ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

TESI DI LAUREA

in

Intelligent Systems M

SOCCER COACH DECISION SUPPORT SYSTEM

CANDIDATO
Gustavo Huatuco Santos

RELATORE:
Chiar.ma Prof.ssa
Michela Milano

Anno Accademico 2017/18

Sessione III

To God ...

Abstract

L'azienda sportiva sta attraversando una gigantesca trasformazione digitale incentrata sulle immagini, tempo reale e analisi dati utilizzati nelle competizioni. I metodi di processo convenzionali nella gestione dello sport come il fitness, la salute, la esercitazione, lo sviluppo, la realizzazione di partite e i fondamenti sono tutti rivoluzionati dalla digitalizzazione dello sport. È sempre noto che è necessaria una metodologia digitale semplice per organizzare e costruire una strategia fattibile. Concordiamo che la soluzione sia una digitalizzazione sportiva cui evoluzione continua richiederebbe sfide così pervasive. La bontà della digitalizzazione degli sport ed atleti si basa su ciò che viene fatto con la raccolta di suoi dati. Il vantaggio competitivo va a coloro che possono produrre la massima intelligenza dei dati e agire tempestivamente. L'impatto su tutti gli caratteristiche delle operazioni della squadra sportiva è potenziale. I dati non prendono tutte le decisioni, ma consentono decisioni consapevoli.

In queste circostanze e con una visione e predilezione per gli sport calcistici, abbiamo concepito un sistema di supporto decisionale, col nostro problema prospettivo di come un sistema di supporto decisionale per allenatori di calcio collabori con loro per prendere decisioni efficientemente.

Per affrontare questo problema, elaboriamo un sistema di supporto alle decisioni degli allenatori di calcio. Il sistema è organizzato in due componenti combinati; la prima simula la predizione del vincitore della partita mediante una rete neurale guidata di dati. Questa uscita attiva successivamente il secondo componente per applicare regole logiche di apprendimento e provedere l'analisi delle informazioni statistiche, consigliare le decisioni da svolgere ed inoltre pianificare le caratteristiche sportive che sperano essere migliorate come esercitazioni e addestramenti di preparazione alle prossime partite allineate con la loro concezione e il loro stile di gioco.

I punti di vista di alcuni allenatori professionisti hanno fornito generosamente estensioni, ad esempio l’analisi delle caratteristiche mentali e morali delle squadre perché la sua importanza è vista quando a volte le performance delle squadre o degli atleti cambiano improvvisamente e si esibiscono inaspettate mostre sportive con stupefacenti risultati. Inoltre dispositivi di tracciamento in tempo reale durante lo svolgimento della partita per avere informazione instantanea.

Contents

Illustration Index

Index of Tables

Introduction

The savage essence and nature of sports means those who work on it hunt for the wins. The sport enterprise is undergoing a gigantic digital transformation focused on imaging, real time and data analysis employed in the competitions. Conventional process methods in sports management such as fitness and health establishments, training, growth and match or game realisation are all being revolutionized by the sport digitization. In team sports it is well known that is needful an enough and simple digital methodology to organize and construct a feasible strategy. Digitization in sports is perpetually evolving and requires pervasive challenges. The sports and athletics digitization success is based on what is being done with collection of more data. Competitive advantages go to those who produce powerful operations using the data and acting on it in real time. The potential impact of these sport features in sport team operations is powerful. Data does not ride all decisions, but it empowers knowledgeable decisions.

In these world circumstances, our vision with this system was born from a dream helping soccer sport management systems embrace and improve its contest success. Our perspective problem is how a decision support system for soccer coaches helps them to take enhancement decisions better.

To face this problem we have created a soccer coach decision support system. This system is organised in two joined components; the first simulates the prediction of the soccer match winner through a data driven neural network. This component output activates the second to operate the logic rules learning and provides the stats, analysis, decision making and additionally plans improvements like drills and training procedures. This helps on the preparation towards upcoming matches as well as being aligned with their style and playing concepts.

Future scalabilities and developments, will analyse timely the mental and moral features of the teams because their importance is seen when teams or athlete’s behavior change and they display unexpected performance with astonishment match results.

Chapter 1

Overview of decision support systems and soccer management methods

1.1 Decision Support Systems, Categorization and Development

When it comes to good decision making [1], relying too heavily on automatic decisions stemming from perception or depending too much on conventions when information is bombarded on to us from all sides can be dangerous. Sometimes, we fail to either notice or seek out crucial information that supports decision making. This may be because of our biasness or shortage of time, funds and other resources.

In many situations, we are unable to apply fundamentals of economics, statistics and operations research to make lucid choices. This is where a decision support system comes into picture. It is a computer-based system that helps us make planning, manufacturing, operations and management decisions, based on information available. But these systems are not the decision makers. They just aid in decision making, by offering insights that we may be missing and in providing exact calculations but the ultimate decision maker are only us.

1.1.1 Decision Support System (DSS)

A decision support system:

- is a computer-based application or program
- that compiles, combines and analyzes raw data, documents, fundamentals of social science, applied science, mathematics and managerial science, and personal knowledge of decision makers

- to identify problems and determine their solutions
- in order to facilitate optimal decision making.

A decision support system helps overcome the barriers to a good decision making, including:

- lack of experience
- biasness
- shortage of time
- wrong calculations
- not considering alternatives

The Decision Support Systems can be divided into following categories:

1.1.1.1 Model-driven DSS

A model-driven DSS is based on simple quantitative models. It uses limited data and emphasized manipulation of financial models. A model-driven DSS is used in production planning, scheduling and management. It provides the most elementary functionality to manufacturing concerns.

1.1.1.2 Data-driven DSS

Data-driven DSS emphasizes the access and manipulation of data tailored to specific tasks using general tools. While it also provides elementary functionality to businesses, it relies heavily on time-series data. It is able to support decision making in a range of situations.

1.1.1.3 Communication-driven DSS

As the name suggests, communication-driven DSS uses communication and network technologies to facilitate decision making. The major difference between this and the previous classes of DSS is that it supports collaboration and communication. It made use of a variety of tools including computer-based bulletin boards, audio and video conferencing.

1.1.1.4 Document-driven DSS

A document-driven DSS uses large document databases that stores documents, images, sounds, videos and hypertext docs. It has a primary search engine tool associated for searching the data when required. The information stored can be facts and figures, historical data, minutes of meetings, catalogs, business correspondences, product specifications, etc.

1.1.1.5 Knowledge-driven DSS

Knowledge-based DSS are human-computer systems that come with a problem-solving expertise. These combine artificial intelligence with human cognitive capacities and can suggest actions to users. The notable point is that these systems have expertise in a particular domain.

1.1.1.6 Web-based DSS

Web-based DSS is considered most sophisticated decision support system that extends its capabilities by making use of worldwide web and internet. The evolution continues with advancement in internet technology.

DSS applications are not single information resources, such as a database or a program that graphically represents sales figures, but the combination of integrated resources working together.

These can support decision making in situations where precision is of importance. Additionally, they provide access to relevant knowledge by integrating various forms and sources of information, aiding human cognitive deficiencies. While DSS employs artificial intelligence to address problems, the end decision remains with the user.

1.1.2 Categorization/Classification of DSS

Let us now look at the categorization on the basis of nature of operations:

1.1.2.1 File Drawer System

As the name suggests, a file drawer decision support system provides information useful for making a specific decision. It works like a file drawer where different types of information are stored under different names or categories.

1.1.2.2 Data Analysis Systems

These decision support systems are based on a formula; and therefore, are used to make comparative analysis. These make use of simple data processing tools, such as inventory analysis.

1.1.2.3 Information Analysis System

This kind of decision support system analyzes different sets of data to generate informational reports that can be used to assess a situation for decision making.

1.1.2.4 Accounting and Financial Support System

This type of support system is based on to keep track of cash and inventory.

1.1.2.5 Representation or Solver Model

This type of system performs or represents decision making in a particular domain or for a specific problem. It calculates and compares the outcomes of different decision paths. The decision maker can conduct a ‘what if’ analysis and make an informed decision basis on the outcomes generated.

1.1.2.6 Optimization Model

This DSS is based on simulated models, majorly providing guidelines for operations management. The focus is on providing optimal solutions on job scheduling, product mix and material mix decisions.

1.1.2.7 Suggestion System

This type of support system suggests optimal decision for a particular situation by assisting in collecting and structuring data.

1.1.2.8 Categorization of DSS on the Basis of Inputs

- Text Oriented DSS
- Database Oriented
- Spreadsheet Oriented
- Rule Oriented
- Solver (specific situation) Oriented
- Compound/Hybrid: This support system combines two or more structures from above to offer multiple functionalities.

1.1.2.9 Categorization of DSS on the Basis of Support Offered

- Personal DSS
- Group DSS
- Organizational DSS

1.1.2.10 Categorization of DSS on the Basis of Type and Frequency of Decision Making

a. Institutional DSS

An institutional decision support system supports recurring decisions on an ongoing basis. Basically, this is for programmed decisions, which are made on daily basis. For example, establishing routine for handling technical problems, taking disciplinary actions, unit manufacturing, a mechanic process of troubleshooting, etc.

b. Ad-hoc DSS

An ad-hoc decision support system supports one kind of decision in an unanticipated situation. The decision made is unique to a problem. This type of system is used to support non-programmed decisions as the information available is incomplete.

1.1.3 Components of a Decision Support System

Like any other software system, DSS also has components and phases of development. No matter what kind of decision support system we are looking to develop, we must plan around these four components:

- **Input:** What kind of input does it require to carry out the analysis? As mentioned earlier, it can be rule, problem, spreadsheet, text or database oriented.

- **User Knowledge/Expertise:** Whether inputs will require manual analysis by the user or not.
- **Output:** Should the outcomes be comparative or generic?
- **Decisions:** Whether it should be a suggestion support system? Or we just want it to analyze the data and outcome of different actions?

1.1.4 Designing and Building a Decision Support System

Developing a DSS is a complex process and thus, takes longer. It goes repetitively through three stages: inputs, activities and outputs during each phase of system development lifecycle. We provide an input, carry out the desired activity and measure the output. Moving us further, if it produces the right output or else we come back to the input phase and make adjustments.

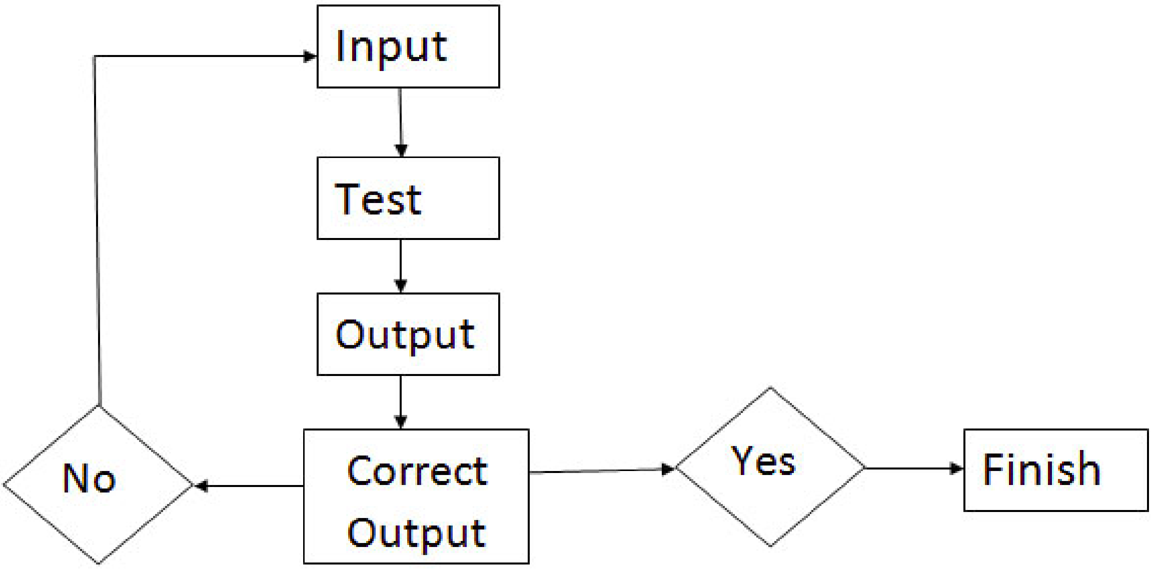


Figure 1.1 Flow diagram of the DSS components.

A DSS framework design and development goes through these stages:

1.1.4.1 Intelligence

At this stage, the objective is to search for problems/situations/conditions that call for decision.

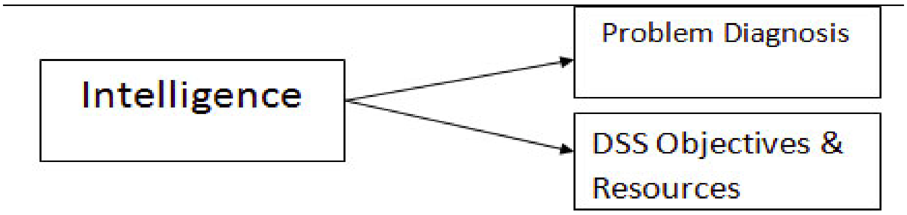


Figure 1.2 Intelligence stage of the DSS framework design.

The user, as an expert, is expected to identify and define the problem context for which support is required. He must define the objectives and available resources, so that the outcomes generated meet his expectations.

1.1.4.2. Design

This stage deals in analyzing all possible actions, along with the determination of system design and system construction.

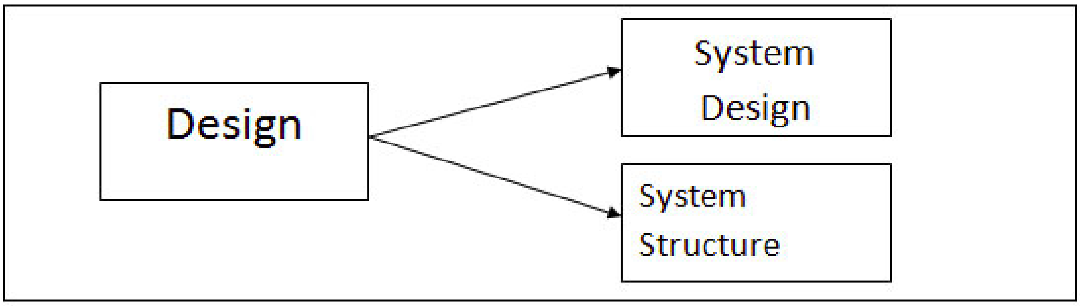


Figure 1.3 Design stage of the DSS framework.

System design includes determination of components, platform, function libraries and special languages while system structure is about deciding the prototype approach. This stage also includes identifying hardware requirements. The development starts here.

1.1.4.3 Choice

Once we shortlist and analyze all possible courses of actions in design phase, now is the time to choose the best from among them, depending upon our business objectives and results generate.

1.1.4.4 Implementation

This is the final stage where testing, evaluation, adjustments and deployment take place. However, this is the final product but this can be tweaked, refined and upgraded basis your activities and requirements.

When developing a custom DSS, these are important factors that must be kept in mind:

- Data management functions
- Available hardware platforms
- User interface
- Compatibility with other applications
- Cost

1.1.5 Building Knowledge-Driven Decision Support System and Mining Data

Before the development of knowledge-driven DSS, people with high intellect had to perform knowledge-intensive tasks. An expert in a particular area would know how to approach a problem and go about it. Similarly, knowledge-based DSS asks relevant questions, offers suggestions and gives advice to solve a problem. The only difference is that it is automated and speeds up the whole process.

1.1.5.1 Knowledge-Driven DSS

A knowledge-driven DSS:

- is a computer-based reasoning system
- that provides information, comprehension and suggestions to users
- to support them in decision-making.

It is an integration of computerized business intelligence tools and technologies customized to the needs and requirements of an organization. So, the focus is on:

- Identifying specific knowledge sharing and distribution needs of a company
- Setting objectives that need to be attained with a knowledge-driven DSS
- The selection of appropriate tools and technologies
- Understanding the nature of work and decision-making performed by its potential users
- Selecting data mining techniques

1.1.5.2 Key Terms and Concepts

A computer-based reasoning system is similar to any other type of decision support system when it comes to their architecture. But it turns into a knowledge-drive decision support system when artificial intelligence technologies, management expert systems, data mining capabilities and other communication mechanisms are integrated.

Let us learn about few important terms and concepts used alongside knowledge-driven decision support system. This will help gain an in-depth understanding of such support systems.

a. Expertise

A knowledge-driven DSS comes with a specific problem-solving expertise. This expertise is based upon three components:

- Knowledge in a particular domain and associated symptoms and signs.
- Understanding of the relationships between varied symptoms of a problems.
- Skills, ways or methods of solving the problem.

b. Expert System

A computer system that imitates the decision making capability of a human expert is called an expert system or an artificial intelligence system. It is designed to solve problems by:

- Using if-then rules
- Reasoning about knowledge
- Drawing inferences from facts and rules

b.1 Production-rule systems

b.1.1 Definition

Programs that implement research methods for problems represented as state space [2]. They consist of:

– a set of rules,

– a 'working memory', which contains the current states reached

– a control strategy to select the rules to be applied at the states of the 'working memory' (matching, verification of preconditions and test on goal state if achieved).

General Architecture:

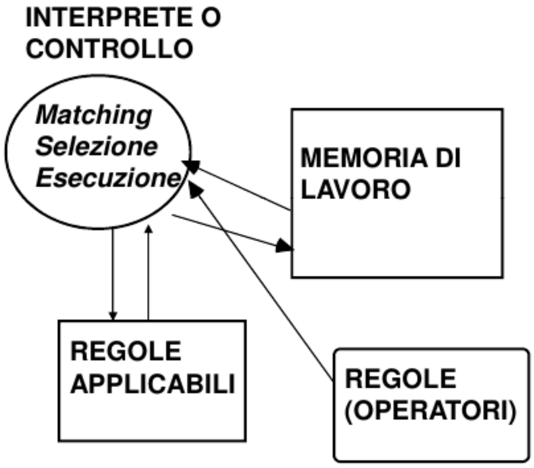


Figure 1.4 Production-rules system architecture.

b.2 Two modalities of "reasoning"

b.2.1 Forward or data-driven

– The working memory in its initial configuration contains the initial knowledge about the problem, those are the known facts.

– The applicable production rules are those whose antecedent can match with the working memory (F-rules).

– Each time a rule is selected and performed, new demonstrated facts are inserted in the working memory.

– The procedure ends with success when in the working memory the goal to be demonstrated is also inserted (termination condition).

b.2.2 Backward or goal-driven

– The initial working memory contains the goal (or goals) of the problem.

– The applicable production rules are those whose consequent can match with the working memory (B-rules).

– Each time a rule is selected and executed, new subgoals to be demonstrated are inserted into the working memory.

