Evaluating the Effectiveness of Bayesian and Neural Networks for Adaptive Schedulling Systems

Bruno Cunha, Ana Madureira, João Paulo Pereira, Ivo Pereira
GECAD - Knowledge Engineering and Decision Support Research Center
School of Engineering - Polytechnic of Porto (ISEP/IPP)
Porto, Portugal
{bmaca, amd, jjp, iaspe}@isep.ipp.pt

***Abstract*— The ability to adjust itself to users' profile is imperative in modern system, given that many people interact with a lot of information in different ways. The creation of adaptive systems is a complex domain that requires very specific methods and the integration of several intelligent techniques, from an intelligent systems development perspective. Designing an adaptive system requires planning and training of user modelling techniques combined with existing system components. Based on the architecture for user modelling on Intelligent and Adaptive Scheduling Systems, this paper presents an analysis of using the mentioned architecture to characterize user's behaviours and a case study comparing the employment of different user classifiers. Bayesian and Artificial Neural Networks were selected as the elements of the computational study and this paper presents a description on how to prepare them to deal with user information.**

Keywords*— *User Modelling, Human-Computer Interaction, Machine Learning, Scalable Intelligence, Scheduling Systems

I. Introduction

The specification of systems that understand their users is a complex research area, as their expertise is continuously evolving. This creates the need for new methods to develop decision support systems that do not limit the user's choice but instead offer suggestions, consent changes and are able to learn from the observation and interaction with the user. As the system adapts to the users, their confidence in it will increase, until it is reckoned as fully trustable. This approach is achieved by employing user modelling techniques.

The major goal of incorporating user modelling techniques into complex systems has always been the increase of its usability [1], i.e. decrease the difficulty that users have on learning to use it. User modelling, when properly employed, shall make systems provide suggestions or necessary tools earlier (amongst others), effectively dropping the time required to achieve the desired outcome. With an increase of usability the system users get higher effectiveness rates, take less time to perform the tasks (i.e. are more efficient), can easily learn and understand the system and experience a gratification feeling, ultimately making them more likely of reusing it.

In order to achieve the adjustment to users' tendencies, this work employs the Architecture for user modelling on Intelligent and Adaptive Scheduling Systems [2]. It consists of a user information unit, a user classifier and a methodology to adapt its information to the user knowledge[3]. The user classifier is able to decode user behaviour and calculate their level of expertise, from expert to beginner. The main goal of this paper is to analyse the performance of the system using an Artificial Neural Network (ANN) and a Bayesian Network (BN) to classify users. A computational study of employing the selected architecture on the ADSyS (Adaptive Decision Support System for Interactive Scheduling with Metacognition and User Modelling Experience [4]) system is put forward.

The remaining sections are organized as follows: Section II presents a literature review on key topics required to fully understand the developed work, such as Human-computer interaction, user modelling and Mixed-initiative interaction; Section III contains a brief description of the ADSyS system; Section IV presents the case study analysed in this paper; Section V discusses the obtained results; and, at last, Section VI contains ideas for future work and presents the final conclusions.

II. Literature Review

This section describes some subjects required to better understand the ADSyS system behaviour and the conducted computational study, including Human-computer interaction, user modelling and Mixed-initiative interaction.

A. Human-computer interaction

The forms of interaction between humans and computers have come a long way since their inception. However, the emergence and development of new technologies, accompanied by an increasing investment in research in this area, make the route of Human-Computer Interaction one of constant evolution.

The progress of Human-Computer Interaction is verified not only in the quality of interaction of current systems, but also on the diverse interaction forms. The different research areas are looking to focus their attention on multimodality concepts (rather than unimodal), intelligent interfaces (instead of interfaces based on commands) and active interfaces (rather than passive interfaces).

As a scientific research area, Human-Computer Interaction is a multidisciplinary field that receives contributions from different areas, such as Psychology, Ergonomics or Artificial Intelligence. Besides, this area focuses its research not only in the study of computers and humans, but also gives special importance to the communication process between them. The multidisciplinary nature of this research area is justified by the contribution from each of the involved research areas: the use of their knowledge for the identification and understanding of the human being limitations (Cognitive Psychology), the restrictions that existing technology imposes (Computer Science), the phenomena that the communication process comprises (Linguistics and Sociology), amongst others.

