more precise classification and prediction model. We describe and compare several artificial intelligence-based hybrid approaches that have not been fully exploited by the application of imitation learning and neuroscience behaviour. We also propose novel ways to pre-process the case base used by an imitative agent, in the Pac-man simulated game domain, offers a significant improvement in imitation ability compared to the case base that is not processed.

The remainder of this chapter contains a summary of the contributions and results of this work as well as the outline the limitations of research and future work areas.

### 6.1.1 Imitation of Human-Like Behaviour

Learning from the demonstration in video games and robots is a domain stimulated by the phenomenon of imitation in humans. LFD, which has a relatively short history as a field of research in artificial intelligence and games, has made significant progress and therefore has many challenges and problems to overcome. Some of them have addressed this thesis through some contributions, thus promoting progress in this area. For learning in practical applications, individual AI tools are not sufficient for LFD. Artificial intelligence tools have been successful in LfD using a meticulous network combination of these tools, performed by experts, in a domain-specific way. Unfortunately, no standard architecture does this at the general level, that is, an architecture that can learn most, if not all, human behaviours. This shows that these tools are not theoretically geared to LfD in their abilities. Therefore, a new framework for LFD is needed that provides a solution of a general nature. The best way to create or discover such a framework is probably to lay down first the theoretical basis from science or, more precisely, of imitation science and proposed a suitable computational model that corresponds to its scientific basis. Evaluation in robot LfD is inherently a difficult task. The intuitive answer to the question of how good a specific learning outcome is, comes as the difference between the learned and demonstrated behaviour. To find out how good a specific learning outcome is, calculated as the difference between the learned and the proven behaviour but this answer is unjustifiable in the context of LfD. Are we saying that the imitative learning software agent can reach the goal of the demonstrated action (s) 100% when tested, would that be perfect learning? In reality, this may not be the case. Any conclusion based solely on goal achievement is of little or no value in the context of LfD. Also, when the imitative learning software agent can fully demonstrate the exact trajectories of the action (s) demonstrated, this does not represent perfect learning either. Consequently, it is exceedingly challenging to define a criterion or an objective measure. One thing is certain; however, this criterion will weigh most heavily on the role of the demonstrated, intention and trajectory. It is not easy to determine the exact behavior, and it can be expensive to force the demonstrated to show behavior in all kinds of test situations. To solve this problem, it is proposed to test a proposal to simulate the LfD based human behaviour on a simple level, allowing for minimal human interpretation and subjectivity.

Firstly, we focused on evaluating the behavioral similarity between human gameplay and AI agents trained on recorded data of human players. The proposal is to use the standard algorithms from the popular domain of neural networks and hybrid learning algorithms. Eight representative standard network architectural variations of ANNs were used to simulate the human player behavior. This work has also been significantly expanded to include hybrid systems, i.e., Adaptive neuro-fuzzy inference system (ANFIS) for human behavior modeling issues in digital games. These methods are the first to be implemented to imitate human-like behavior. The architectures of ANNs from feed-forward, recurrent, extreme learning machines and computational regression classes are selected for this purpose. On the other hand, to find the ideal ANFIS system, we trained the system with Grid partitioning, Subtractive clustering, and Fuzzy C-Means clustering methods. In our work, we used the Pac-man game as a testbed to compare these techniques. Using these methods, we have evaluated the similarity of best-performing agents. GRNN (Generalized regression neural network) surpasses eight neural network architectures by showing the best results with the smallest variance and average accuracy of 89% for both winning and losing behavior. The increasing number of training and testing demonstrations does not affect the GRNN performance, so it remains consistent. These results, achieved with more methodological rigor and trends, also confirm the same as in [4, 84, 204] but in a different domain. Among the trained agents for ANFIS, those trained with subtractive clustering method to generate FIS to model the human player behavior performed much better than those trained with Fuzzy C-Means and grid partitioning. In general, subtractive clustering performance is overall best with an average accuracy of 94% along with Fuzzy C-Means followed closely with an average accuracy of 87%. In other words, we can say that two-hybrid methods outperformed the direct imitation method regarding the accuracy score. All the agents are trained performed the best. This might be due to the nature of the application because there is no unseen scenario in the Pac-man maze compared to some other games. These results may or may not generalize to other domains and other controlling architectures. Two automatic statistical methods are used to determine the believability of our agents to validate the results generated. Both methods validate the results of this study with high accuracy. The decisive victory of the ANFIS hybrid intelligence system based on subtractive clusters over all other agent architectures shows that hybrid training methods best solve the problem for the considered range of methods. In general, Subtractive clustering *(Genfis2)* declared the most human-like AI controller with both automatic methods that have 90% similarity scores with Mann-Whitney U Test and 99% with Cosine Similarity Analysis.

### 6.1.2 Evaluation of Case Knowledge Extraction Techniques

In chapter 4, the research question "How to Improve Case-based Agent Imitation in Simulation Gaming Environments by Pre-processing the Case Base" was explored by examining the removal of certain features of the cases, creating prototype cases, and removing similar cases. To take advantage of experience, a lazy learning technique such as Case-based reasoning (CBR), has also used to imitate human-like behaviour while playing games with an aim in reducing the size of case base without compromising the imitative performance of the imitation agent. Because all techniques are developed in the context of CBR systems, we first discussed the assumptions of similarity in CBR that are vital in the retrieval of cases. It is best to retrieve a set of cases very similar to the problem in question when retrieving cases.

To derive a measure for similarity function for retrieval of nearest neighbours from the likelihood ratios which gives the best results, we described the retrieval strategies and solution algorithm by highlighting the effect of seven (7) different similarity measures. **Feature and Case Removal:** Specific techniques, such as Rough sets and FP-based algorithms are developed to deal with the process of reducing features and cases and learning the similarity between features. Reducing irrelevant features without compromising the ability to distinguish cases in different equivalence classes is one of the significant advantages of using coarse sets. On the contrary, high computational complexity is the major drawback of conventional rough set-based methods. For example, if there are $n$ case and m features, $O(n^2 * m)$ computations are required. In this thesis, the computational complexity has been reduced to be linear with the number of cases and features.