– The process ends with success when in the working memory are inserted known facts (termination condition).

c. Knowledge Discovery and Data Mining

These are interrelated terms used for the process of extracting valuable knowledge and discovering patterns, in order to transform the knowledge into easily comprehendible structure for further use. Data mining is a buzzword but a misnomer. This is because data mining is a process of collection, storing and analysis of data and not finding patterns. Knowledge discovery goes through a series of steps:

- Selection
- Pre-processing
- Transformation
- Data mining
- Interpretation

d. Development Environment

It is the environment in which a decision support system is developed. It typically includes software for creating a DSS and knowledge base. The development environment may vary in size, depending upon production/development needs.

e. Domain Expert

A domain expert is a subject matter expert who has expertise/authority in a particular domain. A domain expert is an integral part of the team working on developing a decision support system.

f. Knowledge Engineer

A technical expert who integrates knowledge into a computer system when developing a decision support system, in order to solve complex problems that require human expertise.

g. Knowledge Acquisition

It is extraction/mining of knowledge from various sources, such as experts, databases and external programs.

h. Knowledge Base

It is the collection and storage of structured (facts, rules, regulations, characteristics, functions, procedures and relationships) and unstructured information that will be used by a DSS in decision making.

i. Interface Engine

It is a software system to simplify the conception and development of application interfaces between application systems. Typically, it is a middleware application to transform, route and translate messages between various communication points.

j. Heuristic

It is an approach to discovery and problem solving by employing practical methods. These methods may not be optimal but can help achieve immediate goals.

These technical jargons are important in this field of use, in order to gain a deeper understanding of knowledge-driven DSS.

1.1.5.3 Characteristics of Knowledge-Driven Decision Support Systems

A knowledge-driven DSS is different from conventional systems in the way knowledge is extracted, processed and presented. The former attempts to emulate human reasoning while the latter responses to an even in a predefined manner. The main characteristics of knowledge-driven decision support systems are:

- These systems aid managers in solving complex problems.
- These systems allow users to interact with them during the process of decision making.
- The recommendations made by these systems are based on human knowledge.
- These systems use knowledge base that is engineered keeping in mind the nature of problems they will solve.
- These systems aid in performing limited tasks.
- These systems use heuristic technique of problem solving.

1.1.5.4 Managing Knowledge-Driven Decision Support System Projects

Knowledge-driven decision support systems are expert systems that are developed when decision-making cannot be supported using traditional methods. A knowledge-driven DSS project goes through various stages and can be difficult to manage. It is important to be committed to monitor the development of a knowledge-driven DSS.

1.1.5.4.1 Development Stages

- Domain identification (choosing a subject matter)
- Conceptualization (idea formation, feasibility testing and commencement)
- Formalization (beginning with development officially)
- Implementation (completion and execution)
- Testing (fixing errors and modifications)

It is important to monitor project development throughout very closely. It is a collective effort of knowledge engineers, domain experts, DSS analysts, users and programmers. And a project manager keeps track of the scope, time, quality and budget, to ensure optimum allocation of resources and creation of a quality product. A project manager is a person responsible for accomplishing the pre-decided objectives of a project.

1.1.5.5 Tools and Techniques

There are a large number of tools and techniques used to extract/mine data. Which technique is to be used depends on the type of data to be extracted.

a. Case-based Reasoning

Case-based reasoning (CBR) tools are used to determine the distance between or relationship among various components. A problem solved using this tool goes through five stages:

- Presentation – the problem is described and entered into the system.
- Retrieval – the system matches it with the cases stored in the system.
- Adaptation – the system matches the retrieved closest-matching case and the problem to generate a solution.
- Validation - the solution then goes through a validity test and is justified if the user gives a positive feedback.
- Update – the valid solution is accepted and added to the case base in the system.

b. Fuzzy Query and Analysis

Fuzzy query and analysis is a data mining tool follows the mathematical concept for ‘fuzzy logics – the logic of uncertainty’ to determine results that are close to a particular criterion. Users can then pick one, depending upon his or her understanding.

c. Data Visualization

As the same suggests, this helps analysts visualize complex relationships in multi-dimensional data. The benefit is that this tool graphically represents relationships among components from different perspectives. Statistical tools, such as regression, classification or cluster analysis are a part of this tool.

d. Genetic Algorithms

Similar to linear programming models, genetic algorithms conduct random experiments by selecting the genes (variables whose values are to be identified) and their values at random to find the fitness function. The software will also combines and mutates genes to find optimized value.

e. Neural Networks

Neural network tools are used to predict future information by learning patterns and then applying them to predict future relationships. Neural networks attempt to learn patterns from data directly by repeatedly examining the data to identify relationships and build a model. They build models by trial and error. The network guesses a value that it compares to the actual number. If the guess is wrong, the model is adjusted. This process involves three iterative steps: predict, compare, and adjust. Neural networks are commonly used in a DSS to classify data and, as noted, to make predictions. [3]

1.1.5.6 Evaluating Development Packages

Whenever it is decided to develop or purchase a knowledge-driven decision support system software application, it is important to consider following criteria:

a. Development Features: Input rules, customizability, capabilities and maintenance**.**

b. Scalability: Ease of integration with other existing hardware and software, web technologies, operating systems.

c. Ease of Use and Installation: The ease with which end user will be able to work on it.

d. Security: Safety of data and company information.

e. Cost: Cost of technology, cost of development, maintenance cost.

Knowledge-driven decision support systems help businesses solve problems and make decisions. However, a caution should be used when employing it. It does not outsmart human intellect; rather it aids decision making.

We illustrate in Figure 1.5 a sport decision support system framework [4].

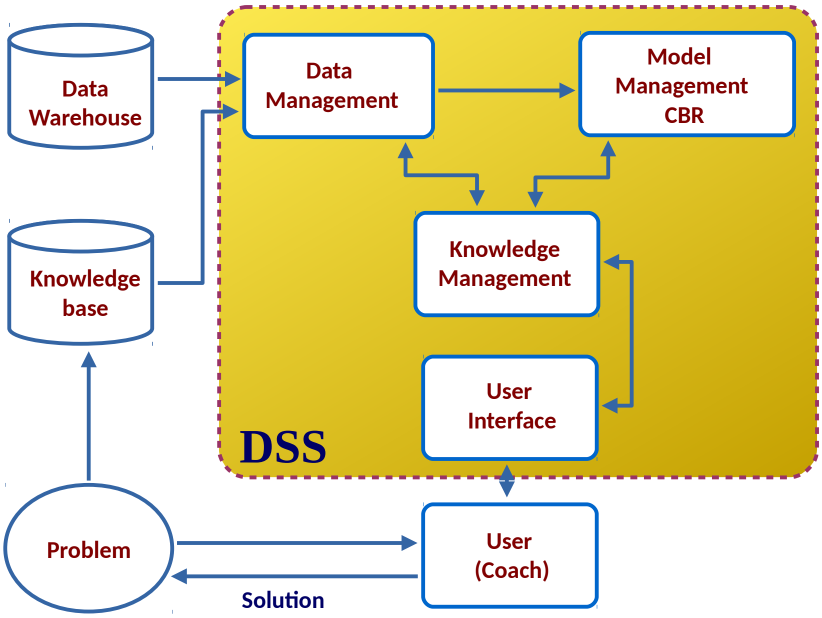


Figure 1.5 DSS conceptual framework.

1.2 Soccer management methods

Soccer coach management methods describe the decision process of a coach for analysing and planning the operations approach used to qualify a team to the competitions. There are three different levels of reviewing this goal: statistical data analysis, prediction method, and afterwards operation planning.

Conventional sport methods labor the most times in drilling plans analysis with habitual stats information of contender teams. This vision generally means a unvarying prospective knowledge and this coach working mostly times gets hard losses.

1.2.1 Stats analysis

FIFA World Cup stats gives a complete analysis for each match played by the two contender teams in competition. Nextly in Figure 1.6 we show a statistical match result of the 2014 championship final contended by Germany and Argentina.

According to previous research and literature relating to soccer prediction [5], we get tactic, planning and attacking as performance indicators in competition matches. Therefore we consider these concepts to model the stats reports, which means: Shots on target, Possession of the ball and Score (Figure 1.6 [6]), Possession of the ball in the attacking third of the pitch, Forwards passes and Passes from defensive to attacking third (Figures 1.7 [7], 1.8 [8]). Passing accuracy is taken from Passing distribution showed on Figure 1.9 [9]. Some description and measurement methods are presented in Table 1.1

2014 FIFA World Cup Brazil ™

Match statistics

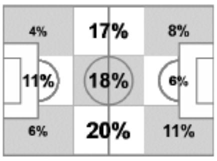
Final

Germany - Argentina 1 : 0 AET

 # 64 13 JUL 2014 16:00 Rio De Janeiro / Estadio do Maracanã / BRA Att: 74,738

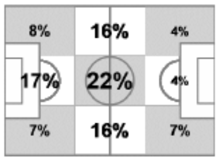
Germany (GER)	Statistics *	Argentina (ARG)
10/7	Shots/Shots on goal	10/2
20	Fouls	16
5	Corner kicks	3
0	Direct free kicks leading to a goal	1
0	Indirect free kicks leading to a goal	0
0/0	Penalty kicks/Converted	0/0
3	Offsides	2
0	Own goals	0
2	Cautions	2
0	Expulsion due to second caution	0
0	Direct expulsions	0
60%	Ball possession	40%
46	Actual playing time	30

Ball possession heat map



Attacking direction

Ball possession heat map



Attacking direction

Budweiser man of the match: 19, Mario GOETZE (Germany)

#	Name	Pos	Min	GF	GA	S/SG	PK	Fouls FC	Fouls FS	Y	2Y=R	R
1	NEUER	GK	120						1			
4	HOEWEDES	DF	120			1 / 0		2		1		
5	HUMMELS	DF	120									
7	SCHWEINSTEIGER	MF	120					3	6	1		
8	OEZIL	MF	119			1 / 1		2	2			
11	KLOSE	FW	88			1 / 1		3	2			
13	MUELLER	FW	120					3	1			
16	LAHM	MF	120						3			
18	KROOS	MF	120			3 / 2		3	1			
20	BOATENG	DF	120									
23	KRAMER	MF	31					1				
Substitutes												
9	SCHUERRLE	FW	89			2 / 1		2				
17	MERTESACKER	DF	1									
19	GOETZE	FW	32	1		2 / 2		1				
Own goals												
Totals				1		10 / 7		20	16	2		

#	Name	Pos	Min	GF	GA	S/SG	PK	Fouls FC	Fouls FS	Y	2Y=R	R
1	ROMERO	GK	120		1							
2	GARAY	DF	120									
4	ZABALETA	DF	120					1	1			
6	BIGLIA	MF	120			1 / 1		2	4			
8	PEREZ	MF	86									
9	HIGUAIN	FW	78			2 / 0		1				
10	MESSI	FW	120			4 / 0		1	2			
14	MASCHERANO	MF	120					2	4	1		
15	DEMICHELIS	DF	120					2	1			
16	ROJO	DF	120					1	1			
22	LAVEZZI	FW	45						3			
Substitutes												
5	GAGO	MF	34					2	3			
18	PALACIO	FW	42			1 / 0		2				
20	AGUERO	FW	75			2 / 1		2		1		
Own goals												
Totals					1	10 / 2		16	19	2		

Pos Position
GK Goalkeeper
MF Midfielder
DF Defender
FW Forward
(C) Captain

Min Minutes played
GF Goals for
GA Goals against
PK Penalty kicks (Goals / Shots)
2Y=R Expulsion due to Second Caution

FC Fouls committed
FS Fouls suffered
Y Single yellow card
R Direct red card

Figure 1.6 Shows the standard statistical information of a final match competition[1].

1 http://resources.fifa.com/mm/document/tournament/competition/02/40/50/17/eng_64_0713_ger-arg_fulltime.pdf

2014 FIFA World Cup Brazil ™

Final

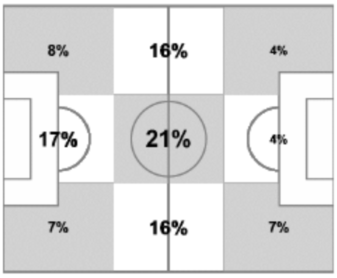
Team Tracking Statistics

Germany - Argentina 1 : 0 AET

64 13 JUL 2014 16:00 Rio De Janeiro / Estadio do Maracanã / BRA

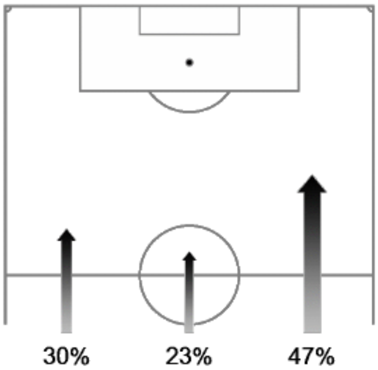
Argentina

Ball possession heat map



Attacking direction

Attack origin



#	Name	Time	Distance covered (metres)			Time spent			Sprints	Top speed	Activity time spent		
			Total	In Poss	Not in Poss	Opp. Half	Att. 3rd	Pen. Area			Low	Medium	High
1	ROMERO	129'09"	4,233	1,294	1,626				6	28.84	98%	1%	1%
2	GARAY	129'09"	12,601	3,545	5,681	9%	4%	2%	37	27.54	87%	6%	7%
4	ZABALETA	129'09"	13,495	4,271	6,071	19%	6%	1%	50	28.15	83%	8%	9%
6	BIGLIA	129'09"	14,681	4,403	**6,901**	35%	9%		50	25.70	78%	10%	12%
8	PEREZ	87'59"	9,775	2,964	4,362	34%	12%	1%	40	31.25	80%	9%	11%
9	HIGUAIN	79'23"	8,972	2,917	3,419	63%	28%	6%	41	30.67	83%	7%	10%
10	MESSI	129'09"	10,721	3,732	4,063	55%	23%	2%	38	30.35	92%	3%	5%
14	MASCHERANO	129'09"	14,037	3,909	6,754	24%	6%	1%	36	28.69	81%	10%	9%
15	DEMICHELIS	129'09"	11,164	3,136	4,888	13%	4%	1%	33	28.87	89%	6%	5%
16	ROJO	129'09"	14,089	4,538	5,891	25%	9%	2%	50	28.30	82%	8%	10%
22	LAVEZZI	47'03"	5,734	1,832	2,495	38%	16%	3%	27	30.53	80%	8%	12%
5	GAGO	41'10"	4,571	1,383	2,093	36%	12%	2%	19	23.94	80%	9%	11%
18	PALACIO	49'45"	6,017	2,011	2,607	57%	31%	6%	30	31.07	79%	9%	12%
20	AGUERO	82'06"	8,307	2,692	3,295	66%	37%	7%	30	31.07	87%	6%	7%

In Poss: Team in possession
Not In Poss: Opposite team in possession

Opp. Half: In opposing half
Att. 3rd: In attaching third
Pen. Area: In penalty area

Bold values: top performing in the match

Figure 1.7 Standard statistical information of the team Argentina describing the zone of attack[2].

2 http://resources.fifa.com/mm/document/tournament/competition/02/40/51/10/64_0713_ger-arg_arg_teamstatistics.pdf

2014 FIFA World Cup Brazil ™

Final

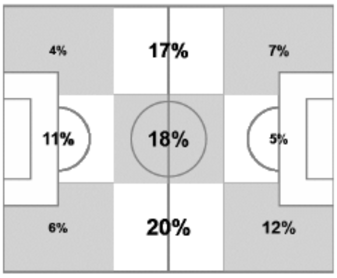
Team Tracking Statistics

Germany - Argentina 1 : 0 AET

 # 64 13 JUL 2014 16:00 Rio De Janeiro / Estadio do Maracanã / BRA

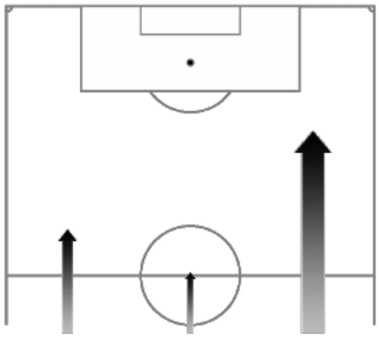
Germany

Ball possession heat map



Attacking direction

Attack origin



#	Name	Time	Distance covered (metres)			Time spent			Sprints	Top speed	Activity time spent		
			Total	In Poss	Not in Poss	Opp. Half	Att. 3rd	Pen. Area			Low	Medium	High
1	NEUER	129'09"	6,985	2,424	2,269	1%			8	30.85	96%	2%	2%
4	HOEWEDES	129'09"	14,115	5,628	4,849	43%	15%	3%	40	27.04	83%	9%	8%
5	HUMMELS	129'09"	12,973	5,043	4,542	21%	6%	2%	35	**31.61**	86%	7%	7%
7	SCHWEINSTEIGER	129'09"	**15,338**	**6,527**	4,906	48%	14%	1%	55	26.14	77%	11%	12%
8	OEZIL	124'58"	14,131	6,153	4,523	67%	35%	4%	58	28.51	80%	8%	12%
11	KLOSE	89'48"	8,942	3,509	2,864	80%	57%	14%	38	26.86	86%	6%	8%
13	MUELLER	129'09"	15,180	6,262	4,729	72%	45%	10%	62	27.22	80%	9%	11%
16	LAHM	129'09"	13,738	5,997	4,621	47%	17%		50	26.28	80%	10%	10%
18	KROOS	129'09"	14,320	6,027	4,807	53%	20%	1%	32	27.97	81%	10%	9%
20	BOATENG	129'09"	13,670	5,481	4,636	28%	2%		53	28.69	83%	9%	8%
23	KRAMER	30'39"	4,107	1,974	1,316	72%	45%	5%	23	28.69	71%	12%	17%
9	SCHUERRLE	98'29"	11,179	4,805	3,572	65%	35%	5%	**65**	30.17	80%	8%	12%
17	MERTESACKER	4'10"	362	60	105				2	24.48	87%	5%	8%
19	GOETZE	39'20"	5,447	2,627	1,781	71%	41%	5%	29	27.97	68%	13%	19%

In Poss: Team in possession
Not In Poss: Opposite team in possession
Opp. Half: In opposing half
Att. 3rd: In attaching third
Pen. Area: In penalty area
Bold values: top performing in the match

Figure 1.8. Standard statistical information of the team Germany describing the zone of attack[3].

