World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:8, 2009

Critical Analysis of Decision Making Experience with a Machine Learning Approach in Playing *Ayo* Game

Ibidapo O. Akinyemi, Ezekiel F. Adebiyi, and Harrison O. D. Longe

Open Science Index, Computer and Information Engineering Vol:3, No:8, 2009 publications.waset.org/3871/pdf

***Abstract*—**The major goal in defining and examining game scenarios is to find good strategies as solutions to the game. A plausible solution is a recommendation to the players on how to play the game, which is represented as strategies guided by the various choices available to the players. These choices invariably compel the players (decision makers) to execute an action following some conscious tactics. In this paper, we proposed a refinement-based heuristic as a machine learning technique for human-like decision making in playing *Ayo* game. The result showed that our machine learning technique is more adaptable and more responsive in making decision than human intelligence. The technique has the advantage that a search is astutely conducted in a shallow horizon game tree. Our simulation was tested against Awale shareware and an appealing result was obtained.

***Keywords*—**Decision making, Machine learning, Strategy, *Ayo* game.

I. Introduction

GAMES have existed among many ancient peoples and are known in all contemporary human cultures. It has been suggested that the playing of games is one of the keys for defining characteristics of man. This is not unconnected with the fact that the basic constituents of any game are its participating autonomous decision makers, called "players". Basically, decision making is the capability of the players to execute an action following some conscious tactical or strategic choices. The major goal in defining and examining game scenarios is to find good strategies as solutions to the game [1]. The solution is a recommendation to the players on how to play the game, and is given as a tuple of strategies. This has given birth to what is called game-playing in computer science, which has been well studied as an intelligent task in Artificial Intelligence (AI). Generally in game playing, the total number of players may be large, but must be finite and must be known. Each player must have more than one choice, because a player with only one way of selecting can have no strategy and therefore cannot alter the outcome of a game.

According to [2], there are a few basic elements, which are common to all games, such as:

i. **Conflicting objectives:** For a game condition to exist there must be conflict and the possibility of winning or losing, or at least of something of value being at stake.

ii. **Rules**: A game must possess rules, delineating the powers and limitations of players, though the rules may not be completely known to the players.

iii. **Visualization**: A game must be **visualizable,** that is, it must be possible to picture what is going on, and it must possess a certain simplicity or elegance.

iv. **Playability:** Finally, a game must be playable. It must have manageable mechanics of play. And it must be nontrivial strategically, permitting the development of more subtle and effective lines of play as players become familiar with the game.

In order to play a game, a player must possess two characteristics [3]: interest in the objectives of play, and sufficient intelligence to understand the consequences of possible lines of play (though not necessarily fully). More than one type of intelligence may be required. All games require at least some degree of abstract intelligence, while many also require sophistication, judgment (particularly where the human factor is important), creativity, or a combination of these. Computers have come to possess impressive abstract abilities, particularly in game playing, to which considerable effort has been devoted by researchers in AI. There have been several researches on computer game-playing and it has seen

Ibidapo O. Akinyemi is a Ph.D. student in the Department of Computer and Information Sciences, Covenant University, Ota, Nigeria. He holds B. Sc. Mathematical Sciences (Computer Science option) and M. Sc. (Computer Science) . His current research interests are on Computational Intelligence and Software Engineering. He is a member of the Nigeria Computer Society and Computer Professional Registration Council of Nigeria (e-mail: dapkin2002@yahoo.co.uk).

Ezekiel F. Adebiyi is a Senior Lecturer and the group leader of the Bioinformatics unit of the Department of Computer and Information Sciences, Covenant University, Ota, Ogun State, Nigeria. He received his Ph.D. in Algorithm and Bioinformatics from the University of Tuebingen, Germany. He is an international scholar and well known scientist in the field of Bioinformatics in Africa (e-mail: eadebiyi@sdsc.edu).

Harrison O. D. Longe is a Professor of Computer Science in the Department of Computer Sciences, University of Lagos, Nigeria. He is currently the Dean of Student Affairs, University of Lagos, Nigeria. His research interest includes Databases and Software Engineering (e-mail: hodlonge@yahoo.co.uk).

Open Science Index, Computer and Information Engineering Vol:3, No:8, 2009 publications.waset.org/3871/pdf

many successes. Nowadays, computers are able to play chess and other games (like checkers, Othello, and backgammon) at world-champion level. Moreover, a number of games have been solved by computers (e.g connect-four and awari). This study attempts to model the method of playing *Ayo* game with a machine learning technique, develop, and describe a decision making strategy that can always guarantee a player winning in any competition.