A novel FP- based approach is proposed by developing an ordered case base using L (1) frequent patterns in alphabetic preferences. And then compare L(N) frequent patterns with ordered case bases that match L(N) item sets. The matched cases replaced with a single case and then updated the original case base. The results of the experiments show that the FP-based algorithm for case selection is the most promising algorithm with the highest accuracy and least storage requirements. Higher prediction accuracy and less storage space requirement were obtained with each algorithm when compared to using the original case base.

In general, the results lead to the following observations: 1) the two approaches proposed have been able to reduce cases substantially. 2) The prediction accuracy is improved or even preserve when applying both approaches. The FP-based approach remains consistent to achieve greater accuracy for all reduced feature sets and with reduced cases. The accuracy achieved by both approaches is nearly the same for selecting the final reduced case base. The FP-based has a stronger ability to remove redundant cases than a rough set-based approach.

### 6.1.3 Improved Diagnostic Accuracy in Dependent Personality Disorders on fMRI Data

We reported a comparative study of the effectiveness of machine learning techniques in a hybrid way to diagnose a dependent personality disorder in reward-less decision-making using fMRI data. This research focuses on the use of ANNs and hybrid adaptive learning algorithms for identification of features and classification of human brain voxels for decision making. We conclude that this was the activation associated with reward less decision making when given to visual tasks as they appear in the data. Because human brain function involves a very complex causal network, a little disturbance can lead to the development of acute pathologies. In such a case, a comparison of neural activation with a predetermined brain area for specific tasks may help to improve the diagnosis of such neurological pathologies. If the activation that is associated is well compared, the patient may free from disorders that affect the decision task. Medical treatment is suggested when abnormal activations in these specific brain areas are observed against these visual tasks. There are some encouraging results from this study. Particularly, Fuzzy C-means clustering that achieves promising higher accuracy of 95%. Among the eight different ANN standard architectures selected, only Generalized Regression neural networks (GRNN) trained model, achieved higher compatible accuracy 90% with the smallest variance. In general, all learning algorithms proved promising in obtaining accuracy with not an enormous difference for this application. To the best of our knowledge, only one publication has been published that focuses on reward-less relative decision making using a neural multilayer perceptron network with minimal training data size [163].

## 6.2. Future Directions

There are many prospects for advancing this work. In the following, our future research is briefly outlined.

### 6.2.1 Evaluation extensions are as follows.

Need to develop a more sophisticated evaluation strategy to thoroughly evaluate the overall framework and theoretical models of calculation. To develop formulas and assessment methods to objectively measure the degree of imitation between goal achievement and trajectory tracking, especially in invisible situations. This would likely involve the construction of supplementary metrics that are more appropriate for the objective of this exercise.

### 6.2.2 Application extensions are as follows:

We believe that these promising results and proposed methodologies can be used to gain insight into human data in any paradigm even though the current work involves a collection of data obtained from game-based test beds. Some important directions in the future will need to be mentioned to complete this preliminary study.

First, all the Evolutionary Algorithms (EAs) algorithms in AI used for the behavioral modeling in all aspects are entirely based on Darwinian law of survival of the fittest. There are other established theories of evolution by equally credible, but less famous philosophers. These alternatives deserve our attention. If only because some of them claim to correct the oversights in Darwin's theory. Lamarck and Peirce propose a triadic conception of the sign. This can be used to design new evolutionary collaborative approaches which do not comprise of a closed system and are more generic. The experimental observations of classical model EA and their variants performed up till now showed they have weak theoretical foundation. Darwin's theory in computer science, lead diagrammatically to stagnation at very stage, unless artificially deviated. This unpredicted behaviour leads to premature convergence in optimization terms. An in-depth investigation reveals that inference methods used by all EAs algorithms in the domain of behavioral modeling are either binary or fuzzy but not triadic.

It is possible that the problems of this study may not arise in a new evolutionary approach. Secondly, these techniques need to be verified on applications that have more actions and respective demonstrations, movement in higher dimensions with large number of controls. Other possible and interesting enhancement may be made by implementing the concept of majority voting to improve the accuracy. This means that all the best performance controllers are implemented independently and the decision for the next move will be decided by the majority vote of trained controllers.

he most important line of work for the future is to create a more reliable and powerful imitation method, that require less data, so that models of a particular player's style of play can be learned quickly. Although the current trend of AI methods in games is towards playing games, generating content and modeling players, but still there are unexplored fora found in the literature. Game developers and testers need desperate AI assistance for bug tracking, balancing player behavior and experience and helping to prioritization of problems to fix. The emergence of online multi-player games on social media not only allow people to collaborate, but is also used to threaten or abuse other players and sexual harassment in game chats. There are some game developers who attempt to implement machine learning to curate such toxic behaviour in the MOBA League of Legends and discover sexual predators in game chats [205, 206].

Although this study has improved the ability of an agent to imitate human behaviour, by preprocessing the case base used by the imitative agent, many areas can still be explored. Future possible research topics concern both how to overcome the limitations of this work and to address topics outside the scope of this thesis. Some notable future work areas are mentioned as:

- **Adaptation Ability:** In order to overcome the limitations of the case-based knowledge extraction techniques mentioned in this research we have considered the three criteria storage requirements, prediction accuracy, and problem-solving efficiency. In our current research, we have not studied criteria important for assessing the problem-solving quality of CBR systems, i.e., the ability to adopt solutions because of its domain-dependent characteristics. This will complete the entire research effort to build efficient and effective CBR systems.
- **Other Preprocessing techniques:** This research sought pre-processing techniques to improve the imitation capability of a case-based imitative agent. The algorithms developed were used in the selected domain, but not necessarily with the highest performance or highest computational efficiency. It would go beyond the scope of this work to make an exhaustive comparison of State-of-the-art feature/case selection algorithms on the data.
- **Other Domains:** In the future work, to test the domain independence of the

software framework, it will also imitate agents in other simulated environments, imitating physical robots, and imitating the behavior of other agents. This will not only test the domain independence of the software framework, but also open up new challenges such as computer vision and action recognition.

# References


[1] G. N. Yannakakis and M. Maragoudakis, "Player modeling impact on player's entertainment in computer games," *Lecture notes in computer science,* vol. 3538, p. 74, 2005.