3 http://resources.fifa.com/mm/document/tournament/competition/02/40/50/84/64_0713_ger-arg_ger_teamstatistics.pdf

2014 FIFA World Cup Brazil ™

Passing Distribution

Final

Germany - Argentina 1 : 0 AET

 # 64 13 JUL 2014 16:00 Rio De Janeiro / Estadio do Maracanã / BRA

From		TP	To: Manuel NEUER 1	Benedikt HOEWEDES 4	Mats HUMMELS 5	Bastian SCHWEINSTEIGER 7	Mesut OEZIL 8	Miroslav KLOSE 11	Thomas MUELLER 13	Philipp LAHM 16	Toni KROOS 18	Jerome BOATENG 20	Christoph KRAMER 23	Andre SCHUERRLE 9	Per MERTESACKER 17	Mario GOETZE 19	Long PC	Long PA	Medium PC	Medium PA	Short PC	Short PA	Total PC	Total PA	%
Manuel NEUER	1	129'09"		7	4	4	3		3	11	3	5		2			20	25	22	23	0	0	42	48	88%
Benedikt HOEWEDES	4	129'09"	2		8	6	3	1	2		13	2		18			2	4	31	41	22	26	55	71	77%
Mats HUMMELS	5	129'09"	6	9		7	2	2	3	7	9	11					10	10	36	46	10	13	56	69	81%
Bastian SCHWEINSTEIGER	7	129'09"	4	9	8		13	1	8	15	23	9	4	7		1	7	11	66	74	29	33	102	118	86%
Mesut OEZIL	8	124'58"		3		16		1	6	13	15	5	1	3		4	2	4	37	45	28	35	67	84	80%
Miroslav KLOSE	11	89'48"					1		6	2	1	2					3	3	2	11	7	9	12	23	52%
Thomas MUELLER	13	129'09"		1		3	7	2		16	9	3	4	3		3	1	7	28	33	22	30	51	70	73%
Philipp LAHM	16	129'09"	1		2	23	**24**	6	12		7	23		1		5	4	9	66	76	34	37	104	122	85%
Toni KROOS	18	129'09"		12	8	17	10	1	8	11		3	2	19		3	13	17	61	71	20	26	94	114	82%
Jerome BOATENG	20	129'09"	9	3	10	13	8		5	22	6			1		1	10	16	55	62	13	13	78	91	86%
Christoph KRAMER	23	30'39"				1			4	3	1	1					1	1	5	7	4	5	10	13	77%
Andre SCHUERRLE	9	98'29"		7	4	3	7		5	3	17	1				3	4	4	28	41	18	27	50	72	69%
Per MERTESACKER	17	4'10"														1	0	0	1	1	0	0	1	1	100%
Mario GOETZE	19	39'20"					4			5	1	1		3			1	1	5	9	8	9	14	19	74%
Total passes received			22	51	44	93	82	14	62	108	105	66	11	57	0	21	78	112	443	540	215	263	736	915	80%

From		TP	To: Sergio ROMERO 1	Ezequiel GARAY 2	Pablo ZABALETA 4	Lucas BIGLIA 6	Enzo PEREZ 8	Gonzalo HIGUAIN 9	Lionel MESSI 10	Javier MASCHERANO 14	Martin DEMICHELIS 15	Marcos ROJO 16	Ezequiel LAVEZZI 22	Fernando GAGO 5	Rodrigo PALACIO 18	Sergio AGUERO 20	Long PC	Long PA	Medium PC	Medium PA	Short PC	Short PA	Total PC	Total PA	%
Sergio ROMERO	1	129'09"			1	1	2		1	3	5	5				2	8	30	11	11	1	1	20	42	48%
Ezequiel GARAY	2	129'09"	3			3	3		2	7	8	**13**		2		2	8	10	30	35	5	7	43	52	83%
Pablo ZABALETA	4	129'09"	2	1		3	4	5	8	2	10		5	1	1	5	3	8	33	39	11	17	47	64	73%
Lucas BIGLIA	6	129'09"	2	7	3		2	1	10	7	1	3	1	2	2	2	7	9	30	38	6	11	43	58	74%
Enzo PEREZ	8	87'59"		1	3	2		3	2	9	3	3					1	1	16	20	9	14	26	35	74%
Gonzalo HIGUAIN	9	79'23"			1	1	7			3			1			1	0	0	7	10	7	8	14	18	78%
Lionel MESSI	10	129'09"		2	4	4	2	1		4			1	5		5	4	5	8	20	16	21	28	46	61%
Javier MASCHERANO	14	129'09"	3	6	4	8	6		5		9	7	2	4		4	7	12	39	48	12	14	58	74	78%
Martin DEMICHELIS	15	129'09"	3	8	6	4	5		3	6		1	3	6		1	0	0	37	43	9	12	46	55	84%
Marcos ROJO	16	129'09"		6		12	3	3	3	9				2	2	2	3	8	27	36	12	17	42	61	69%
Ezequiel LAVEZZI	22	47'03"			2	1		2	2								0	3	3	10	4	6	7	19	37%
Fernando GAGO	5	41'10"			1	5			3	5		3			1	2	4	4	10	14	6	6	20	24	83%
Rodrigo PALACIO	18	49'45"							2	1						2	0	0	4	6	1	4	5	10	50%
Sergio AGUERO	20	82'06"			1	2		3	3	3		1		1	3		0	0	8	16	9	12	17	28	61%
Total passes received			13	31	26	46	34	18	44	59	36	36	13	23	9	28	45	90	263	346	108	150	416	586	71%

TP: Time played PA: Passes attempted PC: Passes completed %: Passing success percentage

Figure 1.9 Passing distribution of the two teams.

Attacking performance indicators	Description	Measurement
1. Possession of the ball	Percentage of time that the team has possession of the ball in the match.	Possession of the ball for the team was collected separately for each half of the match as it is provided by the Amisco system [10]. The average from the possession of the two halves for each team was calculated. These performance indicators were calculated by taking the overall time that the team had the possession of the ball and the time that the team had the possession of the ball in the area corresponding to the performance indicator. Hence the percentage (normalised data) was calculated from these data provided by the Amisco system.
2. Possession of the ball in the attacking third of the pitch.	Percentage of time that the team have the possession of the ball in the attacking third of the pitch (next to the opposite goal) from all the time that the team have the possession of the ball.	
3. Forwards passes	Percentage of passes from the overall number of passes made by the team that are made forwards (towards the opposite goal).	The Amisco system provided the direction of the movements of the ball by looking at the point where the pass started and the point where the pass was received. Data was normalised by calculating the percentage of these passes according to the total number of passes made by the team.
4. Passes from defensive third to attacking third	Percentage of passes from the overall number of passes made by the team that are made directly from the defensive third (next to the own goal) to the attacking third of the pitch (next to the opposite goal).	These performance indicators were measured by calculating the percentage of these kinds of passes from the overall amount of passes made by the team in the match.

Table 1.1 Description and measurement of attacking and defensive performance indicators [11].

1.3 Methods of prediction

Rating methods is commonly used in mostly cases. It is showed below.

1.3.1 Rating Systems for Fixed Odds Football Match Prediction

It is defined [12], by analysing and comparing one or more aspects of past performance for each of the teams, whilst more complex ratings might be based on elaborate match statistics including shots on goal, corners, and even possession if such data are available. A simple way looks at the number of goals scored and conceded by the two teams for a specified number of matches preceding the contest under examination. A match rating must in some way be translated into a probability distribution for the three possible results in a football match: home win, the draw and away win.

1.3.1.1 A Goals Superiority Rating System

The assumption for a goals superiority rating system, then, is that teams who score more goals and concede fewer over the course of a number of matches are more likely to win their next game.

To see how a goal superiority match rating is calculated, consider the following example for a game played between Tottenham and Leeds at White Hart Lane. In their last 6 games, Tottenham have scored 6 goals and conceded 9. Meanwhile, Leeds have scored 8 times and conceded 11 goals. Tottenham's goal superiority rating for the last 6 games is -3; for Leeds it is also -3. The match rating is simply given by the home side's rating minus the away side's rating, and for this match is therefore 0. Using results data for the English Premiership and Divisions 1, 2, and 3 for seasons 1993/94 to 2000/01, goal supremacy ratings for the last 6 matches played by every team have been calculated. Of the 16,272 matches played during these 8 years, 14,002 of them were eligible for a rating calculation, with the matches played in the earlier weeks of each season obviously unsuitable for recent form analysis. The number of matches with each rating and percentages are shown in the figures below.

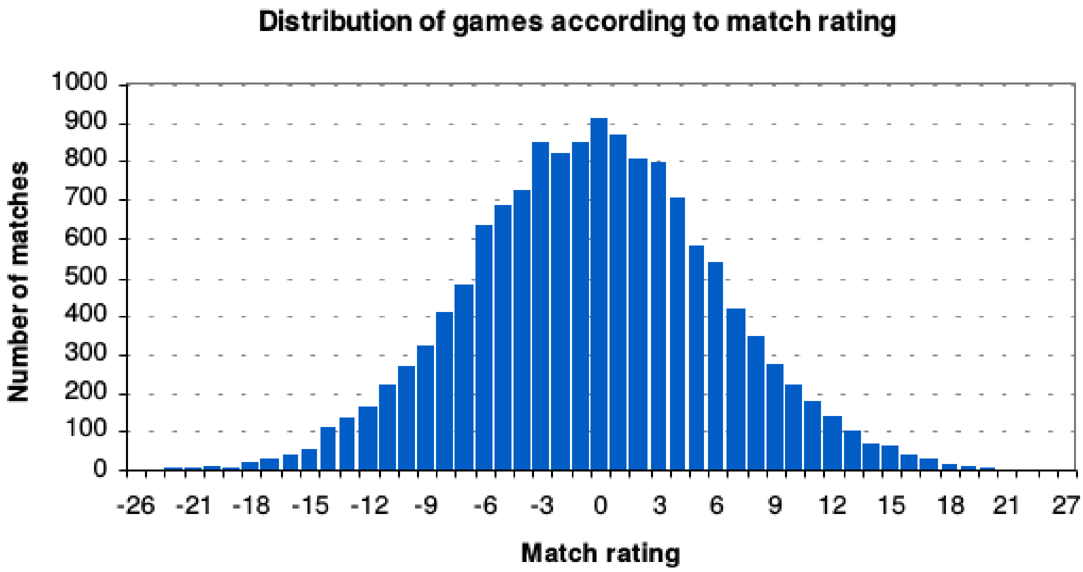


Figure 1.10 Standard normal distribution of matches rating.

1.3.1.2 Defining the Fair Odds

The first task is to consider each result independently, and identify the "best-fit" relationship with the match ratings. The easiest way to determine this best-fit relationship is to draw the match ratings and result probabilities as three scatter plots, one for each result, as shown on Figure 1.11. This can be done simply using any spreadsheet. For each scatter plot, the best fit line (with its equation) has been superimposed on the data points, representing what would statistically be considered to be the best relationship between match rating and result probability.

The value of R^2 shown for each of the three equations is simply a statistical measure of how closely the real data match the best-fit lines. A perfect relationship, denoted by $R^2 = 1$, would mean that the best-fit line and equation describe perfectly the real data. Consequently, a fairly good relationship exists between the match rating and home win probability, where as much as 86% of the variation in the real data is explained by its best-fit equation. For away wins and particularly draws the relationship is weaker. With each equation we can easily determine the expected probability of a home win, draw and away win for any match where we have calculated the goal supremacy rating.

For the Tottenham-Leeds game, where the match rating was 0, the probability of a home win, for example, can be determined using: $y = 1.56x + 46.47$

Match rating	Number of home wins	Number of draws	Number of away wins	% of home wins	% of home draws	% of away wins
-26	0	1	1	0.0%	50.0%	50.0%
-23	0	0	2	0.0%	0.0%	100.0%
-22	0	0	3	0.0%	0.0%	100.0%
-21	0	2	4	0.0%	33.3%	66.7%
-20	2	2	7	18.2%	18.2%	63.6%
-19	1	1	3	20.0%	20.0%	60.0%
-18	5	7	9	23.8%	33.3%	42.9%
-17	7	9	12	25.0%	32.1%	42.9%
-16	6	14	21	14.6%	34.1%	51.2%
-15	25	12	19	44.6%	21.4%	33.9%
-14	32	30	51	28.3%	26.5%	45.1%
-13	43	38	58	30.9%	27.3%	41.7%
-12	51	50	64	30.9%	30.3%	38.8%
-11	75	54	91	34.1%	24.5%	41.4%
-10	84	94	91	31.2%	34.9%	33.8%
-9	123	91	112	37.7%	27.9%	34.4%
-8	171	113	124	41.9%	27.7%	30.4%
-7	190	121	170	39.5%	25.2%	35.3%
-6	242	202	191	38.1%	31.8%	30.1%
-5	279	212	197	40.6%	30.8%	28.6%
-4	293	219	215	40.3%	30.1%	29.6%
-3	374	246	229	44.1%	29.0%	27.0%
-2	372	233	214	45.4%	28.4%	26.1%
-1	375	251	222	44.2%	29.6%	26.2%
0	414	259	235	45.6%	28.5%	25.9%
1	412	243	212	47.5%	28.0%	24.5%
2	401	220	189	49.5%	27.2%	23.3%
3	395	224	175	49.7%	28.2%	22.0%
4	391	177	137	55.5%	25.1%	19.4%
5	297	180	102	51.3%	31.1%	17.6%
6	260	146	131	48.4%	27.2%	24.4%
7	236	98	83	56.6%	23.5%	19.9%
8	197	94	56	56.8%	27.1%	16.1%
9	158	86	32	57.2%	31.2%	11.6%
10	125	57	42	55.8%	25.4%	18.8%
11	113	34	33	62.8%	18.9%	18.3%
12	90	30	22	63.4%	21.1%	15.5%
13	61	23	17	60.4%	22.8%	16.8%
14	48	15	11	64.9%	20.3%	14.9%
15	38	21	8	56.7%	31.3%	11.9%
16	30	9	2	73.2%	22.0%	4.9%
17	20	8	2	66.7%	26.7%	6.7%
18	15	1	1	88.2%	5.9%	5.9%
19	8	4	1	61.5%	30.8%	7.7%
20	5	1	0	83.3%	16.7%	0.0%
21	1	0	0	100.0%	0.0%	0.0%
22	1	0	1	50.0%	0.0%	50.0%
23	1	0	0	100.0%	0.0%	0.0%
27	1	0	0	100.0%	0.0%	0.0%
Total	**6468**	**3932**	**3602**	**46.2%**	**28.1%**	**25.7%**

Table 1.2 Goal supremacy match ratings and historical result percentages.

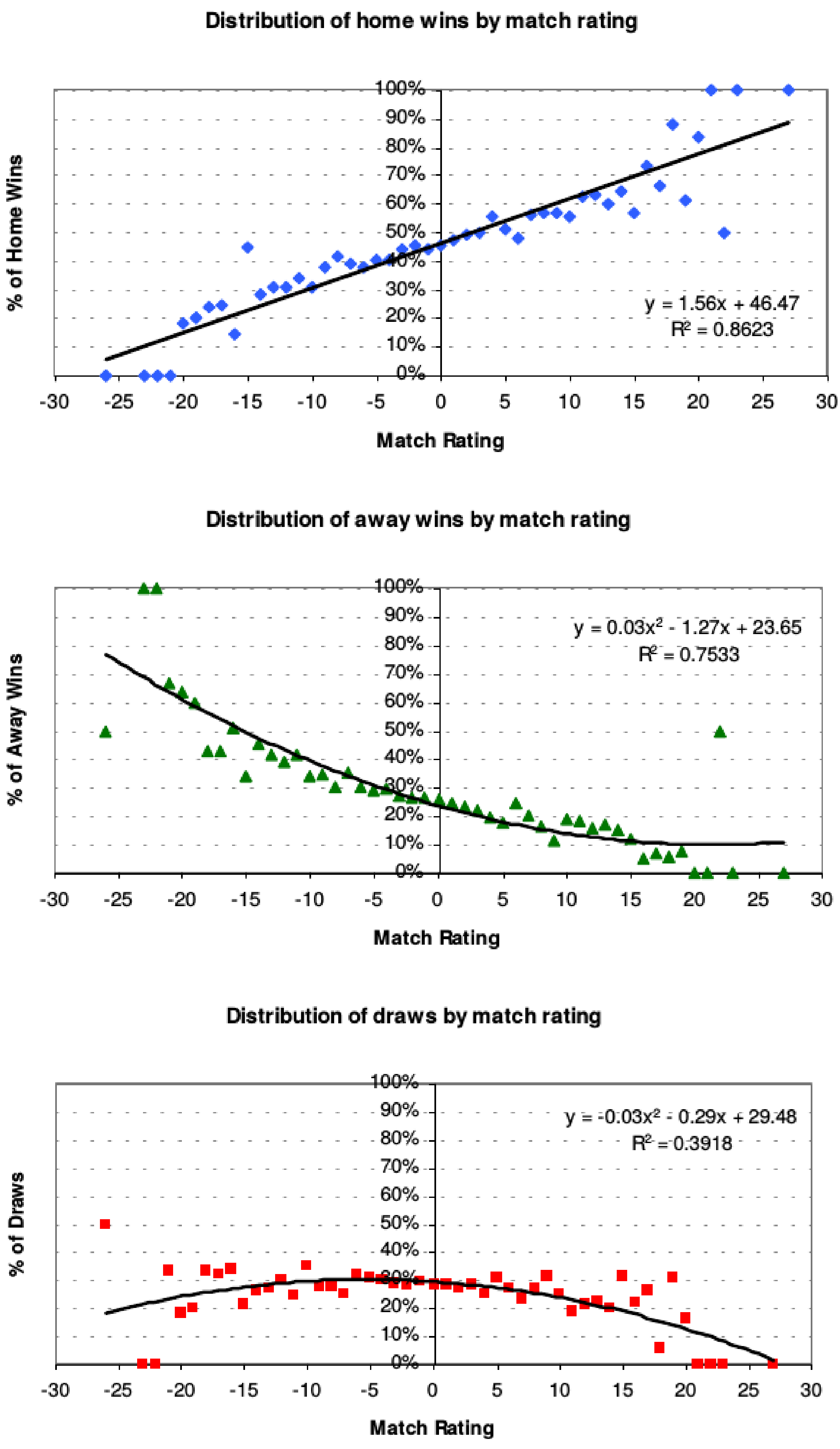


Figure 1.11 Distribution of classes: Home wins, Away wins and Draws.

With an estimation of the probability or expectancy for a home win we can easily define the fair odds for a home win, using:

100 divided by probability of home win

Fair odds for some other match ratings are show tabulated below.

Result expectancies and their fair odds for a goal supremacy rating system

Match rating	Home win expectancy	Draw expectancy	Away win expectancy	Total	Fair home odds	Fair draw odds	Fair away odds
-16	22%	26%	52%	100%	4.65	3.78	1.94
-14	25%	28%	47%	100%	4.06	3.62	2.11
-12	28%	29%	43%	100%	3.60	3.49	2.31
-10	31%	29%	39%	100%	3.24	3.40	2.54
-8	34%	30%	36%	100%	2.94	3.35	2.80
-6	37%	30%	32%	100%	2.69	3.32	3.09
-4	40%	30%	29%	100%	2.49	3.32	3.42
-2	43%	30%	26%	100%	2.31	3.34	3.80
0	46%	29%	24%	100%	2.15	3.39	4.23
2	50%	29%	21%	100%	2.02	3.47	4.71
4	53%	28%	19%	100%	1.90	3.59	5.25
6	56%	27%	17%	100%	1.79	3.75	5.84
8	59%	25%	15%	100%	1.70	3.96	6.49
10	62%	24%	14%	100%	1.61	4.24	7.17
12	65%	22%	13%	100%	1.53	4.61	7.86
14	68%	20%	12%	100%	1.46	5.12	8.51
16	71%	17%	11%	100%	1.40	5.83	9.08

Table 1.3 Fair odds expectancies.

This method could be used for analogous prediction analysis and therefore it can be performed for national soccer team tournament predictions.

1.3.2 Artificial neural networks learning

1.3.2.1 Biological neural network

In neuroscience [13], a biological neural network is a series of interconnected neurons whose activation defines a recognizable linear pathway. The interface through which neurons interact with their neighbors usually consists of several axon terminals connected via synapses to dendrites on other neurons. If the sum of the input signals into one neuron surpasses a certain threshold, the neuron sends an action potential at the axon hillock and transmits this electrical signal along the axon.