The rest of this paper is organized thus: section II gives a brief description of *Ayo* game. Section III presents the decision making process with machine learning technique for playing *Ayo* game and section IV describes the implementation of our decision making with the machine learning in playing *Ayo* game. Section V gives our experimental tests and results. The paper is concluded in section VI.

II. Brief Description of *Ayo* Game

Ayo is the most popular ancient African board game with many different variations among the Yorubas who occupy roughly the south-western states of Nigeria and parts of Republic of Benin. Ayo is a member of a family of board games called Mancala. The term "mancala" is used to indicate a large group of related games that are played almost all over the world [4]. Two persons play Ayo at a time with the board put in between the players as shown in Fig. 1 below. The board is a hollow out plank of wood consisting of two rows of six pits belonging to either rows and each pit contains four seeds of the plant "*Caeselpinia crista*" [5] such that a total of forty-eight seeds are contained in a board at the start of the game. Often, there are two extra hollow normally placed centrally at the end of each rows of the board. These are called "seed bags" that are used to store the captured seeds by each of the players. As the game progresses, each pit can contain any number of seeds or no seed at all. Just like any other games, the ultimate objective of the game is to capture more seeds than the opponent to emerge as winner or have equal number of seeds as a draw game. As a seed is captured, it is removed from the board and put in the seed bag and plays no further part, other than being used to evaluate the current game position.



Fig. 1 Players *of Ayo game*

The objective of the players is to capture their opponent's seeds (as many as possible). A move is made when the first player (which can be any of the players) picks up all the seeds in any of non-empty pits P_1, P_2, …, P_6 on his/her side (home) of the board and deposits one seed at a time into the pits in an anticlockwise direction until all the seeds are deposited. This is called "sowing" the seeds. When the player reaches the end of a row, sowing continues in an anticlockwise direction in the other row. When a player picks a hollow (pit) with so many seeds (12 – 17 seeds otherwise called an *Odu* while the one that contains 23 – 27 seeds also regarded as *Ikare* [6] such that the player passes completely around the board, the originating hollow is skipped and the seed is played in the next hollow on [5]. This means that the originating hollow is always left empty at the end of the turn. If the last seed is sown in the opponent row and the hollow concerned finishes with two or more seeds, those seeds are captured in a clockwise direction until either a hollow does not have two or three seeds in it, or the end of the opponent row is reached. An illustration of players making moves to capture seeds is shown in Fig. 2, where when player 1 plays the seed in the fifth hollow, he/she capture the seeds in hollow 4, 3 and 2 of the player 2 as they result in 3, 2 and 3, making a total of eight seeds being captured, while on the other hand, if player 2 plays the seed in the sixth hollow of his own, he capture the seeds in hollow 1 and 2 of the player 1, making a total of 4 seeds captured. The play progresses this way until the end is reached when a total number of seeds on the board will be equal to or less than five seeds.

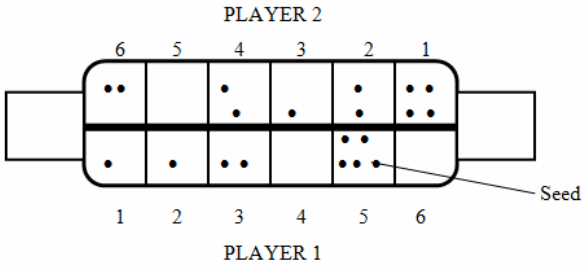


Fig. 2 A capture strategy in Ayo game

One critical rule called the 'golden rule' [7] for capturing seeds is that a player must keep his opponent provided with at least one seed with which to play after he had captured the rest seeds. Any player who, in a bid to capture, contravenes this rule is automatically disallowed from making the capture and so takes nothing even if the content of the holes are technically ripe for capture. As illustrated in Fig. 2, if player 1 plays the seeds in the fourth hole, which contains seven seeds, all the seeds on the player 2 sides qualify for capture. But following this rule, player 1 is not allowed any capture, although the move is allowed.