[2] J. Sum, C.-S. Leung, G. H. Young, and W.-K. Kan, "On the Kalman filtering method in neural network training and pruning," *IEEE Transactions on Neural Networks,* vol. 10, pp. 161-166, 1999.

[3] J. Togelius and G. N. Yannakakis, "General general game AI," in *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, 2016, pp. 1-8.

[4] J. Ortega, N. Shaker, J. Togelius, and G. N. Yannakakis, "Imitating human playing styles in super mario bros," *Entertainment Computing,* vol. 4, pp. 93-104, 2013.

[5] N. A. Taatgen, M. van Oploo, J. Braaksma, and J. Niemantsverdriet, "How to construct a believable opponent using cognitive modeling in the game of set," in *Proceedings of the fifth international conference on cognitive modeling,* 2003, pp. 201-206.

[6] J. Stewart, "Different Approaches to Implementing Player Centered Game Design: A Comparison Study."

[7] B. Medler, "Using recommendation systems to adapt gameplay," in *Discoveries in gaming and computer-mediated simulations: New interdisciplinary applications*, ed: IGI Global, 2011, pp. 64-77.

[8] M. U. Suleman and M. M. Awais, "Learning from demonstration in robots: Experimental comparison of neural architectures," *Robotics and Computer-Integrated Manufacturing,* vol. 27, pp. 794-801, 2011.

[9] M. Miranda, A. A. Sánchez-Ruiz, and F. Peinado, "Pac-man or Pac-Bot? Exploring subjective perception of players' humanity in Ms. Pac-man," in *CoSECivi*, 2017, pp. 163-175.

[10] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, *et al.*, "Procedural Content Generation via Machine Learning (PCGML)," *arXiv preprint arXiv:1702.00539,* 2017.

[11] A. H. Marblestone, G. Wayne, and K. P. Kording, "Toward an integration of deep learning and neuroscience," *Frontiers in computational neuroscience,* vol. 10, p. 94, 2016.

[12] F. Ahmad, N. Lee, E. Kim, S. H. Kim, and H. Park, "A shrinkage method for causal network detection of brain regions," *International Journal of Imaging Systems and Technology,* vol. 23, pp. 140-146, 2013.

[13] O. Sporns, "Analysis and function of large-scale brain networks," *Washington, DC: Society for Neuroscience,* 2010.

[14] K. J. Friston, P. Jezzard, and R. Turner, "Analysis of functional MRI time-series," *Human brain mapping,* vol. 1, pp. 153-171, 1994.

[15] A. R. McIntosh and M. Korostil, "Interpretation of neuroimaging data based on network concepts," *Brain Imaging and Behavior,* vol. 2, p. 264, 2008.

[16] L. A. Cacha and R. R. Poznanski, "Associable representations as field of influence for dynamic cognitive processes," *Journal of integrative neuroscience,* vol. 10, pp. 423-437, 2011.

[17] L. Cacha, S. Parida, S. Dehuri, S.-B. Cho, and R. Poznanski, "A fuzzy integral method based on the ensemble of neural networks to analyze fMRI data for cognitive state classification across multiple subjects," *Journal of integrative neuroscience,* vol. 15, pp. 593-606, 2016.

[18] G. Zhang, Y. Cheng, W. Shen, B. Liu, L. Huang, and S. Xie, "Brain Regional Homogeneity Changes in Cirrhotic Patients with or without Hepatic Encephalopathy Revealed by Multi-Frequency Bands Analysis Based on Resting-State Functional MRI," *Korean journal of radiology,* vol. 19, pp. 452-462, 2018.

[19] E. Belilovsky, K. Gkirtzou, M. Misyrlis, A. B. Konova, J. Honorio, N. Alia-Klein, *et al.*, "Predictive sparse modeling of fMRI data for improved classification, regression, and visualization using the k-support norm," *Computerized Medical Imaging and Graphics,* vol. 46, pp. 40-46, 2015.

[20] B. Jin, A. Strasburger, S. J. Laken, F. A. Kozel, K. A. Johnson, M. S. George, *et al.*, "Feature selection for fMRI-based deception detection," in *BMC bioinformatics*, 2009, p. S15.

[21] J. Honorio, D. Tomasi, R. Z. Goldstein, H.-C. Leung, and D. Samaras, "Can a single brain region predict a disorder?," *IEEE Transactions on Medical Imaging,* vol. 31, pp. 2062-2072, 2012.

[22] S. Parida, S. Dehuri, S.-B. Cho, L. Cacha, and R. Poznanski, "A hybrid method for classifying cognitive states from fMRI data," *Journal of integrative neuroscience,* vol. 14, pp. 355-368, 2015.

[23] L. I. Kuncheva and J. J. Rodríguez, "Classifier ensembles for fMRI data analysis: an experiment," *Magnetic resonance imaging,* vol. 28, pp. 583-593, 2010.

[24] F. Coppedè, E. Grossi, M. Buscema, and L. Migliore, "Application of artificial neural networks to investigate one-carbon metabolism in Alzheimer's disease and healthy matched individuals," *PLoS One,* vol. 8, p. e74012, 2013.

[25] J.-G. Lee, S. Jun, Y.-W. Cho, H. Lee, G. B. Kim, J. B. Seo, *et al.*, "Deep learning in medical imaging: general overview," *Korean journal of radiology,* vol. 18, pp. 570-584, 2017.

[26] Y. Zhao, Z. Zhen, X. Liu, Y. Song, and J. Liu, "The neural network for face recognition: Insights from an fMRI study on developmental prosopagnosia," *NeuroImage,* vol. 169, pp. 151-161, 2018.

[27] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, *et al.*, "Predicting human brain activity associated with the meanings of nouns," *science,* vol. 320, pp. 1191-1195, 2008.

[28] T. Helmy, M. M. Al-Harthi, and M. T. Faheem, "Adaptive ensemble and hybrid models for classification of bioinformatics datasets," *Trans Fuzzy Neural Netw Bioinform Glob J Technol Optim,* vol. 3, pp. 20-29, 2012.

[29] H. D. Navone, P. M. Granitto, P. F. Verdes, and H. A. Ceccatto, "A learning algorithm for neural network ensembles," *Inteligencia artificial. Revista Iberoamericana de Inteligencia Artificial,* vol. 5, 2001.