Biological neural networks have inspired the design of artificial neural networks.

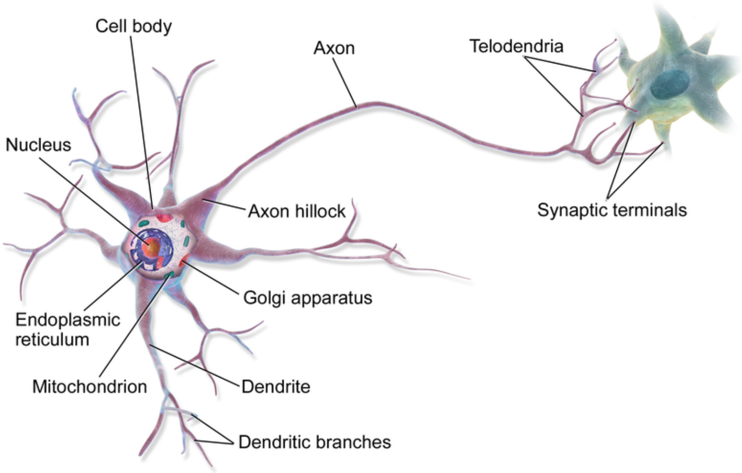


Figure 1.12 Anatomy of a multipolar neuron.

The first rule of neuronal learning was described by Hebb in 1949, Hebbian learning. Thus, Hebbian pairing of pre-synaptic and post-synaptic activity can substantially alter the dynamic characteristics of the synaptic connection and therefore facilitate or inhibit signal transmission. The neuroscientists Warren Sturgis McCulloch and Walter Pitts published the first works on the processing of neural networks. They showed theoretically that networks of artificial neurons could implement logical, arithmetic, and symbolic functions. Simplified models of biological neurons were set up, now usually called perceptrons or artificial neurons.

1.3.2.2 Artificial neural network

Artificial neural networks (ANNs) or connectionist systems are computing systems inspired by the biological neural networks that constitute animal brains [14]. Such systems learn (progressively improve their ability) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the analytic results to identify cats in other images. They do this without any a priori knowledge about cats, e.g., that they have fur, tails, whiskers and cat-like faces. Instead, they evolve their own set of relevant characteristics from the learning material that they process. They have found most use in applications difficult to express with a traditional computer algorithm using rule-based programming.

An ANN is based on a collection of connected units or nodes called artificial neurons (a simplified version of biological neurons in an animal brain). Each connection (a simplified version of a synapse) between artificial neurons can transmit a signal from one to another. The artificial neuron that receives the signal can process it and then signal artificial neurons connected to it.

In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is calculated by a non-linear function of the sum of its inputs. Artificial neurons and connections typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that only if the aggregate signal crosses that threshold is the signal sent. Typically, artificial neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer, possibly after traversing the layers multiple times.

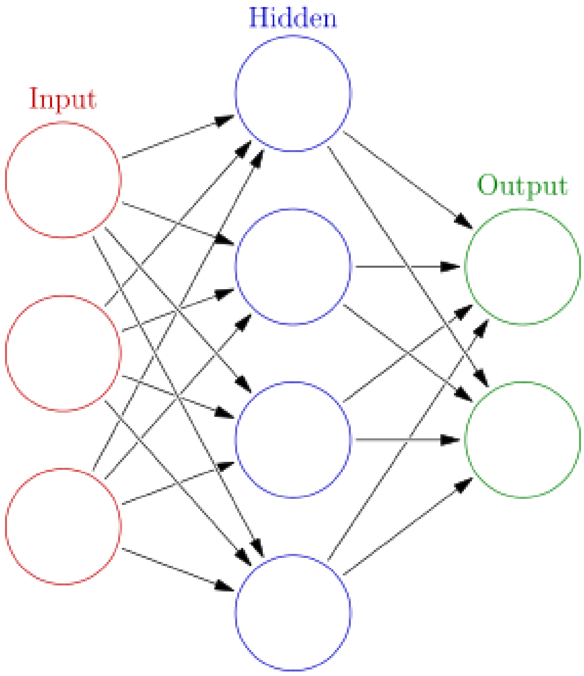


Figure 1.13 An artificial neural network, akin to the vast network of neurons in a brain. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention focused on matching specific tasks, leading to deviations from biology. ANNs have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

1.3.2.2.1 Learning paradigms

The three major learning paradigms applied to ANNs correspond to a particular learning task. These are supervised learning, unsupervised learning (competitive learning) and reinforcement learning.

a. Supervised learning

It is the most widely used where the network learns to recognize a set of desired input configurations [15]. It learns to associate a set of given pairs (X_k,Y_{dk}) (Figure 1.14). The network operates in two distinct phases:

• **Learning phase –** it stores the desired information via examples.

• **Evolution phase –** it retrieves the stored information.

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs [16]. It infers a function from labeled training data consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.

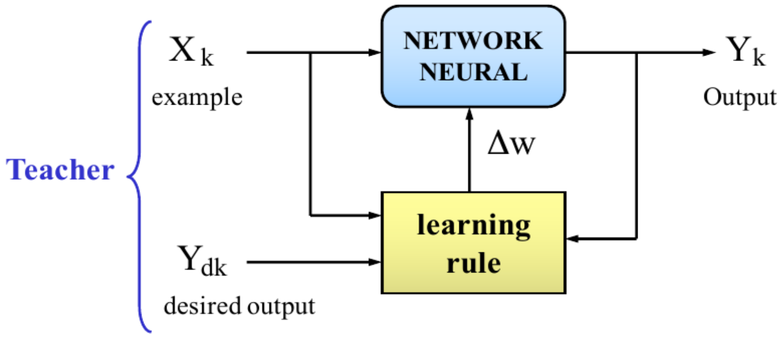


Figure 1.14 Training phase of ANN.

b. Competitive learning

• Neurons compete for specializing in the recognition of a particular stimulus. Similar stimulus end up in the same class. In the end, each neuron is activated by a given stimulus (isomorphism between stimuli and output neurons).

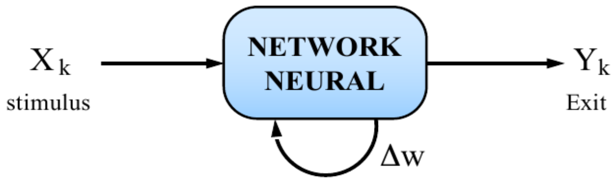


Figure 1.15 Competitive learning of ANN.

c. Reinforcement learning

• Reinforcement learning simulates the learning mechanism in animals based on reward and punishment: used for control systems applications.

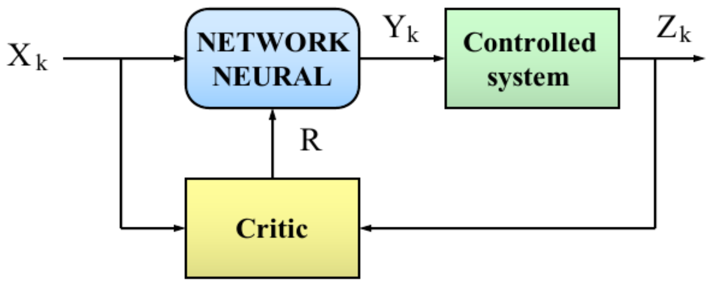


Figure 1.16 Reinforcement learning of ANN.

1.3.2.2.2 Perceptron

In machine learning [17], the perceptron is an algorithm for supervised learning of binary classifiers (functions that can decide whether an input, represented by a vector of numbers, belongs to some specific class or not). It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. The algorithm allows for online learning, in that it processes elements in the training set one at a time.

The perceptron algorithm dates back to the late 1950s. Its first implementation, in custom hardware, was one of the first artificial neural networks to be produced. Figure 1.17 represent this abstraction.

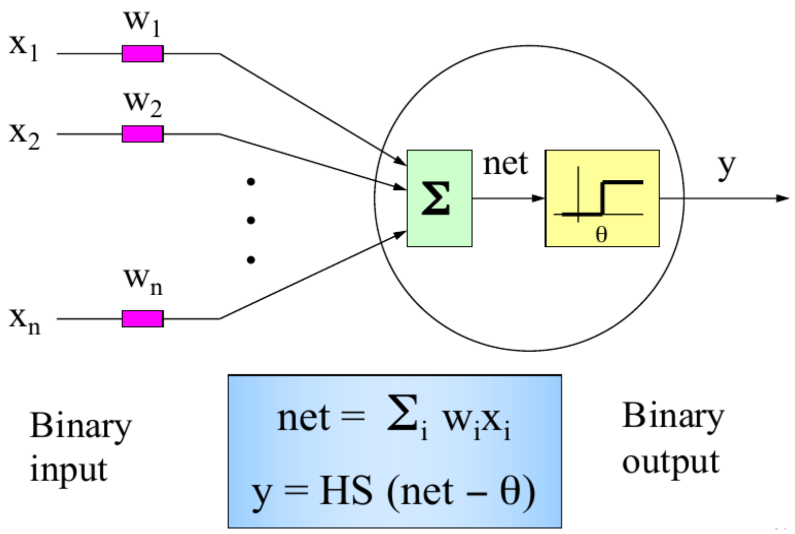


Figure 1.17 The Perceptron (Rosenblatt '58)

1.3.2.2.3 Deep learning

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised. [18]

Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimulus and associated neuronal responses in the brain.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics and drug design, where they have produced results comparable to and in some cases superior to human experts.

Deep learning is a class of machine learning algorithms that:

- use a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.
- learn in supervised (e.g., classification) and/or unsupervised (e.g., pattern analysis) manners.
- learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

a. Concepts

The assumption underlying distributed representations is that observed data are generated by the interactions of layered factors.

Deep learning adds the assumption that these layers of factors correspond to levels of abstraction or composition. Varying numbers of layers and layer sizes can provide different degrees of abstraction.

Deep learning exploits this idea of hierarchical explanatory factors where higher level, more abstract concepts are learned from the lower level ones.

Deep learning architectures are often constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features improve performance.

For supervised learning tasks, deep learning methods obviate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures that remove redundancy in representation.

Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabeled data are more abundant than labeled data. Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief networks.

b. Interpretations

Deep neural networks are generally interpreted in terms of the universal approximation theorem [19] or probabilistic inference [20].

The universal approximation theorem concerns the capacity of feed forward neural networks with a single hidden layer of finite size to approximate continuous functions. In 1989, the first proof was published by Cybenko for sigmoid activation functions and was generalised to feed-forward multi-layer architectures in 1991 by Hornik.

Universal approximation properties and depth: Feed forward network with at least one hidden layer provide a universal approximation framework [21].

The probabilistic interpretation derives from the field of machine learning. It features inference, as well as the optimization concepts of training and testing, related to fitting and generalization, respectively. More specifically, the probabilistic interpretation considers the activation nonlinearity as a cumulative distribution function. The probabilistic interpretation led to the introduction of dropout as regularizer in neural networks. The probabilistic interpretation was introduced by researchers including Hopfield, Widrow and Narendra and popularized in surveys such as the one by Bishop.

c. Deep neural networks

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers [22]. DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.

Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets.

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. Recurrent neural networks (RNNs), in which data can flow in any direction, are used for applications such as language modeling. Long short-term memory is particularly effective for this use.

Convolutional deep neural networks (CNNs) are used in computer vision. CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).

d. Multilayer perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network [23]. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

d.1 Activation function

If a multilayer perceptron has a linear activation function in all neurons, that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model. In MLPs some neurons use a nonlinear activation function that was developed to model the frequency of action potentials, or firing, of biological neurons.

The two common activation functions are both sigmoids, and are described by

$$y(v_i) = \tanh(v_i) \text{ and } y(v_i) = (1 + e^{-v_i})^{-1}$$

The first is a hyperbolic tangent that ranges from -1 to 1, while the other is the logistic function, which is similar in shape but ranges from 0 to 1. Here y_i is the output of the ith node (neuron) and v_i is the weighted sum of the input connections.

d.2 Layers

The MLP consists of three or more layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes making it a deep neural network. Since MLPs are fully connected, each node in one layer connects with a certain weight w_{ij} to every node in the following layer.

d.3 Learning

Learning occurs in the perceptron by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning, and is carried out through backpropagation, a generalization of the least mean squares algorithm in the linear perceptron.

We represent the error in output node j in the nth data point (training example) by

$$e_j(n) = d_j(n) - y_j(n)$$

where d is the target value and y is the value produced by the perceptron. The node weights are adjusted based on corrections that minimize the error in the entire output, given by

$$\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n)$$

Using gradient descent, the change in each weight is

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial v_j(n)} y_i(n)$$

where y_i is the output of the previous neuron and η is the learning rate, which is selected to ensure that the weights quickly converge to a response, without oscillations.

The derivative to be calculated depends on the induced local field v_j , which itself varies. It is easy to prove that for an output node this derivative can be simplified to

$$-\frac{\partial \varepsilon(n)}{\partial v_j(n)} = e_j(n)\,\phi'(v_j(n))$$

where ϕ' is the derivative of the activation function described above, which itself does not vary. The analysis is more difficult for the change in weights to a hidden node, but it can be shown that the relevant derivative is

$$-\frac{\partial \varepsilon(n)}{\partial v_j(n)} = \phi'(v_j(n))\sum_k -\frac{\partial \varepsilon(n)}{\partial v_k(n)} w_{kj}(n)$$

This depends on the change in weights of the kth nodes, which represent the output layer. So to change the hidden layer weights, the output layer weights change according to the derivative of the activation function, and so this algorithm represents a backpropagation of the activation function.

1.3.3 Logic learning machine (LLM)

Machine learning method based on the generation of intelligible rules [24]. LLM is an efficient implementation of the Switching Neural Network (SNN) paradigm, developed by Marco Muselli, Senior Researcher at the Italian National Research Council CNR-IEIIT in Genoa. Logic Learning Machine is implemented in the Rulex suite.

LLM has been employed in different fields, including orthopaedic patient classification, DNA microarray analysis and Clinical Decision Support System.

1.3.3.1 General

Like other machine learning methods, LLM uses data to build a model able to perform a good forecast about future behaviors. LLM starts from a table including a target variable (output) and some inputs and generates a set of rules that return the output value corresponding to a given configuration of inputs. A rule is written in the form:

$$\textbf{if}\ premise\ \textbf{then}\ consequence$$

where consequence contains the output value whereas premise includes one or more conditions on the inputs. According to the input type, conditions can have different forms:

- for categorical variables the input value must be in a given subset:

$$x_1 \in \{A, B, C, \ldots\}$$

- for ordered variables the condition is written as an inequality or an interval:

$$x_2 \leq \alpha \qquad \text{or} \qquad \beta \leq x_3 \leq \gamma$$

A possible rule is therefore in the form:

$$\textbf{if}\ x_1 \in \{A, B, C, \ldots\}\ \text{AND}\ x_2 \leq \alpha\ \text{AND}\ \beta \leq x_3 \leq \gamma\ \textbf{then}\ y = \bar{y}$$

1.3.3.2 Types

According to the output type, different versions of Logic Learning Machine have been developed:

- Logic Learning Machine for classification, when the output is a categorical variable, which can assume values in a finite set.

- Logic Learning Machine for regression, when the output is an integer or real number.

1.3.3.3 Logic rules learning and neural networks learning

Combining deep neural networks with structured logic rules is desirable to harness flexibility and reduce uninterpretability of the neural models. The cognitive process of human beings have indicated that people learn not only from concrete examples (as DNNs do) but also from different forms of general knowledge and rich experiences (Minksy, 1980 [25]; Lake et al., 2015 [26]). Logic rules provide a flexible declarative language for communicating high-level cognition and expressing structured knowledge [27].

The neural network output like a prediction, can be transmitted to the rules set. Consequently logic rules enable the formulation of prototypical linguistic rules of a logic model that can easily be implemented and to do so, knowledge is represented by IF-THEN linguistic rules having the general form:

If X1 is A1 AND X2 is A2 ... AND Xm is Am THEN Y is B;

where X1. . .Xm are linguistic input variables with linguistic values A1, . . ., Am, respectively and where Y is the linguistic output variable with linguistic value B.

The next Figures 1.18 and 1.19 show an example of a rule learning space with an algorithm [28].

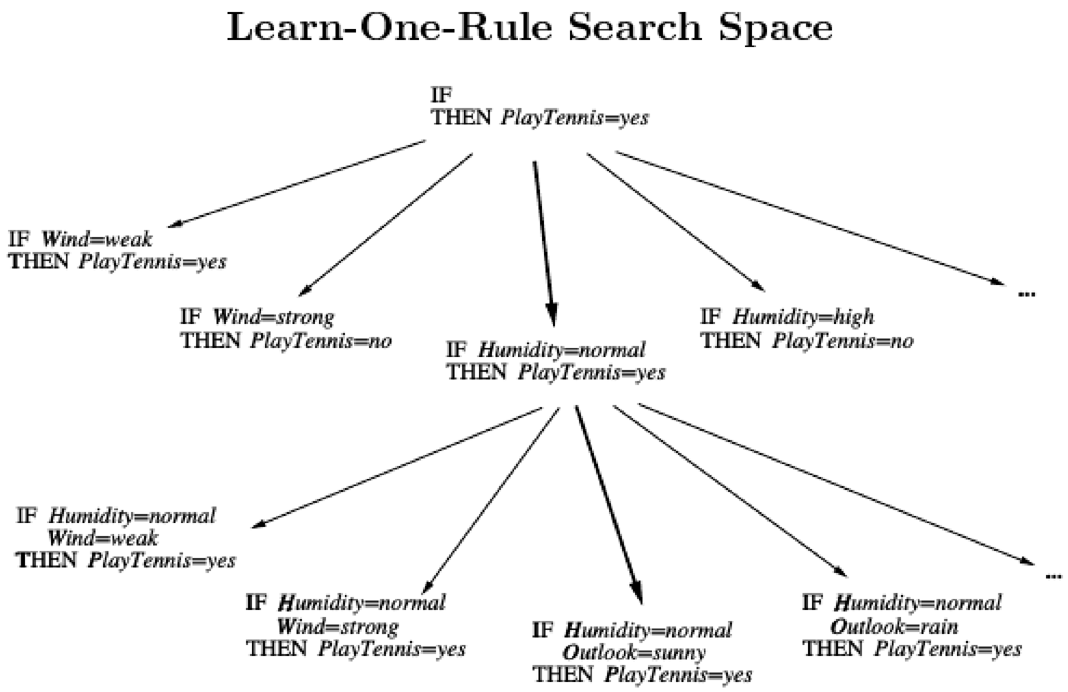


Figure 1.18 Searching space of a learning rule.

LEARN-ONE-RULE($Target\_attr, Attrs, Examples$)

- $Pos \leftarrow$ positive $Examples$; $Neg \leftarrow$ negative $Examples$
- If Pos

  $NewRule \leftarrow$ most general rule possible; $NewRuleNeg \leftarrow Neg$

  While $NewRuleNeg$

  1. $Candidate\_literals(CLs) \leftarrow$ generate candidates
  2. $Best\_literal \leftarrow argmax_{L \in CLs}$ $PERFORMANCE(Specialize(NewRule, L))$
  3. add $Best\_literal$ to $NewRule$ preconditions
  4. $NewRuleNeg \leftarrow$ subset of $NewRuleNeg$ that satisfies $NewRule$ preconditions
- Return $NewRule$

Figure 1.19 A sequential covering algorithm.