The importance of Human-Computer Interaction is related to the fact that even the most sophisticated computer system is useless if not properly used by its users. This argument relies on two main concepts that should be considered in the design of interactive systems: functionality and usability [5]. A computer system may be defined, in the context of Human-Computer Interaction, as what the system can do: the functions provided by the system should contribute to the achievement of the purpose for which the system was created. The functionality of a system is defined as the set of actions or services available to users. However, the value of a certain feature only becomes visible when the user is able to use it effectively. The usability of a system with a given feature is defined by the degree of efficiency and suitability in achieving certain goals for particular users [5].

The activities performed by the user present three levels: physical, cognitive and affective. The physical level determines the type of interaction mechanisms to be used. The cognitive level reads up on the user's method to understand and interact with the system. The affective level tries not only to make the interaction a pleasant experience, but also to encourage the user to continue to use the system.

With the progress of the Human-Computer Interaction field, a new attitude in the development of computer systems as emerged: the user should always be the focus.

B. *User Modelling*

User modelling is an area that appeared in 1979 which focuses on adapting system's content to specific user needs in order to achieve a higher efficiency. Modern systems are complex and have to deal with multiple users with distinct characteristics, so content adaptation should be a focal point.

The point of user modelling is, as the name states, to obtain a representation of the user – the User Model (UM). The UM is a structure that symbolises the system's convictions about its users, using that to present the necessary information to adapt its content to their necessities. The content of a UM is vital to its degree of success. A truthful UM shall cover records about the user's interests, goals, preferences, domain knowledge and progress. Initially, the system can use several techniques to generate the first model. As time goes by there will be changes to user's preferences and characteristics, so a UM should be properly updated.

There are a number of techniques to create a UM. The most used are methods such as Stereotypes, Decision Trees, Bayesian Networks and Artificial Neural Networks. They are further described next, with emphasis being given to the selected techniques to conduct this study.

Stereotypes consist in researching the most frequently occurring characteristics of users and attempting to match them to the current user (operating the system). A Decision Tree is a structure that defines rules on how to divide certain data into groups and is mostly used to classify both users and any type of media the system uses. Additional techniques, which are hardly found in current literature, include Linear Models, Overlay methods or Plan Recognition.

A Bayesian Network is a graphical model for probabilistic relationships among a set of variables that allows representing and reasoning about an uncertain domain. The nodes in a BN represent a set of random variables from the domain. A set of directed arcs connects pairs of nodes, establishing dependencies between the variables. The conditional probability distribution (CPD) is where the dependencies are represented. In this work, the expression "independent node" is used when a node that has no inbound arcs is mentioned and the expressions "parent/child node" when the node where the arc starts/ends are point out, respectively [2]. Since an independent node does not have inbound arcs, the CPD is simply the probability distribution of the variable. A child node (has at least one parent) has a CPD that defines the conditional probability for each value of the node given each combination of values from the parent nodes. The CPD connects all of the potential outcomes from the parent nodes to the probability distribution of the node [6]. The restriction associated to the arcs in a BN is that they shall not generate direct cycle, meaning that the BN stays acyclic.

To outline a BN several modelling decisions are necessary. The design of the CPD for each child node and the overall network structure can be subjective, reliant on the responsible for its creation and the development decisions that are made. Typically, the first step involves the definition of the relevant variables to the global problem. These will be the variables that are represented by the nodes in the final BN. There can be multiple types of nodes depending on the desired outcome; In this paper, *Boolean* (yes or no) and ordered (e.g. values {beginner, intermediate, advanced}) nodes are used. The second step is to define the network structure. The parent nodes are pinpointed and the arcs that connect all nodes on the BN are defined. The main concern is the representation quality of the relations between each node: two nodes should be connected only if one influences other, with the connection arc defining the influence flow (i.e. which one is the parent node). The last step is the definition of the CPD table for all nodes. To do this, it is necessary to delineate the probability that the child node will have for each possible combination from the values of the parent nodes. In this paper, only the child nodes have unique CPDs, where the independent nodes are Boolean nodes with a yes/no outcome, equally distributed (0.5, 0.5). This final step might create a huge CPD table if a node has many parents (e.g. for n parent Boolean nodes the CPD table requires 2^{n+1} definitions), so the network structure has to be carefully defined.