[30] G. Klir and B. Yuan, *Fuzzy sets and fuzzy logic* vol. 4: Prentice hall New Jersey, 1995.

[31] R. Brooks, D. Hassabis, D. Bray, and A. Shashua, "Is the brain a good model for machine intelligence?," *Nature,* vol. 482, p. 462, 2012.

[32] C. Thurau, C. Bauckhage, and G. Sagerer, "Learning human-like movement behavior for computer games," in *Proc. Int. Conf. on the Simulation of Adaptive Behavior*, 2004, pp. 315-323.

[33] G. N. Yannakakis and J. Hallam, "Evolving opponents for interesting interactive computer games," *From animals to animats,* vol. 8, pp. 499-508, 2004.

[34] J. Schell, *The Art of Game Design: A book of lenses*: AK Peters/CRC Press, 2014.

[35] G. N. Yannakakis and J. Togelius, "A panorama of artificial and computational intelligence in games," *IEEE Transactions on Computational Intelligence and AI in Games,* vol. 7, pp. 317-335, 2015.

[36] S. M. Lucas and G. Kendall, "Evolutionary computation and games," *IEEE Computational Intelligence Magazine,* vol. 1, pp. 10-18, 2006.

[37] N. Kirby, "Getting around the limits of machine learning," *AI game programming wisdom,* vol. 2, pp. 603-611, 2004.

[38] R. Miikkulainen, B. D. Bryant, R. Cornelius, I. V. Karpov, K. O. Stanley, and C. H. Yong, "Computational intelligence in games," *Computational Intelligence: Principles and Practice,* pp. 155-191, 2006.

[39] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, "Real-time neuroevolution in the NERO video game," *IEEE transactions on evolutionary computation,* vol. 9, pp. 653-668, 2005.

[40] S. Woodcock, J. Laird, and D. Pottinger, "Game AI: The state of the industry," *Game Developer Magazine,* vol. 8, pp. 7-8, 2000.

[41] J. Togelius, P. Burrow, and S. M. Lucas, "Multi-population competitive co-evolution of car racing controllers," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on,* 2007, pp. 4043-4050.

[42] C. Baekkelund, "A brief comparison of machine learning methods," *AI game programming wisdom,* vol. 3, pp. 617-631, 2006.

[43] M. van Lent and J. Laird, "Developing an artificial intelligence engine," *Ann Arbor,* vol. 1001, pp. 48109-2110, 1998.

[44] M. W. Bell, "Toward a definition of “virtual worlds”," *Journal For Virtual Worlds Research,* vol. 1, 2008.

[45] M. J. Khan, M. M. Awais, and S. Shamail, "Achieving self-configuration capability in autonomic systems using case-based reasoning with a new similarity measure," in *International Conference on Intelligent Computing*, 2007, pp. 97-106.

[46] D. Friedman, A. Steed, and M. Slater, "Spatial social behavior in second life," in *International Workshop on Intelligent Virtual Agents*, 2007, pp. 252-263.

[47] G. Robertson and I. Watson, "A review of real-time strategy game AI," *AI Magazine,* vol. 35, pp. 75-104, 2014.

[48] G. N. Yannakakis and J. Hallam, "Towards optimizing entertainment in computer games," *Applied Artificial Intelligence,* vol. 21, pp. 933-971, 2007.

[49] D. Choi, T. Könik, N. Nejati, C. Park, and P. Langley, "A Believable Agent for First-Person Shooter Games," in *AIIDE*, 2007, pp. 71-73.

[50] A. Liapis, G. N. Yannakakis, and J. Togelius, "Adapting models of visual aesthetics for personalized content creation," *IEEE Transactions on Computational Intelligence and AI in Games,* vol. 4, pp. 213-228, 2012.

[51] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, *et al.*, "Population based training of neural networks," *arXiv preprint arXiv:1711.09846,* 2017.

[52] D. Charles and M. Black, "Dynamic player modeling: A framework for player-centered digital games," in *Proc. of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, 2004, pp. 29-35.

[53] D. Charles, A. Kerr, M. McNeill, M. McAlister, M. Black, J. Kcklich, *et al.*, "Player-centred game design: Player modelling and adaptive digital games," in *Proceedings of the digital games research conference*, 2005, p. 00100.

[54] M. O. Riedl, "Scalable personalization of interactive experiences through creative automation," *Computers in Entertainment (CIE),* vol. 8, p. 26, 2010.

[55] P. Molyneux, "The future of game AI-Lecture," *Imperial College London, London, UK,* 2006.

[56] M. Wittkamp, L. Barone, and P. Hingston, "Using NEAT for continuous adaptation and teamwork formation in Pacman," in *Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On*, 2008, pp. 234-242.

[57] J. Orkin, "Three states and a plan: the AI of FEAR," in *Game Developers Conference*, 2006, p. 4.

[58] J. Togelius, G. N. Yannakakis, S. Karakovskiy, and N. Shaker, "Assessing believability," in *Believable bots*, ed: Springer, 2013, pp. 215-230.

[59] S. Parsons and M. Wooldridge, "Game theory and decision theory in multi-agent systems," *Autonomous Agents and Multi-Agent Systems,* vol. 5, pp. 243-254, 2002.

[60] Y. Murakami, T. Ishida, T. Kawasoe, and R. Hishiyama, "Scenario description for multi-agent simulation," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003, pp. 369-376.

[61] M. Booth, "The official counter-strike bot," in *Game Developers Conference*, 2004.

[62] D. Isla, "Managing complexity in the Halo 2 AI system," in *Proceedings of the Game Developers Conference*, 2005.

[63] R. Pillosu, "Coordinating agents with behavior trees," in *Paris Game AI Conference*, 2009.

[64] A. J. Champandard, *AI game development: Synthetic creatures with learning and reactive behaviors*: New Riders, 2003.

[65] M. Miranda, A. A. Sánchez-Ruiz, and F. Peinado, "Pac-man or Pac-Bot? Exploring Subjective Perception of Players' Humanity in Ms. Pac-man."