Chapter 2

Soccer coach decision support system scenario

Methods of planning for upcoming tournaments are frequently done analising the points, the current position in the classification table and past video competitions. Examining all the soccer mass media, soccer clubs and national soccer teams we discovered that the sport model approach is changing quickly with the employment of technological solutions.

Custom techniques in sport management, physical and mental health, fitness, progress, training, match enactment, competitor analysis and realization are all being revolutionized by the sport digitization to magnify the athlete performance in such manner and triumph.

The method of the soccer coach decision support system (SCDSS) firstly predicts the eventual match output utilizing a multilayer perceptron with a hidden layer. It is trained with all results and scores data of trustly sources, we linked the following soccer site [29]. The implementation integrated the java API open source code of weka multilayer perceptron [30].

The second phase was to get the principal feature stats that according to the literature and media information they influence the high performance and success in a world or continental cup competition. Our research was focused in the website of the FIFA World Cup archives [31].

2.1 Stats analysis of the SCDSS

With the features defined in Table 1.1 of the previous chapter and adding to them other ones, as a product of our examinations, we reached to the following concepts:

2.1.1 Possession in the attacking third of the pitch

Percentage of time that the team have the possession of the ball in the attacking third of the pitch (next to the opposite goal) from all the time that the team have the possession of the ball. Indicator was calculated by taking the overall time that the team had the possession of the ball and the time that the team had the possession of the ball in the area corresponding to the performance indicator.

$$\frac{TimePoss.third}{TimePoss.} * 100$$

2.1.2 Forwards passing

Percentage of passes from the overall number of passes made by the team that are made forwards (towards the opposite goal).

$$\frac{ForwPasses}{TotalPasses} * 100$$

2.1.3 Passing from defensive to attacking third

Percentage of passes from the overall number of passes made by the team that are made directly from the defensive third (next to the own goal) to the attacking third of the pitch (next to the opposite goal). This performance indicator was measured by calculating the percentage of these kinds of passes from the overall amount of passes made by the team in the match for all the matches played.

$$\frac{AttackPasses}{TotalPasses} * 100\%$$

2.1.4 Possession for match

Percentage of time that the team has the ball possession in the match for all matches played.

$$\frac{TimePossession}{TimeOfMatch} * 100\%$$

2.1.5 Passing accuracy for match

Kicking the ball from one player to another player on the same side [32], and putting the ball where you want to go [33]**.** This performance indicator was measured by counting the passes attempted and passes completed and then it is obtained the passing success percentage for each team in the match for all the matches played.

$$\frac{PassesCompleted}{PassesAttempted} * 100\%$$

2.1.6 Shots on target for match

Average of the number of goals that a player or team would have scored if the defending team had not got in the way. This stats is for all the matches played.

2.1.7 Score (a goal)

The ball put into the opposition goal [34].

We can see a panoramic pitch in the Figure 2.1 that illustrates our concepts [11].

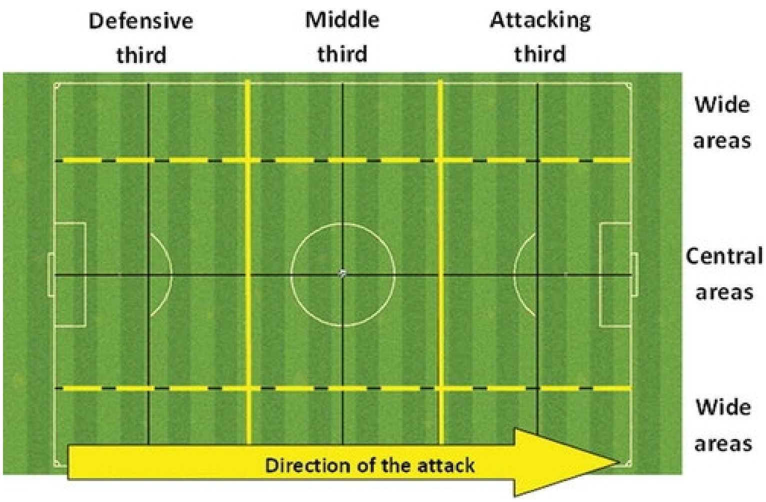


Figure 2.1 Pitch divisions in three thirds parallel to the goal lines and parallel to the touchlines.

2.2 Data driven neural network prediction

The system is determined by using multilayer perceptron and logic rules learning. The MLP is used to generate the prediction of who would win the match. Then this output is taken and analised by logic learning rules to elaborate the final planning of drill and training operations. The system input variables comprise feature stats filtered and refined from the websites: SOCCER-DB.info|Football Database [35], and Fifa World Cup [36]. Our first phase begins getting the data from the first site which will be the input of the MLP. The second site gives us the data stats analysis of each team. Then the second phase of the system makes up the actions of planning and training. Our system model is founded on the literature and the current sport mass media information. The Figure 2.2 shows the design of the proposed system.

Our data carefully verified, extracted and refined was loaded into the **soccerdb** database implemented in mysql wich is the system database application. In Figure 2.3 we show the file **matchevent2.arff** as the input data of the MLP for the match event predictions; being a result of a mysql query inside our eclipse java project. The input attributes are **name**, **name1**, **scoreT1**, **scoreT2** and the prediction attribute class is **namewin** which represents the *winner* or otherwise a *draw*. This arff file is being kept up to date through the updated data and the execution of pertinent mysql queries within the decision support system.

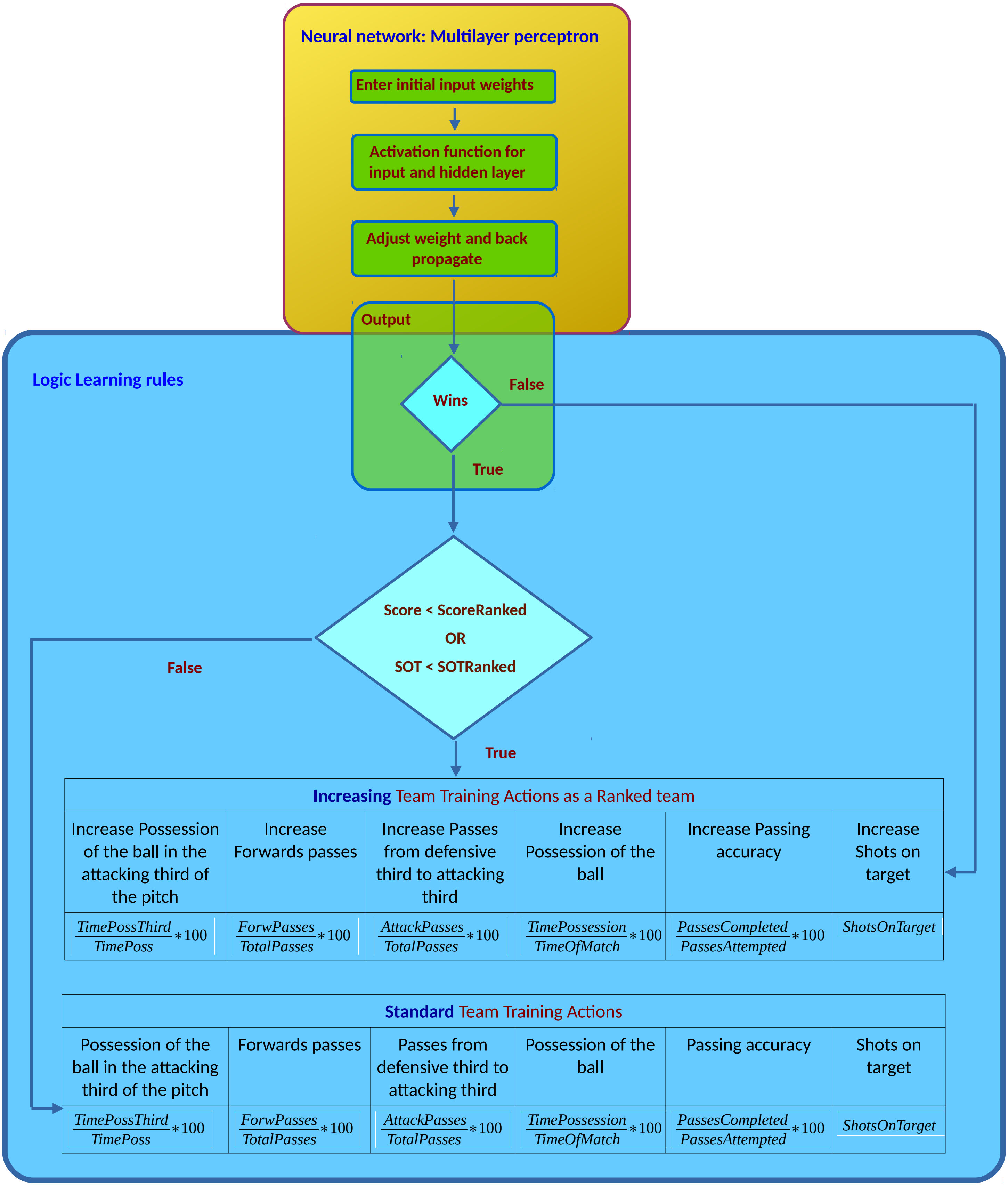
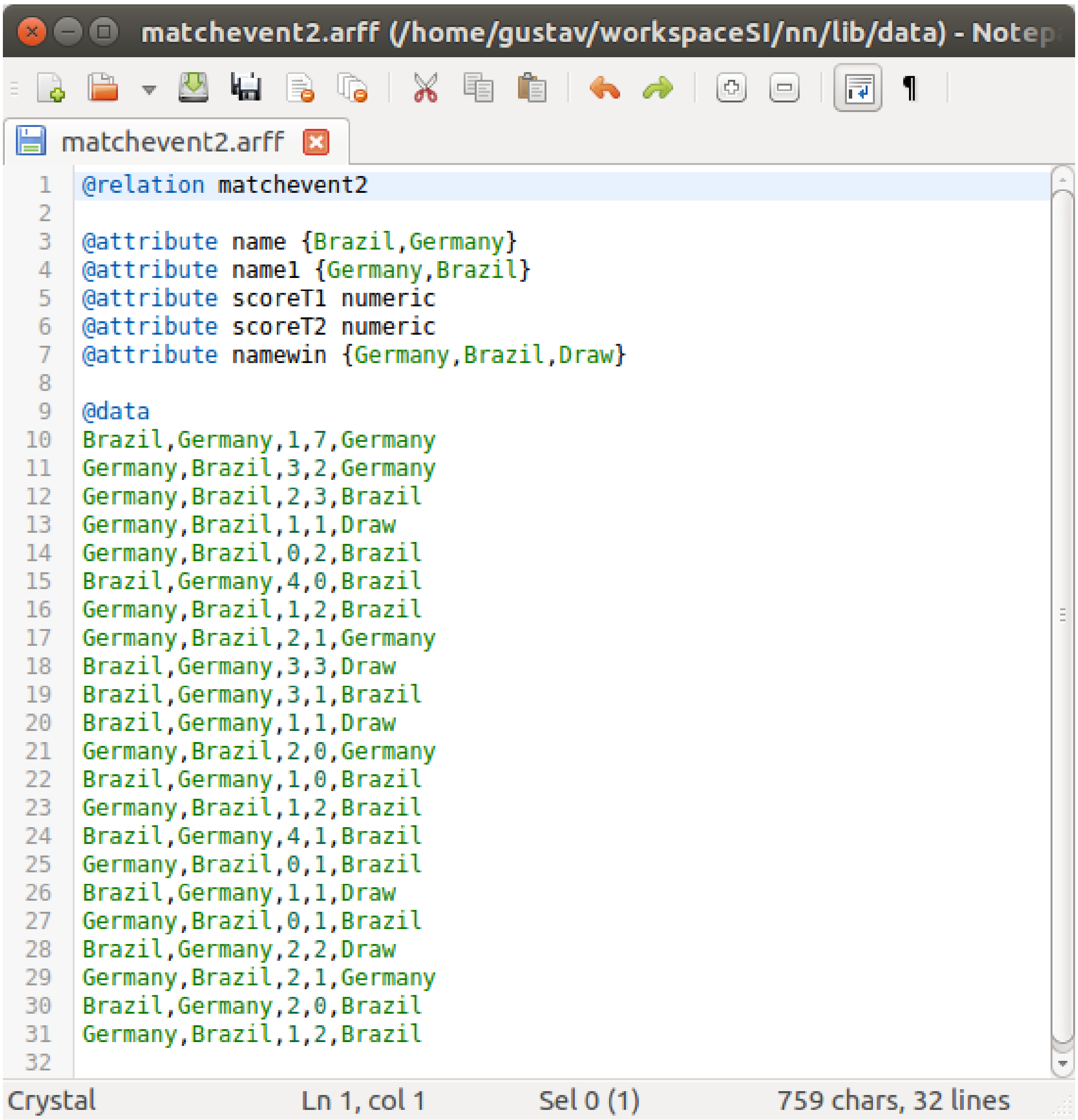


Figure 2.2 A diagrammatic representation of the proposed system framework. **SOT**: Avg Shots on target, **SOTRanked**: Avg Shots on target of a ranked team, **Score**: Avg Score, **ScoreRanked**: Avg Score of a ranked team




```
@relation matchevent2

@attribute name {Brazil,Germany}
@attribute name1 {Germany,Brazil}
@attribute scoreT1 numeric
@attribute scoreT2 numeric
@attribute namewin {Germany,Brazil,Draw}

@data
Brazil,Germany,1,7,Germany
Germany,Brazil,3,2,Germany
Germany,Brazil,2,3,Brazil
Germany,Brazil,1,1,Draw
Germany,Brazil,0,2,Brazil
Brazil,Germany,4,0,Brazil
Germany,Brazil,1,2,Brazil
Germany,Brazil,2,1,Germany
Brazil,Germany,3,3,Draw
Brazil,Germany,3,1,Brazil
Brazil,Germany,1,1,Draw
Germany,Brazil,2,0,Germany
Brazil,Germany,1,0,Brazil
Germany,Brazil,1,2,Brazil
Brazil,Germany,4,1,Brazil
Germany,Brazil,0,1,Brazil
Brazil,Germany,1,1,Draw
Germany,Brazil,0,1,Brazil
Brazil,Germany,2,2,Draw
Germany,Brazil,2,1,Germany
Brazil,Germany,2,0,Brazil
Germany,Brazil,1,2,Brazil
```

Figure 2.3 Data input of the weka multilayer perceptron obtained from the *soccerdb* through the mysql queries.

2.2.1. Weka API

We can see an instance of multilayer perceptron in Figure 2.4. The neurons used in Weka are unipolar sigmoids [37], as the following expression valuations:

$$\boxed{\begin{aligned} y &= logsig\left(\sum_i w_i x_i - \theta\right) \\ logsig(x) &= \frac{1}{1+e^{-x}} \end{aligned}}$$

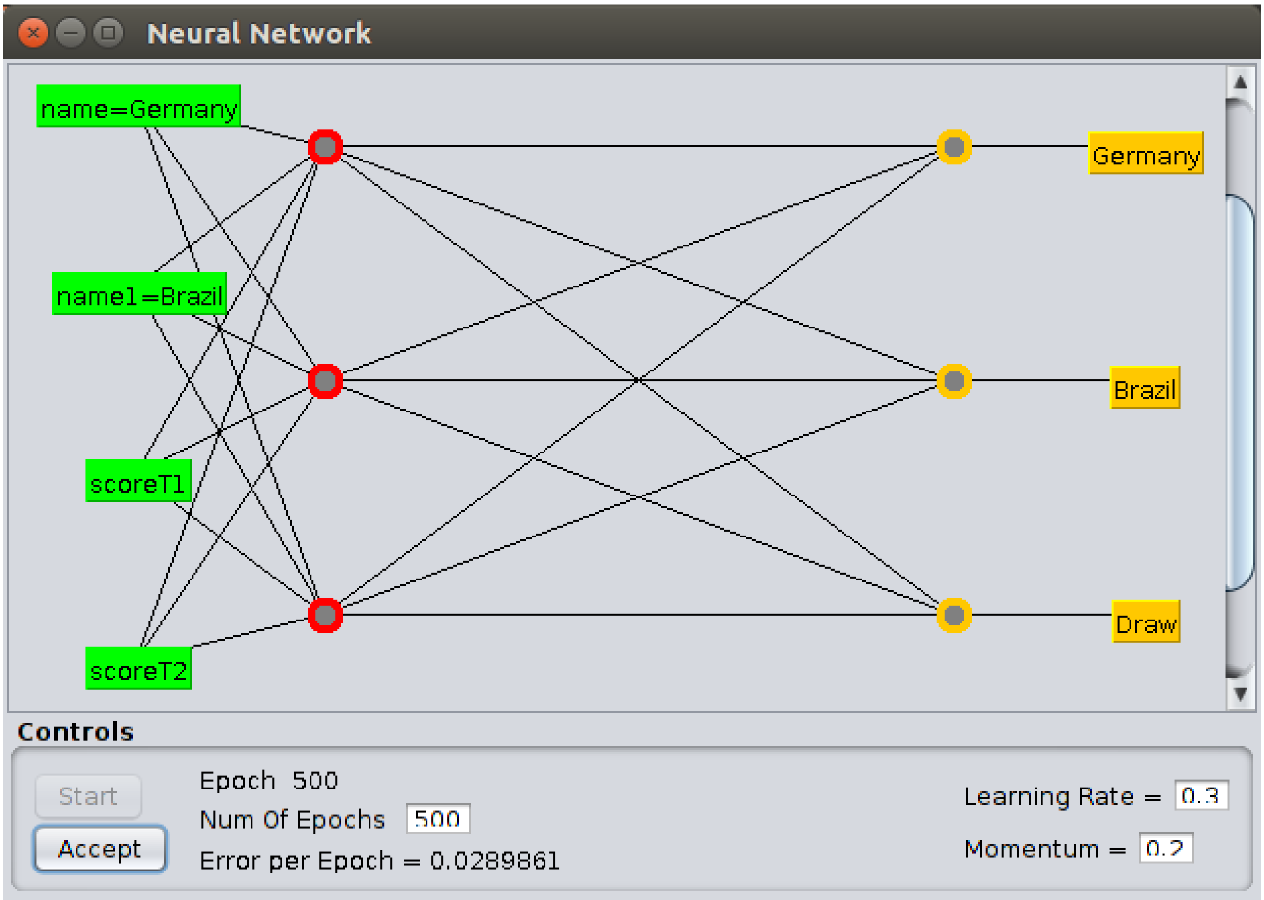


Figure 2.4 Example of multilayer perceptron in weka with a hidden layer near to the output classes.

2.3 Drill plan method of the SCDSS

This component of the system was constructed having a considerable soccer sport literature [38], additionally it was an effort to get openly reliable and generous mass media information [39]. It has been an assiduous understanding and interpretation of the stats analysis features, each one was seen at 2.1.1 and afterward, the studies revealed many kinds of soccer drills and trainings for every feature that we named feature drills. The lines below expose some of our correlated drills.