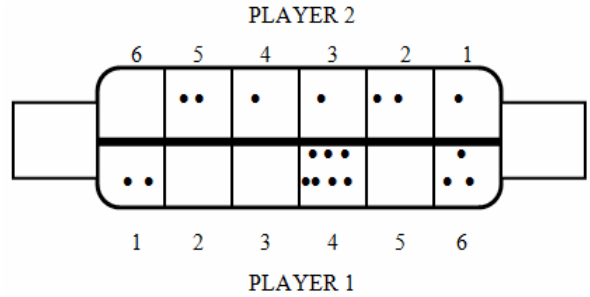


Fig. 3 A golden rule capture strategy in Ayo game

Therefore, player 1 will rather play (or move) the three seeds in hole six to capture seven seeds, while player 2 can play the seeds in his fifth hole to capture three seeds. Sometimes, towards the end of a game, when there are only a few seeds to capture, a player is forced to give up one or more of his seeds for capture in order not to break the golden rule. If during the game, it is found that there are not enough seeds to make a capture, but both players can always proceed with a legal move, the game is stopped and the players are awarded seeds that reside on their respective side of the board. The initial game is rapid and much more interesting, where both the players capture seeds in quick succession. To determine the optimal strategy during the initial play is hard and thus has not yet been studied [8]. It involves planning at least 2–3 moves in advance, and remembering the number of seeds in every pit [5], [9].

III. Decision Making with Machine Learning Approach for Playing *Ayo* Game

The history of the interaction of machine learning and computer game-playing dated back to the early days of AI, when Arthur Samuel worked on his famous checker-playing program, pioneering many machine and game-playing techniques [10], [11]. Game, whether created for entertainment, simulation, or education, provides great opportunity for machine learning [12]. Machine learning is the branch of AI which studies learning methods for creating intelligent systems. These systems are trained with the use of a learning algorithm for a domain specific problem or tasks [13]. Generally, minimax search has been the fundamental concept of obtaining solution to game problems. However, there are a number of limitations associated with using minimax search [8]. These are:

i. improper design of a suitable evaluator for moves before the moves are made, and
ii. inability to select a correct move without assuming that players will play optimally.

It is our believe that eliminating these limitations of the minimax search will improve the playing of *Ayo* game. For example, in a game scenario, a player can be irrational in move selection as a play strategy such as bluffing as applicable in *Ayo*. Bluffing is a powerful play strategy and is defined as the ability to tradeoff an invaluable seeds so as to gain advantage. Hence, it involves sacrificing immediate reward to obtain a greater reward in the long term. But two important factors that must be taken into consideration when bluffing are:

i. when to bluff, and
ii. the number of seeds (i.e tradeoff seeds) to sacrifice.

Consequently, the effect of a single move can be so large that it becomes incalculable for human in competitive situations. More importantly, it is unnecessary to search large game positions to evolve an Ayo player that performs pretty well. Human's brain functions in a way that a person does not necessarily take much time to solve a problem. Human solves complex problems using approximate matching. Machine intelligence is supposed to be an impersonation of human intelligence, therefore, there is no need looking for exact solution (that may not exist) to a complex problem when its approximation is equally good. Hence, complementing minimax search at shallow depth with machine learning techniques provides a good approximation to deeper search. In order to construct an efficient evaluator for the minimax search, a refinement-based machine learning techniques is imperative. Machine learning techniques like neural network, nearest neighbor search and case-based reasoning are important refinement tools that we have investigated for accomplishing this arduous prediction task. The idea of minimax algorithm is synonymously related to these otimization procedures. Max player tries as much as possible to increase the minimum value of the game, while Min tends to decrease its maximum value at a node as both players play towards optimality. This process can be described by the algorithm in Fig. 4.

```
MinMax (GamePosition game) {
        return MaxMove (game);
}
MaxMove (GamePosition game) {
        if (GameEnded(game)) {
                return EvalGameState(game);
        }
        else {
                best_move <- {};
                moves <- GenerateMoves(game);
                ForEach moves {
                        move <- MinMove(ApplyMove(game));
                        if (Value(move) > Value(best_move)) {
                                best_move <- move;
                        }
                }
                return best_move;
        }
}
MinMove (GamePosition game) {
        best_move <- {};
        moves <- GenerateMoves(game);
        ForEach moves {
                move <- MaxMove(ApplyMove(game));
                if (Value(move) > Value(best_move)) {
                        best_move <- move;
                }
        }
        return best_move;
`
```

Fig. 4 Basic Minimax Algorithm

Open Science Index, Computer and Information Engineering Vol:3, No:8, 2009 publications.waset.org/3871/pdf

Open Science Index, Computer and Information Engineering Vol:3, No:8, 2009 publications.waset.org/3871/pdf

In this work, we complemented the minimax algorithm with a refinement method and the entire algorithm then has three main components thus: (1) build game tree, (2) compute game value and (3) refine feasible moves. The "buildTree" procedure constructs a game tree in top-down fashion using breath-first traversal and nodes are evaluated as fan out is made to all nodes adjacent to their immediate parents. The "computeValue" procedure computes the game value bottom-up and "predictMove" uses move refinement procedure (MRP) to predict the best move among few ones recommended by minimax.