[66] G. Lee, M. Luo, F. Zambetta, and X. Li, "Learning a super mario controller from examples of human play," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 2014, pp. 1-8.

[67] C. Holmgård, A. Liapis, J. Togelius, and G. N. Yannakakis, "Personas versus clones for player decision modeling," in *International Conference on Entertainment Computing*, 2014, pp. 159-166.

[68] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development,* vol. 44, pp. 206-226, 2000.

[69] G. Synnaeve, N. Nardelli, A. Auvolat, S. Chintala, T. Lacroix, Z. Lin, *et al.*, "TorchCraft: a Library for Machine Learning Research on Real-Time Strategy Games," *arXiv preprint arXiv:1611.00625,* 2016.

[70] C. Hecker, "My liner notes for spore/spore behavior tree docs," *Chris Hecker's Website. Retrieved July from* http://www. *chrishecker. com/My_Liner_Notes_for_Spore/Spore_Behavior_Tree_Docs,* 2009.

[71] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of electronic imaging,* vol. 16, p. 049901, 2007.

[72] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation,* vol. 1, pp. 67-82, 1997.

[73] C. L. Moriarty and A. J. Gonzalez, "Learning Human Behavior from Observation for Gaming Applications," in *FLAIRS Conference*, 2009.

[74] J. Sum, W.-K. Kan, and G. H. Young, "A Note on the Equivalence of NARX and RNN," *Neural computing & applications,* vol. 8, pp. 33-39, 1999.

[75] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature,* vol. 529, pp. 484-489, 2016.

[76] S. Haykin, *Neural networks: a comprehensive foundation*: Prentice Hall PTR, 1994.

[77] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science1985.

[78] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE transactions on Neural Networks,* vol. 5, pp. 989-993, 1994.

[79] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics,* vol. 43, pp. 59-69, 1982.

[80] G. N. Yannakakis, J. Levine, J. Hallam, and M. Papageorgiou, "Performance, robustness and effort cost comparison of machine learning mechanisms in FlatLand," in *Proceedings of the 11th Mediterranean Conference on Control and Automation MED'03*, 2003.

[81] B. Mac Namee and P. Cunningham, "A Proposal for an Agent Architecture for Proactive Persistent Non Player Characters," Trinity College Dublin, Department of Computer Science2001.

[82] B. Geisler, "Integrated machine learning for behavior modeling in video games," in *Challenges in game artificial intelligence: papers from the 2004 AAAI workshop. AAAI Press, Menlo Park*, 2004, pp. 54-62.

[83] B. Chaperot and C. Fyfe, "Improving Artificial Intelligence In a Motocross Game," in *CIG*, 2006, pp. 181-186.

[84] J. Togelius, R. De Nardi, and S. M. Lucas, "Towards automatic personalised content creation for racing games," in *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, 2007, pp. 252-259.

[85] P. Spronck, M. Ponsen, I. Sprinkhuizen-Kuyper, and E. Postma, "Adaptive game AI with dynamic scripting," *Machine Learning*, vol. 63, pp. 217-248, 2006.

[86] M. Ponsen and P. Spronck, "Improving adaptive game AI with evolutionary learning," Citeseer, 2004.

[87] M. Gallagher and A. Ryan, "Learning to play Pac-man: An evolutionary, rule-based approach," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, 2003, pp. 2462-2469.

[88] S. Baluja, "Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning," Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science1994.

[89] G. B. Parker, T. S. Doherty, and M. Parker, "Evolution and prioritization of survival strategies for a simulated robot in Xpilot," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, 2005, pp. 2410-2415.

[90] G. B. Parker and M. Parker, "Evolving parameters for Xpilot combat agents," in *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, 2007, pp. 238-243.

[91] P. Demasi and J. d. O. Adriano, "On-line coevolution for action games," *International Journal of Intelligent Games & Simulation*, vol. 2, 2003.

[92] G. N. Yannakakis, R. Cowie, and C. Busso, "The Ordinal Nature of Emotions," in *Int. Conference on Affective Computing and Intelligent Interaction*, 2017.

[93] R. S. d Baker, G. R. Moore, A. Z. Wagner, J. Kalka, A. Salvi, M. Karabinos, *et al.*, "The dynamics between student affect and behavior occurring outside of

educational software," in *International Conference on Affective Computing and Intelligent Interaction*, 2011, pp. 14-24.

[94] A. Kleinsmith and N. Bianchi-Berthouze, "Form as a cue in the automatic recognition of non-acted affective body expressions," in *International Conference on Affective Computing and Intelligent Interaction*, 2011, pp. 155-164.

[95] H. P. Martinez, G. N. Yannakakis, and J. Hallam, "Don't classify ratings of affect; rank them!," *IEEE transactions on affective computing,* vol. 5, pp. 314-326, 2014.

[96] H. P. Martinez, Y. Bengio, and G. N. Yannakakis, "Learning deep physiological models of affect," *IEEE Computational Intelligence Magazine,* vol. 8, pp. 20-33, 2013.

[97] K. Yamamoto, S. Mizuno, C. Y. Chu, and R. Thawonmas, "Deduction of fighting-game countermeasures using the k-nearest neighbor algorithm and a game simulator," in *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, 2014, pp. 1-5.

[98] M. Garbarino, S. Tognetti, M. Matteucci, and A. Bonarini, "Learning general preference models from physiological responses in video games: How complex is it?," in *International Conference on Affective Computing and Intelligent Interaction*, 2011, pp. 517-526.

[99] I. Erev and A. E. Roth, "Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria," *American economic review,* pp. 848-881, 1998.

[100] B. Tastan and G. R. Sukthankar, "Learning Policies for First Person Shooter Games Using Inverse Reinforcement Learning," in *AIIDE,* 2011.

[101] F. Liberatore, A. M. Mora, P. A. Castillo, and J. J. Merelo, "Comparing heterogeneous and homogeneous flocking strategies for the ghost team in the game of Ms. Pac-man," *IEEE Transactions on Computational Intelligence and AI in Games,* vol. 8, pp. 278-287, 2016.

[102] A. J. Champandard, "Getting started with decision making and control systems," *AI Game Programming Wisdom,* vol. 4, pp. 257-264, 2008.