2.3.1 Possession in attacking 3rd pitch

2.3.1.1 Pulling along defenders

Example of good mobility [40]. In Figure 2.6, attacker A2 has the ball. Attacker A1, the right winger, makes a checking run and moves back toward the ball, pulling along defender D1, the left back. This creates space in the right corner of the field. Attacker A3 moves into this space and D3 follows. This leaves space in front of the goal. Attacker A4 running from midfield receives the ball from A2 with the opportunity to shoot on goal. A1 and A3 made runs to open up space for A4.

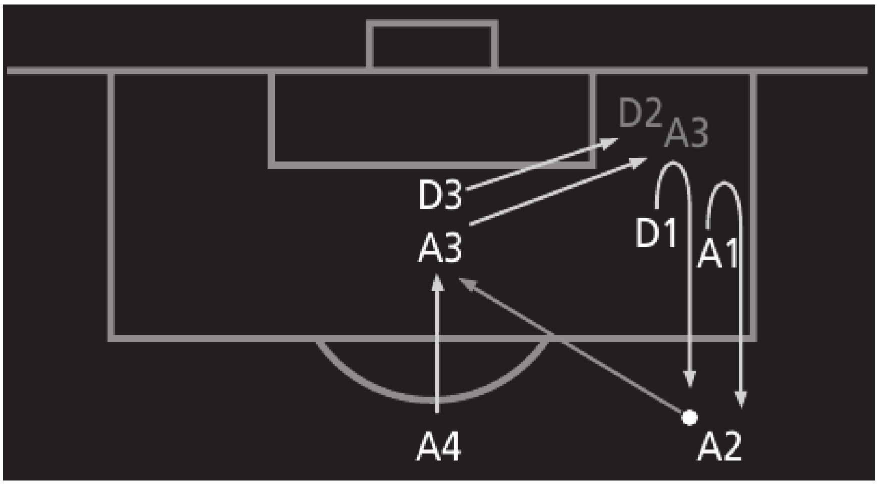


Figure 2.5 Drill design of **Pulling along defenders** into the decision support system.

2.3.1.2 Central penetrating run

Example of a good penetrating run. Attacker A1 has the ball, and A2 is marked by defender D2. A2 makes a checking run back toward A1, creating space to the left of the goalkeeper. Left side Attacker A4, makes a diagonal run toward the left sideline. This will draw defender D3 toward A4, opening up the middle for a penetrating run by A5.

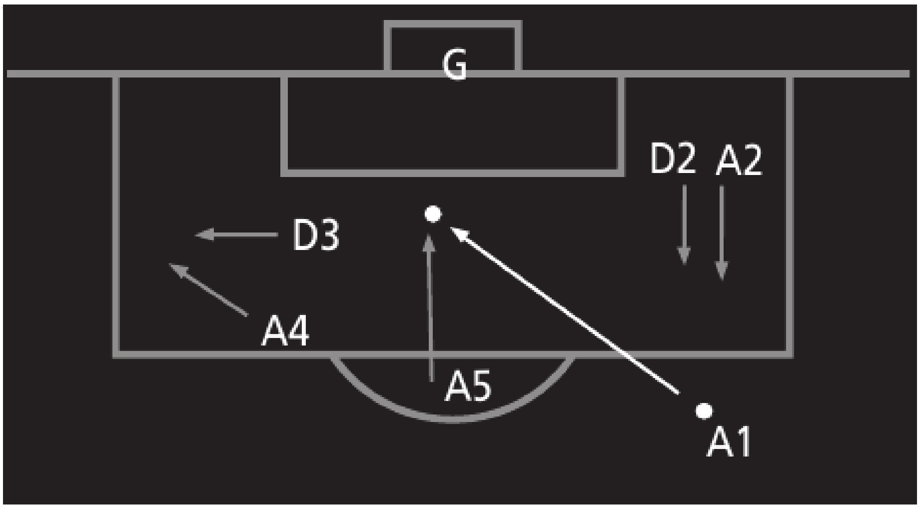


Figure 2.6 Drill design of **Central penetrating run** into the decision support system.

2.3.1.3 Fullback crossing

Two servers, 2 crossing players and 2 players in the middle to get on the end of the cross. Large supply of balls. The game starts on the left wing (Figure 2.8). The first server plays the fullback in, who crosses first time into the penalty area. The crosses must be high quality (e.g. of the right pace and height). No. 9 must time the run to get in a shot or header on goal. Once the action has finished, the second server plays the ball to no. 2, who crosses for no. 8. The players in the middle can switch sides after 5 crosses.

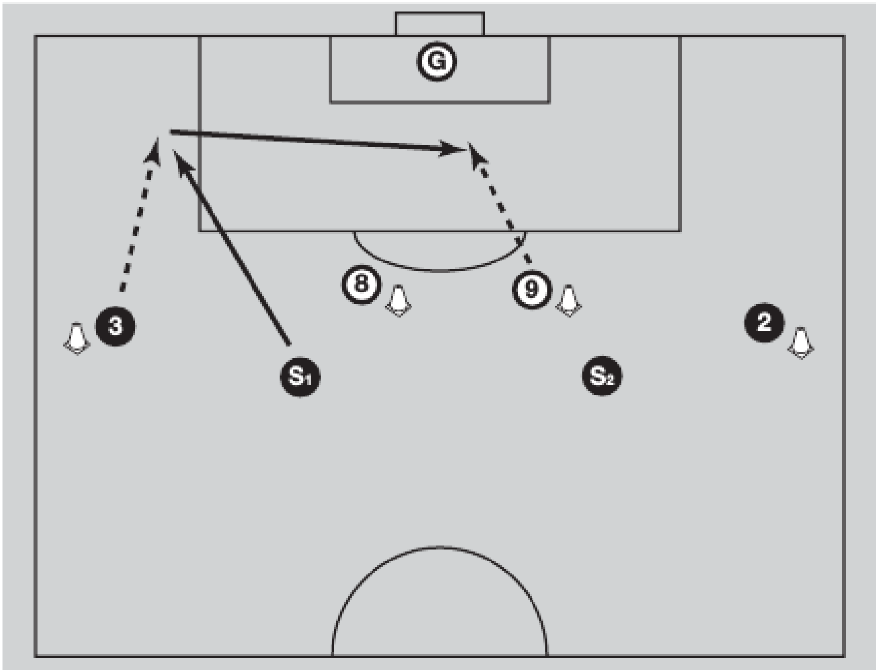


Figure 2.7 Drill design of **Fullback crossing** into the decision support system.

2.3.2 Forward passes

2.3.2.1 Arranging players with a deep attack

By arranging players behind each other, you create passing lanes and triangles that are more likely to penetrate the defense. A deep attack creates triangles of offensive players over the field. These triangles allow an offense to beat defenders by passing. With passing triangles akin showed in Figure 2.9, offensive players are never completely marked out of a play. Passing triangles open lanes between players and increase the chances for successful penetrating passes.

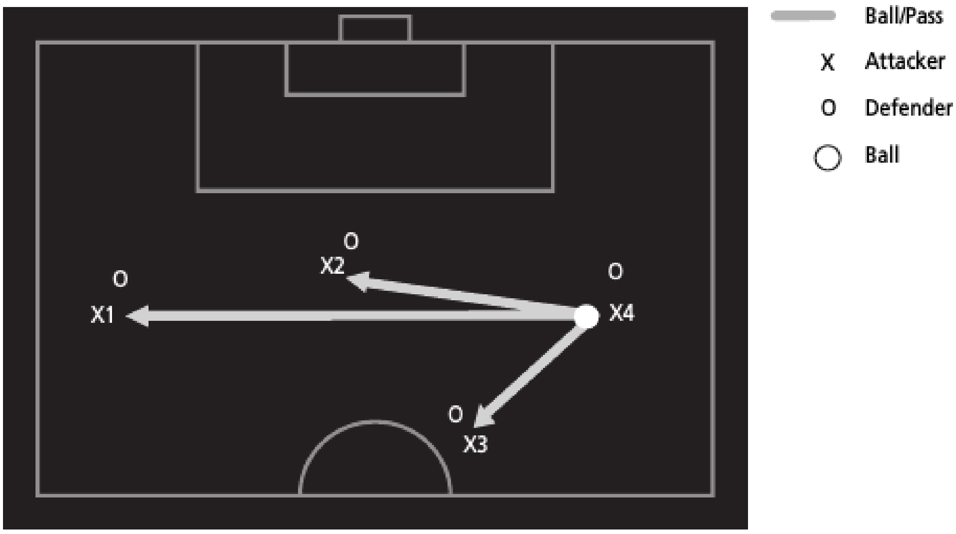


Figure 2.8 Drill design of **Arranging players with a deep attack** into the system.

2.3.2.2 Passing out wide to the fullback

With a pitch of around three-quarters length, 8 attackers versus 4 defenders + goalkeeper. Attacking team contains 4 defenders, 2 midfielders and 2 attackers. The game starts with defender no. 4 (Dark circle of figure) who plays the ball forward to midfield no. 7, who turns and plays it into no. 9 dropping off the defender. No. 9 plays the ball with one touch back to midfield no. 8 in support, who plays a one-touch pass out wide to the fullback, who has moved forward.

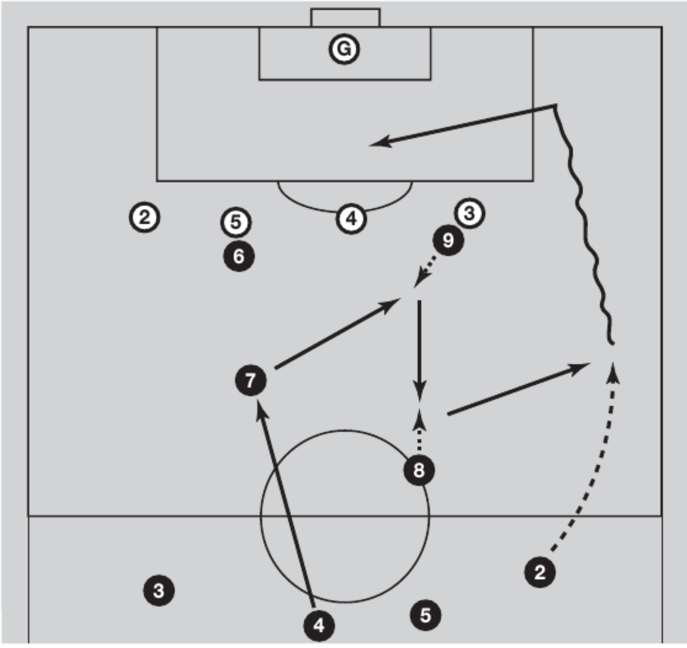


Figure 2.9 Drill of **Passing out wide to the fullback** in the decision support system.

2.3.2.3 Play for fullbacks

A server plays a ball to a fullback inside the penalty area. The fullback must control the ball while facing high pressure from a striker. The fullback must control the ball until able to make a clear, accurate pass back to a target player outside the penalty area. The drill teaches fullbacks the importance of control and good decision making in the defensive third of the field (Figure 2.10).

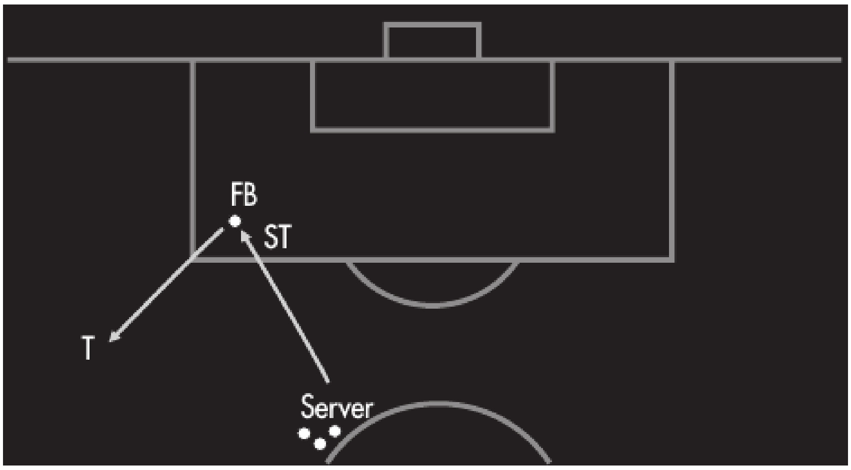


Figure 2.10 Drill of **Play for fullbacks** in the decision support system.

2.3.3 Passing from defensive to attacking third

2.3.3.1 Play for wingers

Winger A1 dribbles the ball to the goal line and passes back toward the top of the penalty box. A2 and A3 make runs forward for a shot on goal. They should time their runs so that they do not arrive at the spot before the ball. Defenders play passive defense. The drill's primary purpose is to develop the winger's ability to make precise offensive passes.

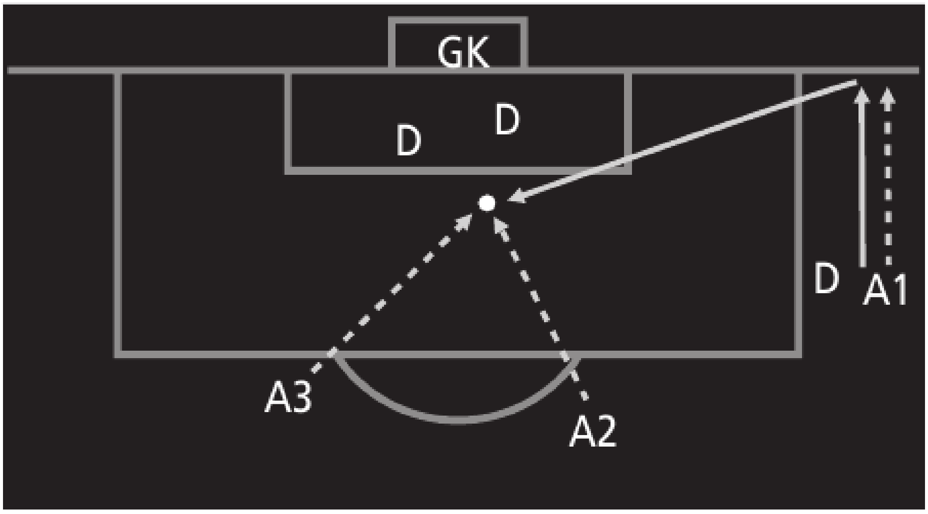


Figure 2.11 Drill of **Play for wingers** into the system.

2.3.3.2 Passing and receiving, angles of support

Player 1 starts with a ball and passes to player 2 who shows at an angle - 2 plays to 3 who shows at an angle and receives on the back foot, then dribbles into the open grid in the next channel - repeat combination - 3 to 4, 4 to 5, etc. - Players follow each pass.

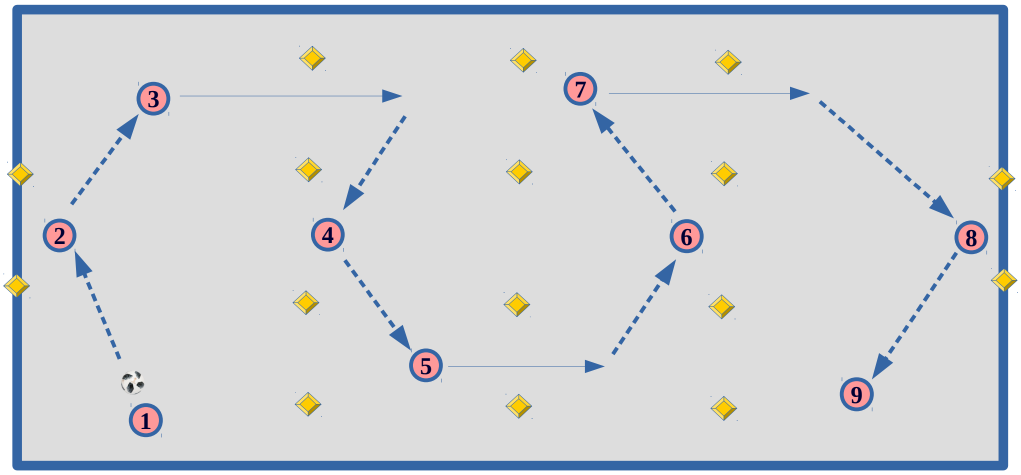


Figure 2.12 Drill of **Passing and receiving, angles of support** into the system.

2.3.4 Possession for match

2.3.4.1 Play with fullbacks

The midfielder (MF) plays the ball to the goalkeeper. The fullback runs wide, taking position facing the GK as the MF passes to the GK. The GK then passes the ball to the fullback, who must control the ball and pass forward to the winger. Practice first without defensive opposition. As the level of play improves, add defenders and pressure.

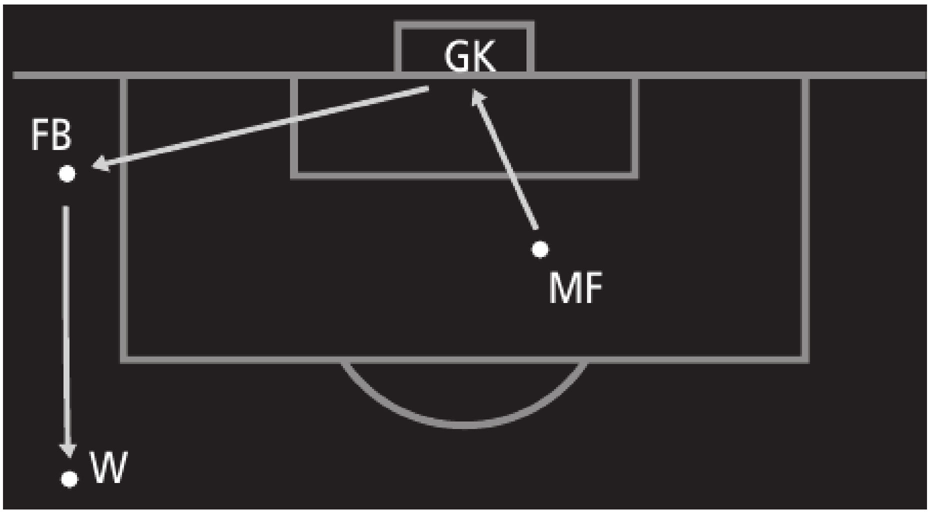


Figure 2.13 Drill of **Play with fullbacks** into the system.

2.3.4.2 Possession

Organization: Limited range, from the central line to the penalty area line. The whole field is divided into 4 side areas. The defensive line acts in front of the area, the two medians in front of them. 1-2 striker/s line up between the lines. Progress: The attackers pass the ball from right to left and vice versa, until they find a space to pass to the attackers. Only attackers can finalize the action. The defenders play on the line and the defensive midfielders press the attackers (ball possession). The defenders restart counterattack on the small doors, in case of interception. Advice: Attack; Create depth for shots on goal. Defense; Quick change of non possession of the ball to possession with a counterattack. Pitch size: half pitch. Margins cone: 4 equal lateral zones in the pitch.