IV. Implementing Decision Making of the Refinement-Based Machine Learning Approach

We followed [8] for the implementation of our refinement-based procedure. It comprises of three modules; Basic Refinement Minimax (BRM), Priority, and Similarity. A move refinement is a mapping that accepts a set of moves and then evaluates each move and returns a single move with best advantage using a simple myopic decision. The basic refinement minimax is represented in a myopic rule as:
Given a game state, let move[k] = {m_1, m_2, …, m_k} be a set of k feasible moves. We call m_k the head and m_1 the tail. A move is protected if it is not vulnerable to being forfeited when the opponent plays

(1) If k=1 Then select the only available move and stop
(2) If tail/head is not protected for South/North player respectively Then select it Else select a move with the highest mobility strength (that is, tendency to have more possibilities of move).

For the purpose of move classification, we described an algorithm called "priority" in which moves are respectively classified into two classes: C_1 (class of moves that gives the south player a better advantage) and C_2 (class of moves that gives opponent a better advantage) using the perceptron learning algorithm [14]:

$$w_{k+1} = \begin{cases} w_k + \eta y_i, & if \quad g(x) < 0 \quad and \quad y_i \in C_1 \\ w_k - \eta y_i, & if \quad g(x) \geq 0 \quad and \quad y_i \in C_2 \end{cases} \tag{1}$$

Where y_i is an example misclassified by the weight w_k, η is the learning rate, w_0 is a threshold weight and g(x) is a linear discriminant function of the input vector x given by the following equation:

$$g(x) = w_0 + \sum_{i=1}^{m} w_i x_i \tag{2}$$

If the feature vectors in C_1 have higher priority than those in C_2. Then a vector in C_1 with farthest distance from the separating hyper-plane is selected. However, if all vectors are found in C_2, BRM algorithm is applied. This algorithm is described more compactly by the following pseudo-code.

(1) Let x_1, x_2, …, x_n be moves recommended by minimax algorithm
(2) Classify these moves into C_1 and C_2
(3) If C_1 is not empty Then
select k moves, which satisfy x_k=max{g($V(x_k)$)} where $V(x_k)$ is a feature vector of x_k Else
select all moves in C_2, store selected moves in the array move[m] and store the dimension of move[m] as m
(4) Apply BRP to the moves stored in move[m] array.

In order to measure the similarity for effective classification, strategies are simply regarded as episodes. The similarity between the i^{th} target episode x_i and the source episodes y_j of the j^{th} class is computed and the largest similarity measure is returned. The target episode with the maximum similarity greater than or equal to a given threshold value (called bluffing threshold) and a game value less or equal to the tradeoff seed is selected. Bluffing is the ability to tradeoff some seeds so as to gain an advantage of capturing more seeds in the nearest future. Otherwise priority algorithm is applied. The procedure is described succinctly with the flowchart below:

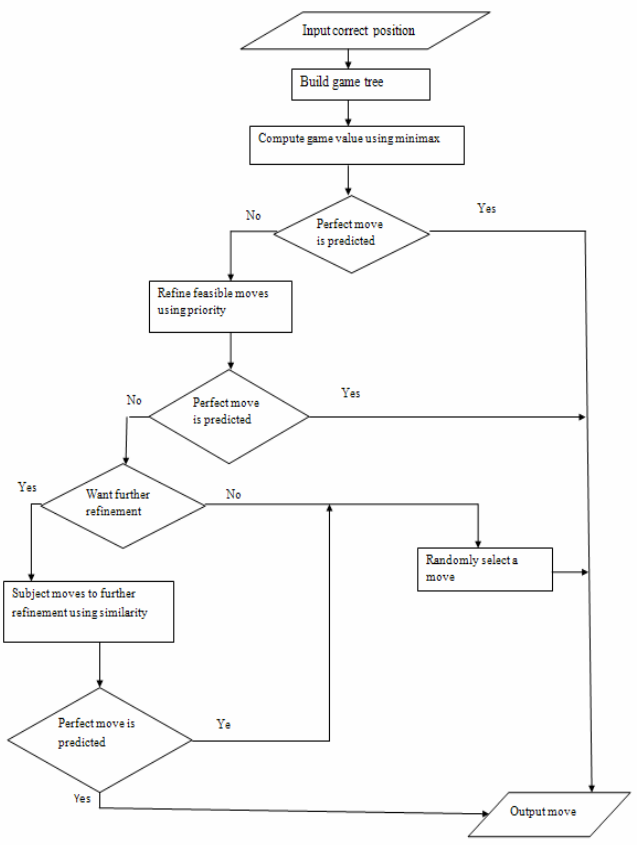


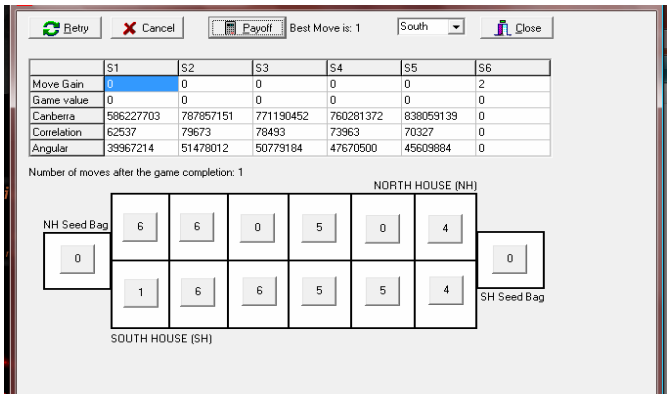
Fig. 5 Flowchart Showing the Refinement Procedure

The similarity between two episodes x_i and y_j is calculated using the Canberra, Correlation, and Angular product-moment coefficient [15] for the purpose of comparison because none of them could function well when used in isolation.

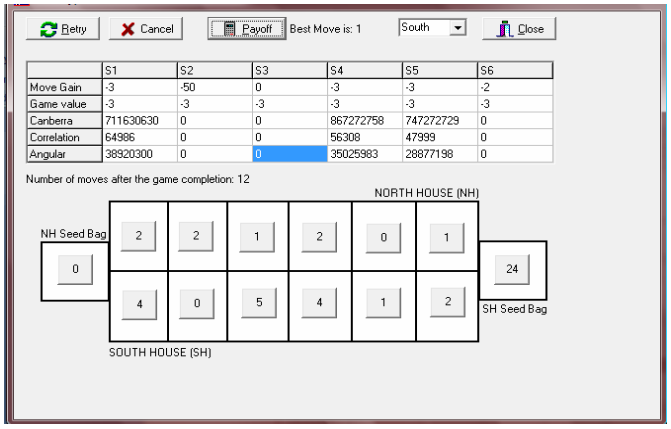
V. Experimental Tests and Results

The above method was implemented using C++ and a sample simulation showing decision making process for a best move in playing *Ayo* game is shown in Fig. 6.

Open Science Index, Computer and Information Engineering Vol:3, No:8, 2009 publications.waset.org/3871/pdf



(a)



(b)

Fig. 6 A Prototype Simulation of Ayo Game

From our simulation, we observed that the Canberra distance is very sensitive to small changes near zero, while the correlation metric suggests a move faster than the other two metrics. In Fig. 6a, the metrics suggested that the best house for south player to move is house s_1, which in actual fact agrees with the thought of an expert player. Here the least value for the similarity measure is selected but for the purpose of mobility and bluffing, the highest value is selected (see correlation and angular metrics) as illustrated in Fig. 6b to complement our decision making process.

The performance of the refinement procedure was evaluated by playing a series of games with Awale shareware, which we registered to play with. The results obtained from a series of six games played at each level, for which each player was allowed to start thrice are recorded in Table I using the playing rules presented in section II.