[103] K. Dill, "Introducing GAIA: A Reusable, Extensible architecture for AI behavior," in *Proceedings of the 2012 Spring Simulation Interoperability Workshop*, 2012.

[104] B. Schwab, D. Mark, K. Dill, M. Lewis, and R. Evans, "GDC: Turing tantrums: AI developers rant," ed, 2011.

[105] K. Dill and L. Martin, "A game AI approach to autonomous control of virtual characters," in *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, 2011.

[106] J. Rasmussen, "Are behavior trees a thing of the past," *Retrieved August,* vol. 2, p. 2016, 2016.

[107] B. Bonet and H. Geffner, "Planning as heuristic search," *Artificial Intelligence,* vol. 129, pp. 5-33, 2001.

[108] A. Botea, M. Müller, and J. Schaeffer, "Near optimal hierarchical path-finding," *Journal of game development,* vol. 1, pp. 7-28, 2004.

[109] A. M. Turing, "Digital computers applied to games," *Faster than thought,* 1953.

[110] D. Fudenberg and D. K. Levine, *The theory of learning in games* vol. 2: MIT press, 1998.

[111] W. R. It, "Portfolio selection: efficient diversification of investments," 1959.

[112] N. Justesen and S. Risi, "Continual online evolutionary planning for in-game build order adaptation in StarCraft," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017, pp. 187-194.

[113] N. Shaker, G. N. Yannakakis, and J. Togelius, "Towards Automatic Personalized Content Generation for Platform Games," in *AIIDE*, 2010.

[114] N. Shaker, S. Asteriadis, G. N. Yannakakis, and K. Karpouzis, "Fusing visual and behavioral cues for modeling user experience in games," *IEEE transactions on cybernetics,* vol. 43, pp. 1519-1531, 2013.

[115] P. M. Blom, S. Bakkes, C. T. Tan, S. Whiteson, D. M. Roijers, R. Valenti, *et al.*, "Towards Personalised Gaming via Facial Expression Recognition," in *AIIDE*, 2014.

[116] C. Holmgård, A. Liapis, J. Togelius, and G. N. Yannakakis, "Evolving personas for player decision modeling," in *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, 2014, pp. 1-8.

[117] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 661-668.

[118] D. Loiacono, P. L. Lanzi, J. Togelius, E. Onieva, D. A. Pelta, M. V. Butz, *et al.*, "The 2009 simulated car racing championship," *IEEE Transactions on Computational Intelligence and AI in Games,* vol. 2, pp. 131-147, 2010.

[119] M. W. Floyd, B. Esfandiari, and K. Lam, "A Case-Based Reasoning Approach to Imitating RoboCup Players," in *FLAIRS Conference*, 2008, pp. 251-256.

[120] S. Ontanón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, "RTS AI problems and techniques," *Encyclopedia of Computer Graphics and Games,* pp. 1-12, 2015.

[121] J. Rubin and I. Watson, "On combining decisions from multiple expert imitators for performance," *IJCAI-11,* 2011.

[122] L. Lamontagne, F. Rugamba, and G. Mineau, "Acquisition of cases in sequential games using conditional entropy," in *ICCBR 2012 workshop on TRUE: Traces for reusing users' experience*, 2012.

[123] K. T. Andersen, Y. Zeng, D. D. Christensen, and D. Tran, "Experiments with online reinforcement learning in real-time strategy games," *Applied Artificial Intelligence,* vol. 23, pp. 855-871, 2009.

[124] C. Holmgård, A. Liapis, J. Togelius, and G. N. Yannakakis, "Generative agents for player decision modeling in games," in *FDG*, 2014.

[125] C. Liaw, W.-H. Wang, C.-T. Tsai, C.-H. Ko, and G. Hao, "Evolving a team in a first-person shooter game by using a genetic algorithm," *Applied Artificial Intelligence,* vol. 27, pp. 199-212, 2013.

[126] T. S. Ferguson, *Mathematical statistics: A decision theoretic approach* vol. 1: Academic press, 2014.

[127] M. Mateas and A. Stern, "An Experiment in Building a Fully-Realized Interactive Drama," in *Game Developers Conference, Game Design Track (March 2003) Google Scholar.*

[128] M. Lopes, T. Cederbourg, and P.-Y. Oudeyer, "Simultaneous acquisition of task and feedback models," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, 2011, pp. 1-7.

[129] M. Molineaux, D. W. Aha, and P. Moore, "Learning Continuous Action Models in a Real-Time Strategy Environment," in *FLAIRS Conference*, 2008, pp. 257-262.

[130] H. Romdhane and L. Lamontagne, "Reinforcement of Local Pattern Cases for Playing Tetris," in *FLAIRS Conference*, 2008, pp. 263-268.

[131] B. Auslander, S. Lee-Urban, C. Hogg, and H. Munoz-Avila, "Recognizing the enemy: Combining reinforcement learning with strategy selection using case-based reasoning," in *European Conference on Case-Based Reasoning*, 2008, pp. 59-73.

[132] N. Sugandh, S. Ontañón, and A. Ram, "On-Line Case-Based Plan Adaptation for Real-Time Strategy Games," in *AAAI*, 2008, pp. 702-707.

[133] M. Floyd, "Improving the performance of a RoboCup case-based imitation agent through preprocessing of the case base," Carleton University, 2009.

[134] M. M. Altaf, B. M. Elbagoury, F. Alraddady, and M. Roushdy, "Extended case-based behavior control for multi-humanoid robots," *International Journal of Humanoid Robotics*, vol. 13, p. 1550035, 2016.

[135] D. S. L. Ariza, A. A. Sánchez-Ruiz, and P. A. González-Calero, "Time Series and Case-Based Reasoning for an Intelligent Tetris Game," in *International Conference on Case-Based Reasoning*, 2017, pp. 185-199.

[136] T. P. Homem, D. H. Perico, P. E. Santos, R. A. Bianchi, and R. L. de Mantaras, "Qualitative case-based reasoning for humanoid robot soccer: A new retrieval and reuse algorithm," in *International Conference on Case-Based Reasoning*, 2016, pp. 170-185.