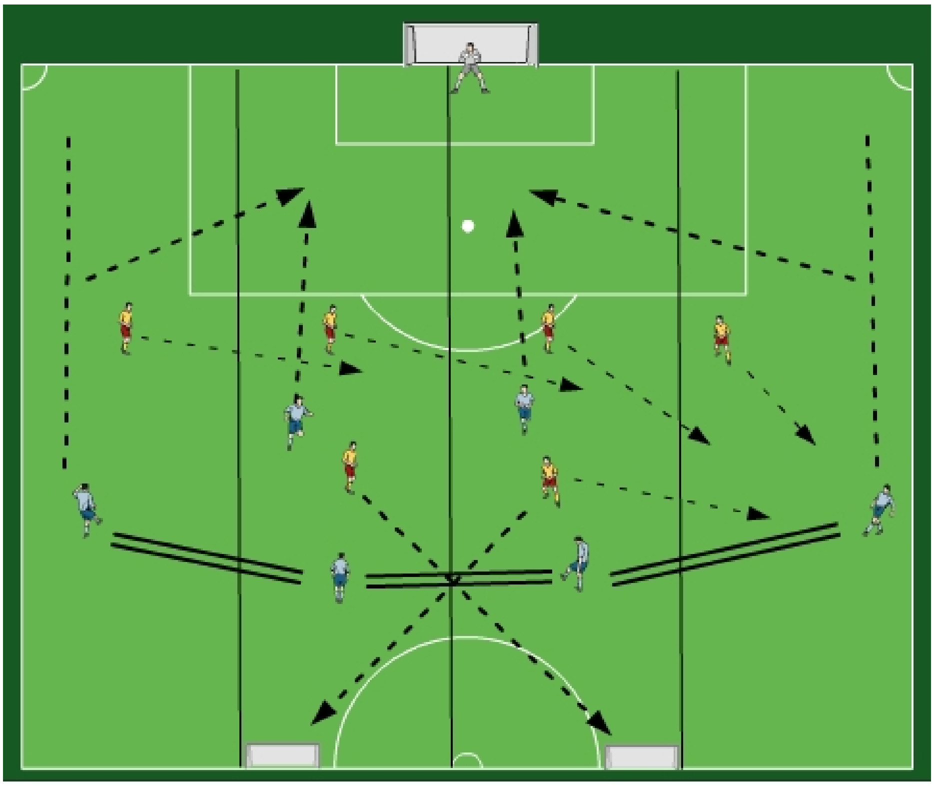


Figure 2.14 Drill of **Possession** into the system.

2.3.5 Passing accuracy for match

2.3.5.1 Passing in two boxes

Organisation: Two boxes, one large and one small. A player starts at each cone. The start cone has two players and the ball. Process: The passing starts with a diagonal ball. The receiver lays the ball off to the player opposite to him. This player passed the ball out of the inner box. These three variations are continuously repeated. Each player follows his pass. Alternative: Start with maximum 3 touches, then two, then one. When the drill is flowing, a second ball can be introduced. Start at the same time at cones A + B. 10 players are necessary in this variation because of the 2 balls. Two players start at each of the start cones (i.e. A + B). Now the players have to concentrate on the other ball in order to keep the rhythm. Tip: 8 cones, - When the drill is done with two touches, (control + pass) attention needs to be paid to the position of the foot and should be corrected accordingly. - The receiver should make a dynamic dummy run away from the player before receiving the pass. - Strong, precise passes on the ground. - The foot should be slightly lifted when laying the ball off. Field size: Outer box; 24 x 24. Cone margins: The outer cones are 24 meters apart. The inner cones are 6 meters away. The distance between the cones in the inner box is 11m.

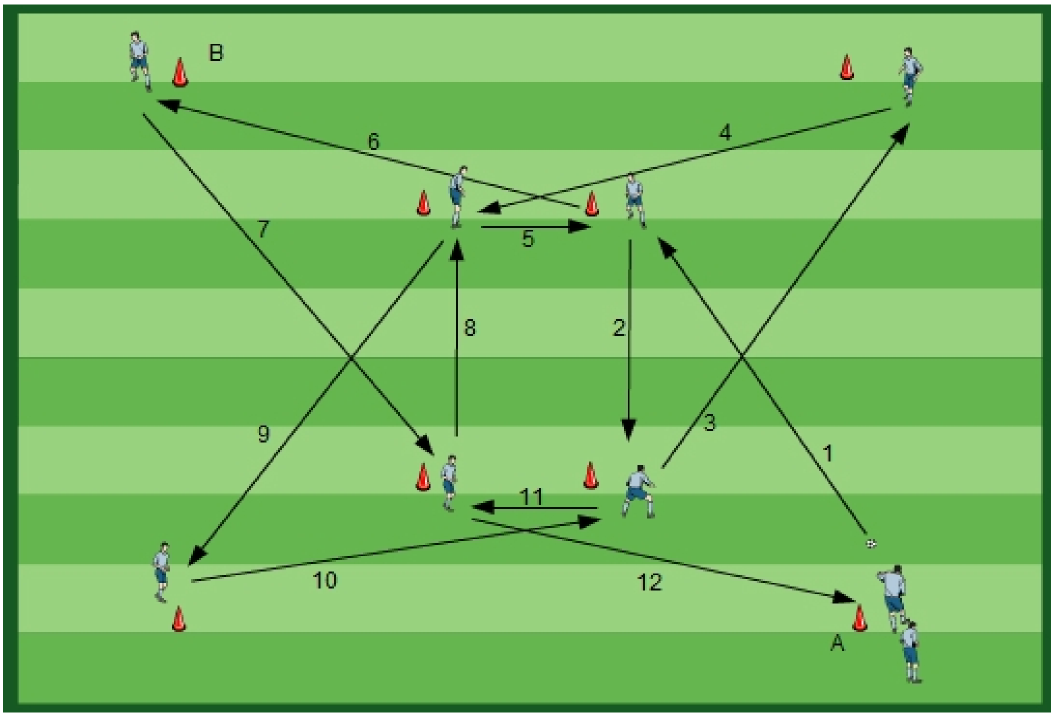


Figure 2.15 Drill of **Passing in two boxes** into the system.

2.3.6 Shots on target for match

2.3.6.1 Three servers, 1 shooting player

Half pitch. Three servers, 1 shooting player, 1 goalkeeper. Large supply of balls and ball retrievers. Server 1 plays the ball in from behind the forward no. 9, who shoots at goal (Figure). After following the shot in, the player jogs back to his original starting position. Server 2 plays the ball in horizontally from the right side and server 3 provides an angled pass from the left side. This sequence can be repeated ten to twenty times. The servers aim to make the player use both feet.

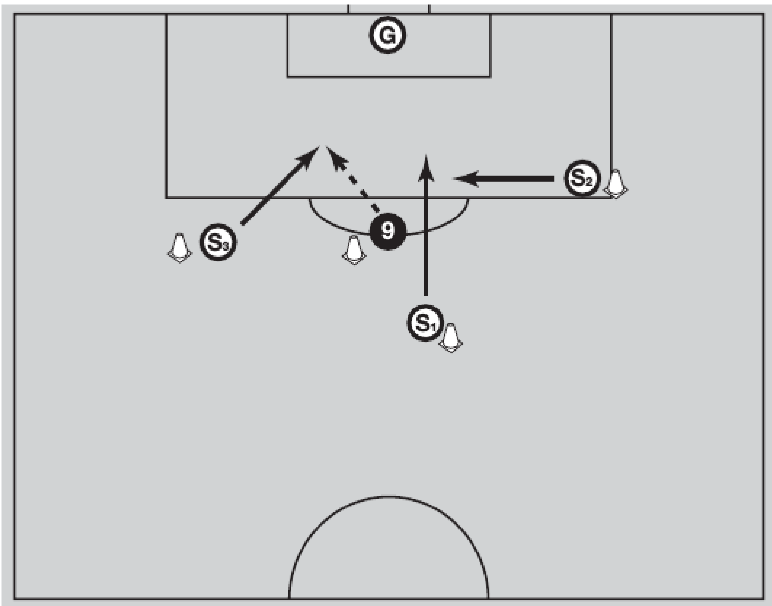


Figure 2.16 Drill of **Three servers, 1 shooting player** into the system.

2.3.6.2 One-touch finishing

Penalty area. Three servers outside penalty area, 3 forwards, 1 defender and 1 goalkeeper. Large supply of balls. This drill uses a conditioned game with limited ball touches to force players into taking rapid decisions such as analysing the time and space they have when shooting as well as practising general shooting technique. The ball is played from outside the box (Figure 8.5) by a server (alternate the service from each side of the box and from behind) to one of the forwards who has created space. The forwards are limited to two-touch passing, one-touch scoring and to three or fewer passes before a shot is taken. The other players should follow the shot in.

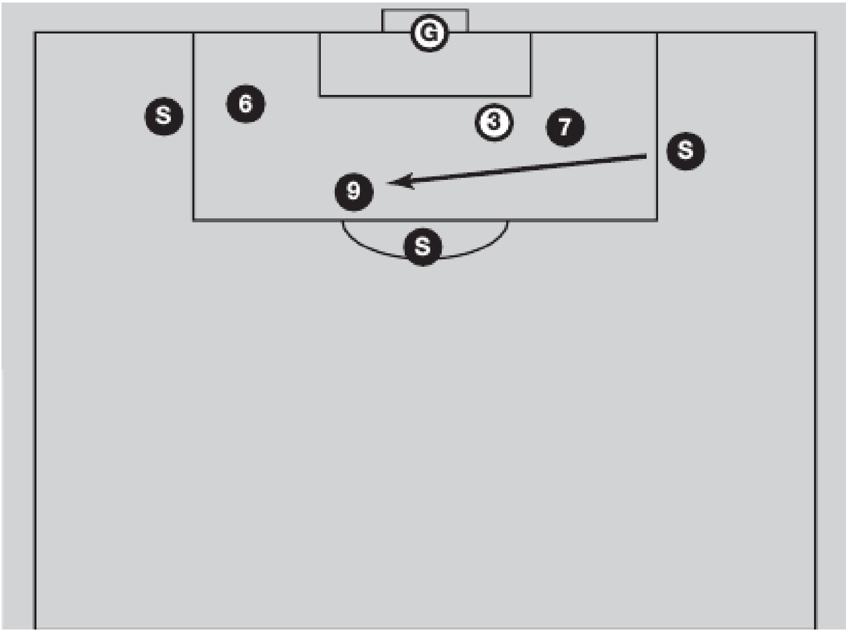


Figure 2.17 Drill of **One-touch finishing** inside the system.

2.4 Development of the decision support system application

From the features and their analysis information reviewed on last items we established the MySQL database in view of it delivers a popular open source database management system with ease of use, scalability and performance. Consequently we named it *soccerdb* as it is displayed in Figure 2.18.

After the database design we started the implementation of the frontend GUI object. That identified four elements just as tabbedpanes components coded in java within eclipse plataform. These elements are:

2.4.1 Prediction

The system displays and lets to choose freely in couple the antagonist country teams and then pushing button **Predict**, it gives who wins the game.

2.4.2 Stats

This tabbedpane returns a direction for training work and gives the feature stats analysis list.

2.4.3 Model

This tabbedpane graphics a histogram chart comparison of the statistical team features.

2.4.4 Plan

The next tabbedpane gives and directs what training and drill features develop as a management of action planning for the coming match competition.

Figure 2.18 Database model in mysql workbench called **soccerdb** of the decision support system.

Figures 2.19 and 2.20 give an idea of our development ambiance. They illustrate the object application and a portion of the java program code.

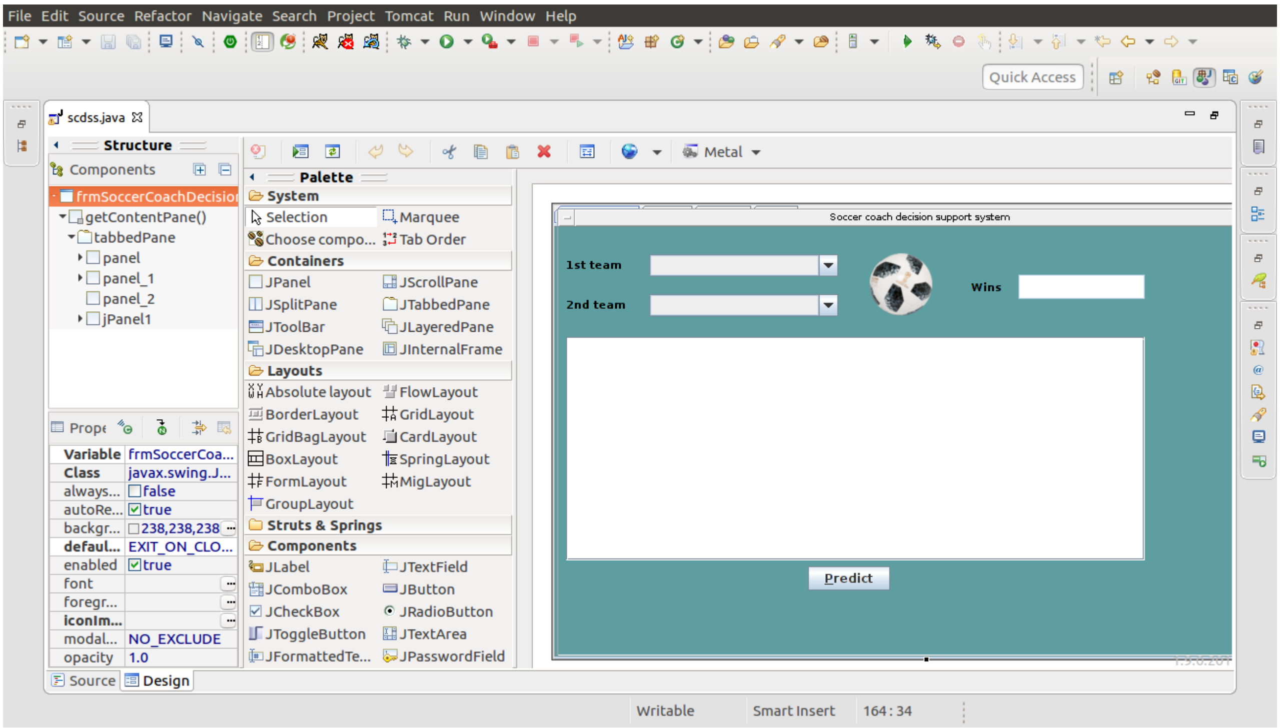


Figure 2.19 Implementation of the SCDSS in Eclipse platform.

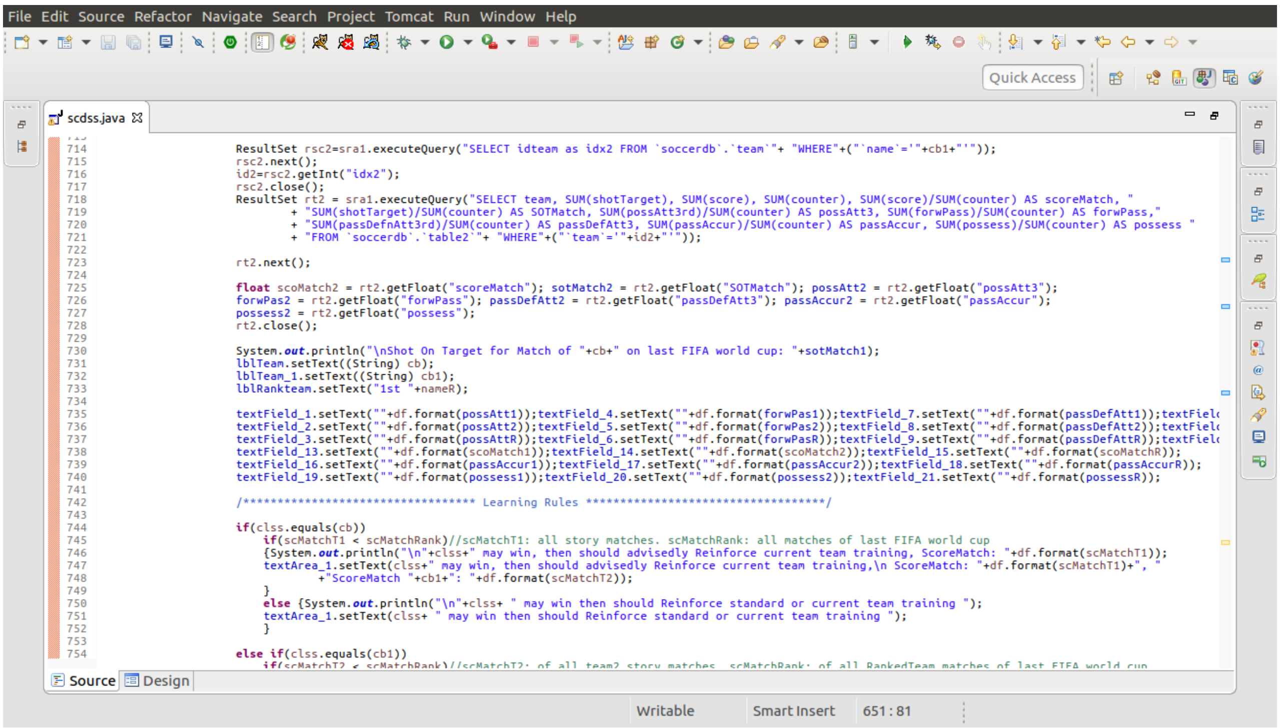


Figure 2.20 Java object code fragment inside the system.

Chapter 3

Experimental results

The statistical data of FIFA World Cup and Football Database [41], was used to test the system. From the experiment we can say that this system can decide soccer training and drill plans for coaches objectively. The process of planning tactics and training in a football team is an unceasing problem during all the tournament if we have evidently objectives. Coaches are required to consider qualitative and quantitative attributes in the team soccer match planning process. In the first phase the plan was chosen based on the prediction and their stats attributes, then in the second phase was used logic if-then-statement to select suitable training and drill plans for the objective team.

Differently to rating systems and betting models such as massively affairs extended all around; our system performs an ANN for making the prediction. We show hence an experimental test with a complete example of the SCDSS application on the following lines.

We executed the system and simulated an eventual match between Italy versus Brazil, having the Italy as the objective team who is being trained. The displaying of their results and analysis describes:

3.1 Match prediction

In the tabbedpane Prediction we entered Italy as 1st team and Brazil as 2nd team. Pushing button **Predict**, it showed some analysis results and delivered Brazil as the simulated winner of the match. Check out Figure 3.1.

3.2 Match stats

The tabbedpane Stats returned a direction for training work to the Italy team and additionally gave the features stats analysis list in comparison with the current top ranked team. See Figure 3.2.

3.3 Match model

Model tabbedpane showed the feature histograms of the teams Italy, Brazil and Germany (top ranked team to current date). See Figure 3.3.

3.4 Match plan

Plan tabbedpane gave the directions of training and feature drills to develop. Having the feature *Forward passes*; the drill to do is *Passing out wide to the fullback*. See Figure 3.4.