TABLE I
RESULTS OF PLAYING AWALE USING REFINEMENT HEURISTICS AT FOUR LEVELS

Play Level	Our Results Using BRM		
	Average Moves	Seeds Captured	
		Evolved Player	Awale Shareware
Initiation	19	26	4
Beginner	37	26	11
Amateur	70	23	17
Grandmaster	36	6	31
	Our Results Using Priority		
Initiation	18	26	3
Beginner	38	25	10
Amateur	41	21	18
Grandmaster	43	13	31
	Our Results Using Similarity		
Initiation	27	26	7
Beginner	36	27	10
Amateur	48	25	14
Grandmaster	41	25	15

Our *Ayo* simulation (henceforth called evolved player) favorably competed with Awale shareware at all levels as results show. On average, it captured majority of seeds after playing at initiation level using the three refinement algorithms respectively. Similarly, it equally won at beginner and amateur levels, although Awale defeated once at beginner level and twice at the amateur level using priority algorithm. The evolved player shows a promising performance over previous methods, since it captured more seeds. However, only the similarity algorithm defeated Awale convincingly at the grandmaster level having captured 25 seeds as shown in column 3 and 4 of Table I, column 1 indicates the various play level in Awale shareware and column 2 gives the number of game moves before the end of game play.

VI. CONCLUSION

We have been able to simulate the human decision making process in playing *Ayo* game, which can make optimal or nearly optimal decisions and is able to maintain those decisions over time with little or no human supervision. The algorithm employed to evolve our *Ayo* player is computationally effective and has the tendency to incorporate new play strategies in form of expert instruction and then become more sensitive to its mistakes/weaknesses. In our future work, we intend to research in using neural network to predict game move and compare its behaviour with our refinement-based heuristic.

ACKNOWLEDGMENT

We would like to acknowledge Mr. J. O. Daramola of the Department of Computer and Information Sciences, Covenant University, Ota, Nigeria for his valuable comments towards improving the quality of this paper. Dr. Olugbara, O. O. is equally acknowledged for his contribution in the implementation aspect of this work.

Open Science Index, Computer and Information Engineering Vol:3, No:8, 2009 publications.waset.org/3871/pdf

REFERENCES

[1] O. A. Adeday*o,* Optimal Strategies in Two-Person Zero-Sum Games. A Seminar paper presented at Covenant University, Ota, Nigeria. May 25, 2005

[2] M. O. Ajayi, The Soccer Pitch and the Arena of Politics in Nigeria. A Paper Presented at the Public Lecture Delivered at Covenant University, Ota, Nigeria, on 27th February, 2007.

[3] D. .M. Kreeps, Game Theory and Economic Modeling, Oxford Unversity Press, 1990.

[4] H.J.R. Murray, A History of Board Games other than Chess. Clarendon Press, Oxford, (1952..

[5] A. O. Odeleye, "Ayo": A popular Yoruba game, Oxford University Press, Ibadan, Nigeria, 1977.B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.

[6] E. F. Adebiyi, A step further toward Improving "Ayo" Game System. *AMSE*, 3(2);, pp.53- 69, 1999.

[7] T. O. Adewoye on Certain Combinatorial Number Theoretic Aspects of the African Game of "Ayo". *AMSE REVIEW*, vol. 14, no, 2, pp. 41-63, 1990.

[8] O. O. Olugbara, T. O. Adewoye, and I. O. Akinyemi, An Investigation of Miinimax Search Techniques for Evolving Ayo/Awari Player. Accepted for Publication, in IEEE - ICICT Conference Proceedings, 10 – 12 December 2006

[9] D.M. Broline, and D.E. Loeb, The Combinatorics of Mancala-type games: *Ayo*, Tchoukaillon and 1/π. *UMAP*, J., 10(1), 1995.

[10] A. L. Samuel, Some Studies in Machine Learning Using the Game of Checkers, *IBM J. of Res. And Dev*. 3, 210-229, 1959.

[11] A.L. Samuel, Some Studies in Machine Learning Using the Game of Checkers II- recent Progress, *IBM J. of Res. And Dev*. 11(6), 601 – 617, 1967.

[12] M. Bowling, J. Furukranz, T. Graepel, and R. Musick, Machine Learning and Games. *Springer Science plus Business Media, LLC, Mach Learn* (2006), m vol. 63, pp 211 – 215.

[13] M. Henk, learning to play chess unsing Reinforement Learniing with Database Games. M.Sc. Thesis, Cognitive Artifiial Intelligene, Utreht universty, 2003.

[14] J. Jantzen, Introduction to Perceptron Networks. Technical Report No 98-H873, Department of Automation, Technical University of Denmark, 1998.

[15] V. L. kristof, Basic Statistics and Metrics for Sensor Analysis, 2004. Available online – http://ubicomp.jancs.ac.uk/~kristof/research/notes/basicstats/index.html. Downloaded 20/10/2007.