[137] C. L. Johnson and A. J. Gonzalez, "Learning collaborative team behavior from observation," *Expert Systems with Applications*, vol. 41, pp. 2316-2328, 2014.

[138] T. Fitzgerald and A. Goel, "A case-based approach to imitation learning in robotic agents," in *Intl. Conf. on Case-Based Reasoning Workshop on Case-Based Agents*, 2014.

[139] J. Rubin and I. Watson, "Case-based strategies in computer poker," *AI communications*, vol. 25, pp. 19-48, 2012.

[140] G. Robertson, "Applying Learning by Observation and Case-Based Reasoning to Improve Commercial RTS Game AI," in *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2012.

[141] K. Lam, *A scene learning and recognition framework for robocup clients*: Citeseer, 2005.

[142] A. Davoust, M. W. Floyd, and B. Esfandiari, "Use of fuzzy histograms to model the spatial distribution of objects in case-based reasoning," in *Conference of the Canadian Society for Computational Studies of Intelligence*, 2008, pp. 72-83.

[143] K. Lam, B. Esfandiari, and D. Tudino, "A scene-based imitation framework for robocup clients," *MOO-Modeling Other Agents from Observations,* 2006.

[144] A. Drachen, C. Thurau, J. Togelius, G. N. Yannakakis, and C. Bauckhage, "Game data mining," in *Game analytics*, ed: Springer, 2013, pp. 205-253.

[145] N. Shaker, G. N. Yannakakis, and J. Togelius, "Crowdsourcing the aesthetics of platform games," *IEEE Transactions on Computational Intelligence and AI in Games,* vol. 5, pp. 276-290, 2013.

[146] Z. Chen, M. S. El-Nasr, A. Canossa, J. Badler, S. Tignor, and R. Colvin, "Modeling individual differences through frequent pattern mining on role-playing game actions," in *Eleventh artificial intelligence and interactive digital entertainment conference, AIIDE*, 2015.

[147] E. Klein, M. Geist, and O. Pietquin, "Reducing the dimentionality of the reward space in the Inverse Reinforcement Learning problem," in *Proceedings of the IEEE Workshop on Machine Learning Algorithms, Systems and Applications (MLASA 2011). Honolulu (USA)*, 2011.

[148] E. Klein, M. Geist, and O. Pietquin, "Batch, off-policy and model-free apprenticeship learning," in *European Workshop on Reinforcement Learning*, 2011, pp. 285-296.

[149] N. Shaker, M. Abou-Zleikha, and M. Shaker, "Active Learning for Player Modeling," in *FDG*, 2015.

[150] E. Tomai and R. Flores, "Adapting in-game agent behavior by observation of players using learning behavior trees," in *FDG*, 2014.

[151] R. Zhao and D. Szafron, "Using Cyclic Scheduling to Generate Believable Behavior in Games," in *AIIDE*, 2014.

[152] D. J. Gagne and C. B. Congdon, "Fright: A flexible rule-based intelligent ghost team for ms. Pac-man," in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, 2012, pp. 273-280.

[153] J.-Y. Dai, Y. Li, J.-F. Chen, and F. Zhang, "Evolutionary neural network for ghost in Ms. Pac-man," in *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, 2011, pp. 732-736.

[154] S. Bojarski and C. B. Congdon, "Realm: A rule-based evolutionary computation agent that learns to play mario," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, 2010, pp. 83-90.

[155] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, *et al.*, "Human-level control through deep reinforcement learning," *Nature,* vol. 518, pp. 529-533, 2015.

[156] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*: MIT press, 1998.

[157] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning* vol. 1: Springer series in statistics New York, NY, USA:, 2001.

[158] D. C. Wilson and D. B. Leake, "Maintaining Case-Based Reasoners: Dimensions and Directions," *Computational Intelligence,* vol. 17, pp. 196-213, 2001.

[159] D. B. Leake and D. C. Wilson, "Categorizing case-base maintenance: Dimensions and directions," in *European Workshop on Advances in Case-Based Reasoning*, 1998, pp. 196-207.

[160] B. Smyth and M. T. Keane, "Remembering to forget," in *Proceedings of the 14th international joint conference on Artificial intelligence*, 1995, pp. 377-382.

[161] P. A. Devijver and J. Kittler, *Pattern recognition: A statistical approach*: Prentice hall, 1982.

[162] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI communications,* vol. 7, pp. 39-59, 1994.

[163] F. Ahmad, I. Ahmad, and W. M. Dar, "Identification and classification of voxels of human brain for rewardless-related decision making using ANN technique," *Neural Computing and Applications,* vol. 28, pp. 1035-1041, 2017.

[164] F. Ahmad, M. Maqbool, and N. Lee, "Regularization of voxelwise autoregressive model for analysis of functional magnetic resonance imaging data," *Concepts in Magnetic Resonance Part A,* vol. 38, pp. 187-196, 2011.

[165] S. Song, Z. Zhan, Z. Long, J. Zhang, and L. Yao, "Comparative study of SVM methods combined with voxel selection for object category classification on fMRI data," *PloS one,* vol. 6, p. e17191, 2011.

[166] H. Liu and H. Motoda, *Feature extraction, construction and selection: A data mining perspective* vol. 453: Springer Science & Business Media, 1998.

[167] M. Franceschi, P. Caffarra, R. Savarè, R. Cerutti, E. Grossi, and T. R. Group, "Tower of London test: a comparison between conventional statistic approach and modelling based on artificial neural network in differentiating fronto-temporal dementia from Alzheimer's disease," *Behavioural neurology,* vol. 24, pp. 149-158, 2011.

[168] M. Kumar and S. Mishra, "Teaching learning based optimization-functional link artificial neural network filter for mixed noise reduction from magnetic resonance image," *Bio-medical materials and engineering,* vol. 28, pp. 643-654, 2017.

[169] J. A. Lewis-Peacock and K. A. Norman, "Multi-voxel pattern analysis of fMRI data," *The cognitive neurosciences,* pp. 911-920, 2013.

[170] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in *BMVC,* 2015, p. 6.

[171] J. Togelius, T. Schaul, J. Schmidhuber, and F. Gomez, "Countering poisonous inputs with memetic neuroevolution," in *International Conference on Parallel Problem Solving from Nature,* 2008, pp. 610-619.