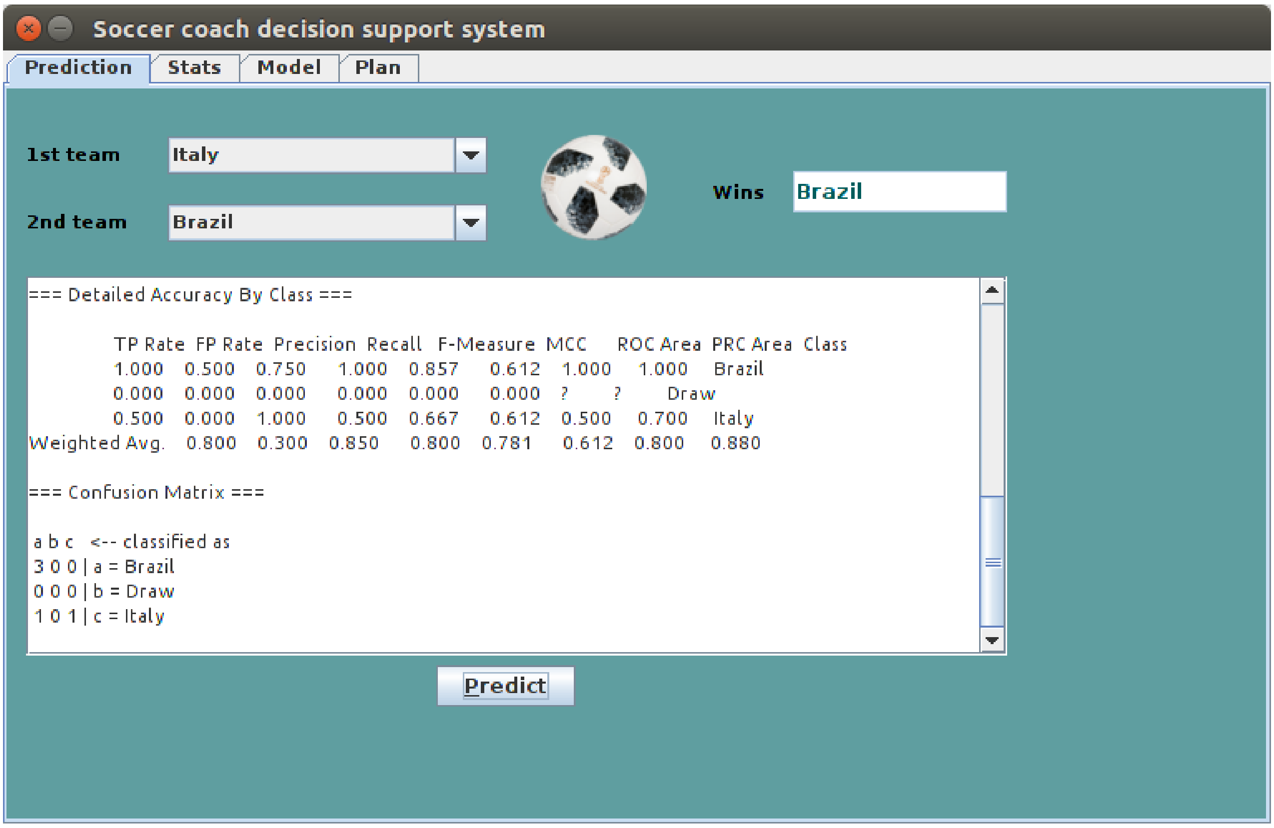


Figure 3.1 Prediction tabbedpane of the winner or otherwise a draw as a simulated result.

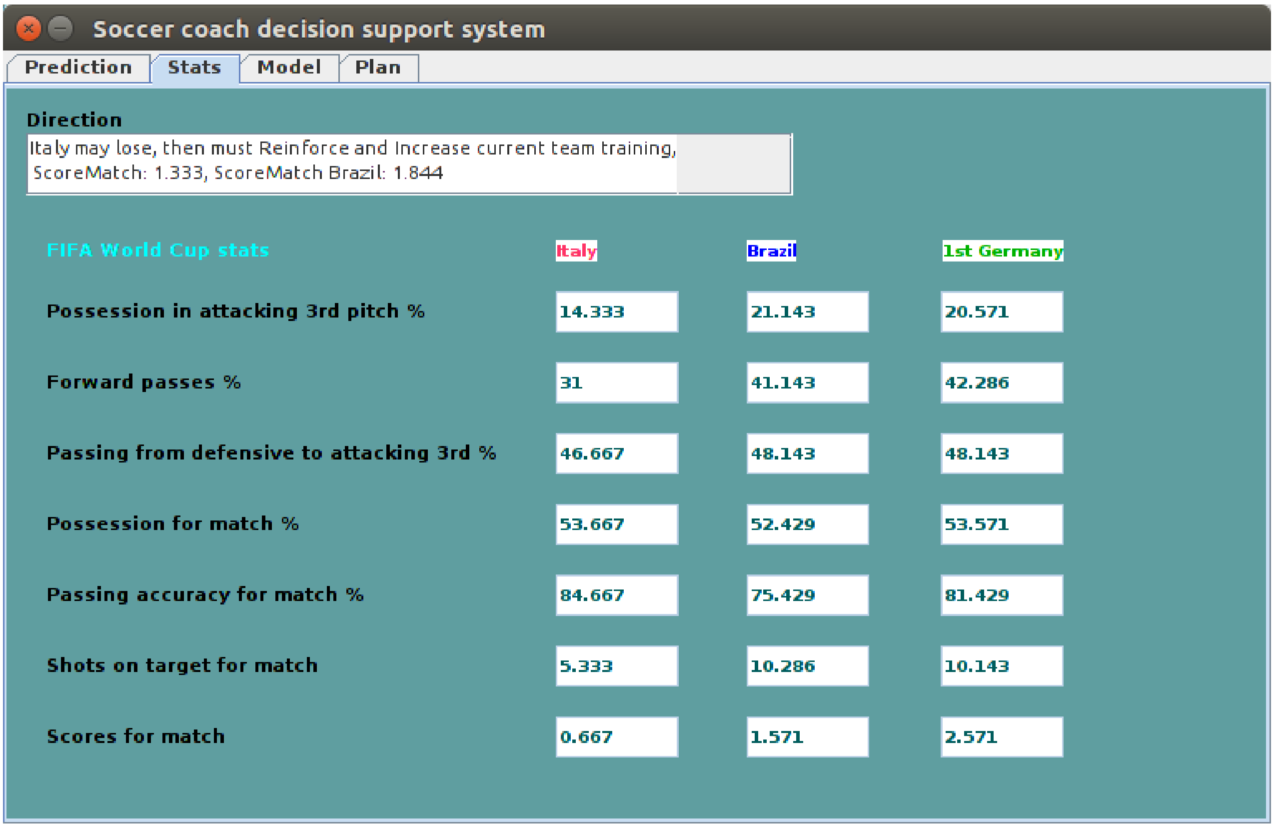


Figure 3.2 Stats tabbedpane for the training direction work and feature stats analysis.

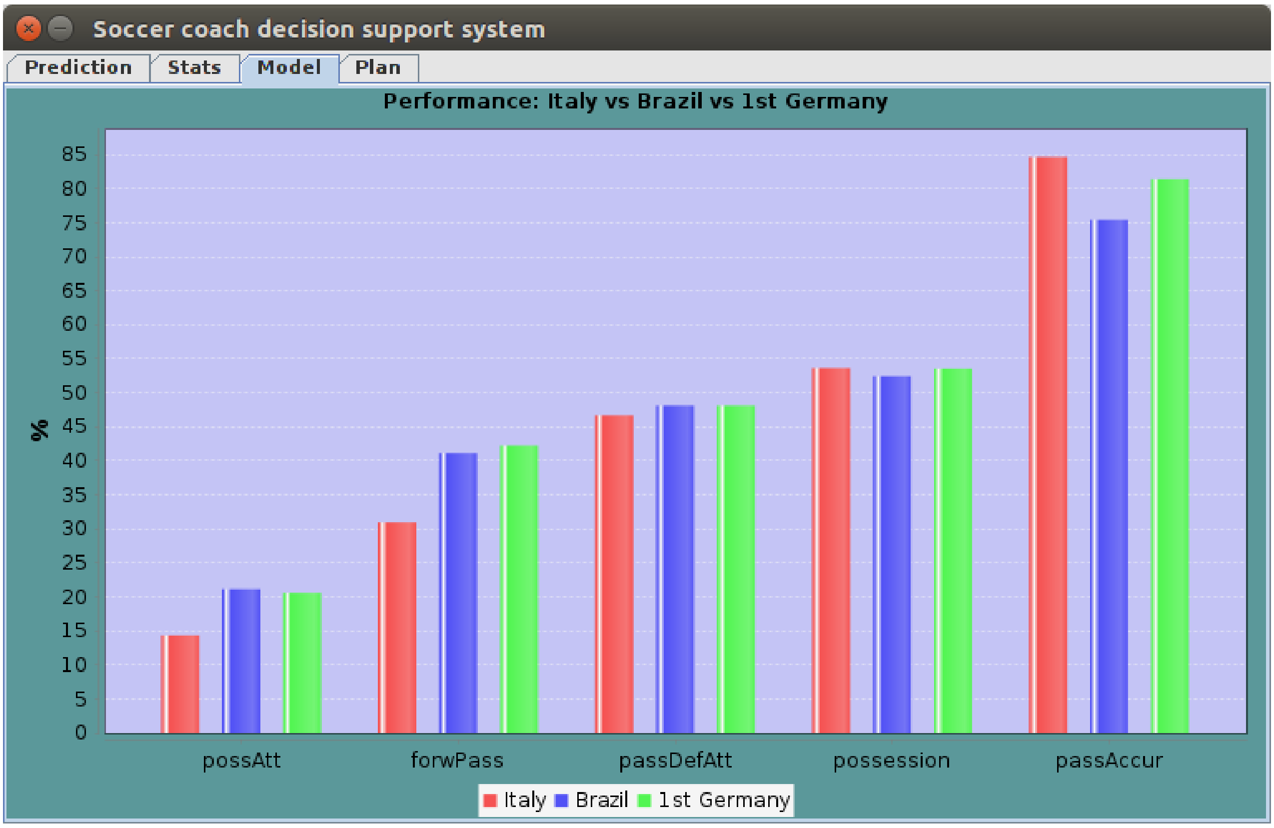


Figure 3.3 Model tabbedpane for the teams feature stats analysis comparison.

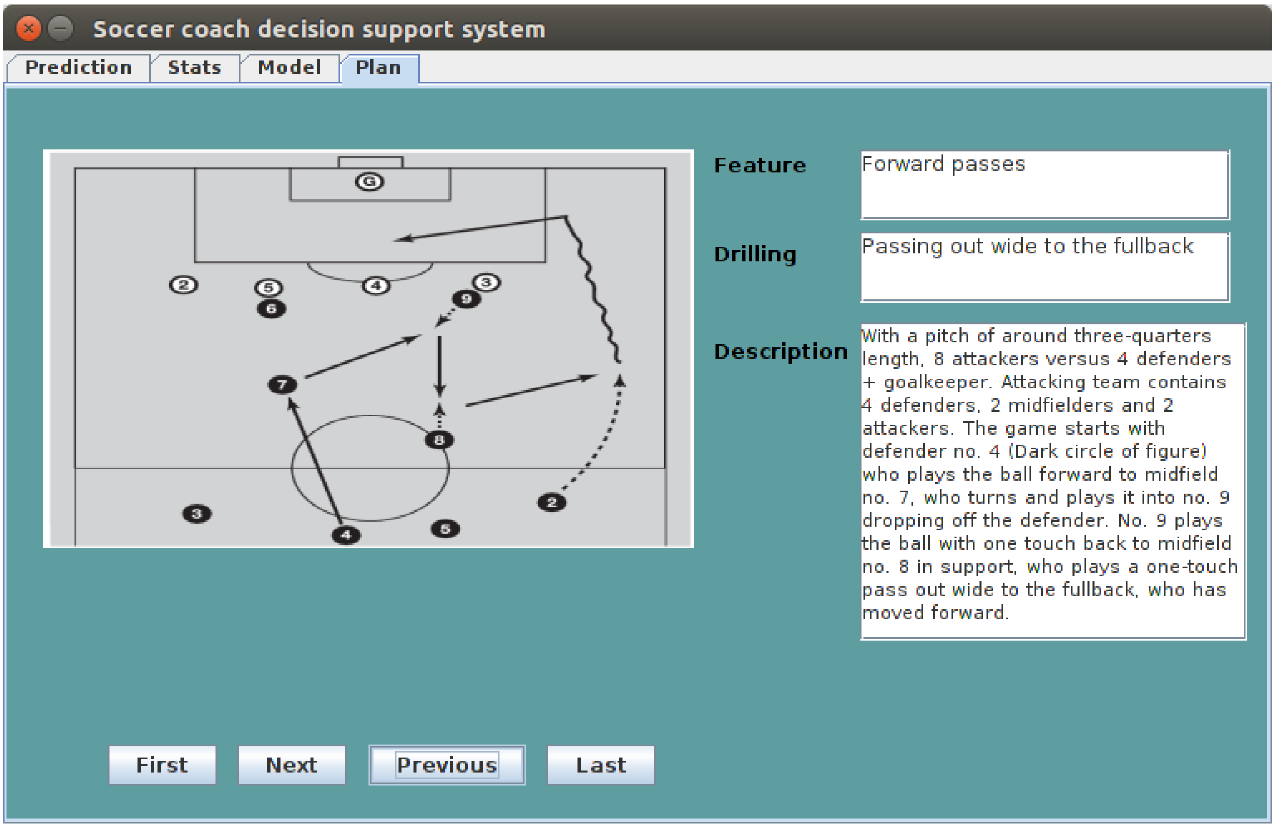


Figure 3.4 Plan tabbedpane for training and drill features of the coming match.

We have seen an application example of two antagonist teams: Italy versus Brazil.

The system returns Brazil as winner in Figure 3.1, then in Figure 3.2 offers a guide for the direction the italian team should take and also the stats analysis of the teams Italy and Brazil in parallel with the current top ranked team who is Germany.

Figure 3.3 illustrates their feature stats values. The sequential displaying of the system in Figure 3.4 prescribes to the Italy coach an action planning to take in this virtual match.

3.4 Soccer coach viewpoints

Showing our application to some soccer coaches of the Centro Sportivo Italiano and the Federazione Italiana Giuoco Calcio, they unstintingly said that despite being a suitable system, it would be agreeable persist its scalability adding more features and lineaments including equipment devices for tracking and monitoring the physical output and growth, and also mental and moral assessment tests for getting precision and effective information of the athletes.

Conclusions

We displayed a neural network model as a multilayer perceptron trained and focused on predicting the future soccer sport scenario.

With the state of art of predictive odds methods and ANN, we get introduced in data analysis and orderly we integrated logic rules to ANN to transfer domain knowledge and human intention at the learning process of the DSS framework.

We indicated the standard paradigm of DSS and established this criterion at once, we implemented the soccer coach decision support system in order to his definitions and methods. Consequently the testing was done at each simulation of match event.

The system delivers to the current teams their soccer stats, their graphic models and consents the training decisions over a feasible soccer planning.

The system has the effectiveness and improves the strategy and planning involving soccer training, drill and team performance concluding in a national soccer championship.

In our future big challenge we are looking forward to add in the features of human behaviours and the quicksilver attitudes of teammates into the system. In addition we considerate the incorporation of match event real time tracking equipment to get instantaneously information during the match meeting and act on it.

Bibliography

[1] http://www.managementstudyguide.com/decision-support-systems.htm

[2] http://lia.disi.unibo.it/Courses/AI/fundamentalsAI2017-18/lucidi/02bis-KBesempioToy.pdf

[3] http://dssresources.com/dsstypes/kddss.html

[4] http://docsdrive.com/pdfs/medwelljournals/jeasci/2016/1788-1791.pdf

[5] http://daim.idi.ntnu.no/masteroppgaver/013/13619/masteroppgave.pdf

[6] http://resources.fifa.com/mm/document/tournament/competition/02/40/51/17/64_0713_ger-arg_passingdistribution.pdf

[7] http://resources.fifa.com/mm/document/tournament/competition/02/40/51/10/64_0713_ger-arg_arg_teamstatistics.pdf

[8] http://resources.fifa.com/mm/document/tournament/competition/02/40/50/84/64_0713_ger-arg_ger_teamstatistics.pdf

[9] http://resources.fifa.com/mm/document/tournament/competition/02/40/51/17/64_0713_ger-arg_passingdistribution.pdf

[10] https://www.afp.com/en/products/partners/amisco

[11] http://www.tandfonline.com/doi/full/10.1080/02640414.2016.1169309

[12] http://www.football-data.co.uk/ratings.pdf

[13] http://en.wikipedia.org/wiki/Biological_neural_network

[14] http://en.wikipedia.org/wiki/Artificial_neural_network

[15] http://ai.unibo.it/webfm_send/238

[16] http://en.wikipedia.org/wiki/Supervised_learning

[17] http://en.wikipedia.org/wiki/Perceptron

[18] http://en.wikipedia.org/wiki/Deep_learning

[19] http://en.wikipedia.org/wiki/Universal_approximation_theorem

[20] http://en.wikipedia.org/wiki/Bayesian_inference

[21] http://ai.unibo.it/webfm_send/239

[22] http://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks

[23] http://en.wikipedia.org/wiki/Multilayer_perceptron

[24] http://en.wikipedia.org/wiki/Logic_learning_machine

[25] https://web.media.mit.edu/~minsky/papers/jokes.cognitive.txt

[26] http://web.mit.edu/cocosci/Papers/Science-2015-Lake-1332-8.pdf

[27] http://www.cs.cmu.edu/~xuezhem/publications/P16-1228.pdf

[28] http://www.cs.cornell.edu/courses/cs478/2000SP/lectures/rule-learning.pdf

[29] http://football-data.co.uk/data.php

[30] http://weka.wikispaces.com/Use+WEKA+in+your+Java+code

[31] http://www.fifa.com/worldcup/archive/brazil2014/index.html

[32] https://www.usingenglish.com/glossary/football-vocabulary/pass.html

[33] https://www.usingenglish.com/glossary/football-vocabulary/accuracy.html

[34] http://www.usingenglish.com/glossary/football-vocabulary/score-goal.html

[35] http://www.soccer-db.info/index.php?option=com_php&view=php&Itemid=224

[36] http://www.fifa.com/worldcup/archive/brazil2014/index.html

[37] http://ai.unibo.it/webfm_send/246

[38] Christopher Carling, A. Mark Williams and Thomas Reilly, Handbook of Soccer Match Analysis – A Systematic Approach to Improving Performance, 2007

[39] https://www.usingenglish.com/glossary/football-vocabulary/

[40] Stacey Chapman, Edward Derse, Jacqueline Hansen, Soccer Coaching Manual, 2012

[41] http://www.soccer-db.info/

[42] http://www.easy2coach.net/it/esercizio-calcistico/allenamento-calcistico-possesso3colorixscambioamuroepallasoprasul3uomo.html

[43] http://www.italia1910.com/

[44] http://weka.8497.n7.nabble.com/percentage-split-question-td23582.html

[45] http://weka.8497.n7.nabble.com/RE-evaluation-with-percentage-split-in-my-java-code-td22831.html

[46] http://weka.8497.n7.nabble.com/Multi-layer-perception-td2896.html

[47] http://weka.8497.n7.nabble.com/MULTILAYER-PERCEPTRON-td26873.html

[48] https://dev.mysql.com/doc/refman/5.7/en/