[172] D. Patel, M. D. Thompson, S. K. Manna, K. W. Krausz, L. Zhang, N. Nilubol, *et al.*, "Unique and novel urinary metabolomic features in malignant versus benign adrenal neoplasms," *Clinical Cancer Research,* vol. 23, pp. 5302-5310, 2017.

[173] P. Gorniak and I. Davis, "SquadSmart: Hierarchical Planning and Coordinated Plan Execution for Squads of Characters," in *AIIDE,* 2007, pp. 14-19.

[174] M. Ponsen, P. Spronck, and K. Tuyls, "Hierarchical reinforcement learning with deictic representation in a computer game," in *Proceedings of the 18th Belgium-Netherlands Conference on Artificial Intelligence*, 2006, pp. 251-258.

[175] L. Galway, D. Charles, M. M. Black, and C. Fyfe, "Temporal Difference Control within a Dynamic Environment," in *GAMEON*, 2007, pp. 42-47.

[176] S. M. Lucas and J. Togelius, "Point-to-point car racing: an initial study of evolution versus temporal difference learning," in *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, 2007, pp. 260-267.

[177] B. Horne, M. Jamshidi, and N. Vadiee, "Neural networks in robotics: A survey," *Journal of Intelligent & Robotic Systems*, vol. 3, pp. 51-66, 1990.

[178] J. M. P. Menezes and G. A. Barreto, "Long-term time series prediction with the NARX network: An empirical evaluation," *Neurocomputing*, vol. 71, pp. 3335-3343, 2008.

[179] B. De Vries and J. C. Principe, "The gamma model—A new neural model for temporal processing," *Neural networks*, vol. 5, pp. 565-576, 1992.

[180] mathworks.com, " Time Series and Dynamic Systems - MATLAB & Simulink," 2016.

[181] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, pp. 328-339, 1989.

[182] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 7, pp. 1329-1338, 1996.

[183] D. F. Specht, "A general regression neural network," *IEEE transactions on neural networks*, vol. 2, pp. 568-576, 1991.

[184] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, pp. 665-685, 1993.

[185] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence," 1997.

[186] Y. Chen, L. Shen, R. Li, X. Xu, H. Hong, H. Lin, *et al.*, "Quantification of interfacial energies associated with membrane fouling in a membrane bioreactor

by using BP and GRNN artificial neural networks," *Journal of Colloid and Interface Science,* vol. 565, pp. 1-10, 2020.

[187] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Zeitschrift für Medizinische Physik,* vol. 29, pp. 102-127, 2019.

[188] M. Miranda, A. A. Sánchez-Ruiz, and F. Peinado, "A CBR approach for imitating human playing style in Ms. Pac-man video game," in *International Conference on Case-Based Reasoning,* 2018, pp. 292-308.

[189] M. Miranda, A. A. Sánchez-Ruiz, and F. Peinado, "A Neuroevolution Approach to Imitating Human-Like Play in Ms. Pac-man Video Game," in *CoSECivi,* 2016, pp. 113-124.

[190] A. Edwards, H. Sahni, Y. Schroecker, and C. Isbell, "Imitating latent policies from observation," in *International Conference on Machine Learning,* 2019, pp. 1755-1763.

[191] A. Khalifa, P. Bontrager, S. Earle, and J. Togelius, "Pcgrl: Procedural content generation via reinforcement learning," *arXiv preprint arXiv:2001.09212,* 2020.

[192] P. Hingston, *Believable Bots: Can Computers Play Like People?*: Springer, 2012.

[193] Z. Pawlak, "Rough sets," *International journal of computer & information sciences,* vol. 11, pp. 341-356, 1982.

[194] Z. Pawlak, "Rough sets: theoretical aspects of reasoning about data, system theory, Knowledge Engineering and Problem Solving, vol. 9," *Hu, Q., Yu, D., Liu, J., & Wu, C.(2008). Neighborhood rough set based heterogeneous feature subset selection. Information sciences,* vol. 178, pp. 3577-3594, 1991.

[195] F. E. Tay and L. Shen, "Fault diagnosis based on rough set theory," *Engineering applications of artificial intelligence,* vol. 16, pp. 39-43, 2003.

[196] R. Jensen and Q. Shen, "Fuzzy–rough attribute reduction with application to web categorization," *Fuzzy sets and systems,* vol. 141, pp. 469-485, 2004.

[197] F.-y. Deng, "An Ameliorating FP-growth Algorithm Based on Patterns-matrix," *JOURNAL-XIAMEN UNIVERSITY NATURAL SCIENCE,* vol. 44, p. 629, 2005.

[198] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB,* 1994, pp. 487-499.

[199] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data mining and knowledge discovery,* vol. 8, pp. 53-87, 2004.

[200] M. Miranda, A. A. Sánchez-Ruiz, and F. Peinado, "Towards Human-Like Bots Using Online Interactive Case-Based Reasoning," in *International Conference on Case-Based Reasoning,* 2019, pp. 314-328.

[201] C. Lueck, S. Zeki, K. Friston, M.-P. Deiber, P. Cope, V. J. Cunningham, *et al.*, "The colour centre in the cerebral cortex of man," *Nature,* vol. 340, p. 386, 1989.

[202] F. Ahmad, G. Ullah, and S.-H. Kim, "A neighborhood method for statistical analysis of fMRI data," *Open Journal of Biophysics,* vol. 2, p. 15, 2012.

[203] K. J. Friston, "Functional and effective connectivity in neuroimaging: a synthesis," *Human brain mapping,* vol. 2, pp. 56-78, 1994.

[204] N. Van Hoorn, J. Togelius, D. Wierstra, and J. Schmidhuber, "Robust player imitation using multiobjective evolution," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on,* 2009, pp. 652-659.

[205] E. R. Gudnadottir, A. K. Jensen, Y.-G. Cheong, J. Togelius, B. C. Bae, and C. H. Pedersen, "Detecting predatory behaviour in online game chats," in *The 2nd Workshop on Games and NLP,* 2014.

[206] B. Maher, "Can a video game company tame toxic behaviour?," *Nature,* vol. 531, pp. 568-571, 2016.