Bayesian Networks present multiple advantages over other techniques due to their flexibility, the capacity to deal with missing (unknown) values, the capacity to work as a framework for expert knowledge and due to its morphing nature, adapting itself to the user as he learns the system.

An Artificial Neural Network is a computational technique that offers a mathematical model inspired on the neural structure of intelligent organisms that gather knowledge trough their experience [7]. ANNs are systems of organised nodes (simulating neurons) which exchange messages between each other. The node connections have weight values that can be tuned based on experience, making ANNs capable of learning but also highly dependent to their inputs [8]. These nodes are able to work in parallel to find a solution to a certain problem. From an abstract perspective, an ANN (after a proper training phase) is fairly simple: the network receives a set of input values, returning and appropriate output. ANNs can be of multiple types (e.g. feedforward) and topographies, which must be chosen taking into consideration the different categories of problems [9].

Real-world applications of ANNs involve classification problems (e.g. pattern recognition), the elaboration of predictive models (to forecast values of a particular variable) and data processing (e.g. clustering and compression) [3].

The latest developments on the User Modelling field are related to the its (lack of) standardization (to increase the cooperation between systems with user modelling techniques), the possible legislation on the data privacy and the innovative Virtual Reality trend, which requires new models and guidelines on how to design User Modelling systems [3].

C. *Mixed Initiative Interaction*

As previously stated, personalization is a key aspect of effective Human-Computer Interaction [10]. Even if using a Mixed-initiative (MI) approach does not primarily require a human, it is one of the most used techniques, with user acceptance to evidence its popularity [11]. MI is defined as the mutual control by the system and the user in the communication between them. Its main goal is to deliver an ambitious system that is autonomous and able to recognize gains from modifying the interface and interacting with the user, doing it whenever it is beneficial. The other main advantage is allowing users to refine the interface according to their needs.

There are, effectively, two ways to adapt information without user input: either by drawing attention to certain content or by showing/hiding specific information. The most frequent content adaptation techniques are as simple as sorting, zooming or scaling specific content. MI key principles and problems followed in this proposal have been defined by E. Horvitz [12], with the key points being the significance of the value added by the automation, allowing efficient direct invocation/termination and considering the overall uncertainty about the user ambition.

III. ADSyS System

At this stage has been considered relevant to study the influence of the used machine learning technique. This case study has the purpose of determining the impact of using Bayesian or Artificial Neural Networks to model ADSyS users. This section starts by presenting the ADSyS scheduling system and, on the next section, it is presented the application of the previously proposed Architecture for user modelling for scheduling systems, incorporating an ANN [3] and a BN [2] within the user modelling module, as seen in Fig. 1.

ADSyS is a scheduling system where communities of agents model real-world manufacturing problems that are affected by disturbances. These agents are able to learn and administer their internal behaviour and interaction amongst them, collaborating to achieve the desired goal.

ADSyS is composed by four main modules [4]: the integrated interface module; the scheduling module, the user modelling module; and the dynamic adaptation module. Furthermore, four interaction modules (task editor, machines editor, order set editor and Gantt chart editor) are responsible for the input data, i.e., for the definition of the scheduling problem and the visualization of the results.

The Scheduling Module constructs a scheduling solution to the problem using a combination of metaheuristic; the scheduling problem is decomposed into a series of smaller problems, and later, through cooperation, a global schedule is achieved.

The ADSyS interface [4] enables interaction between the user and the scheduling module in order to make possible operations such as the definition of meta-heuristic to be used (and its parameters), the results via Gantt charts or even the possibility to interact with it to modify results (e.g. incorporate dynamic events).

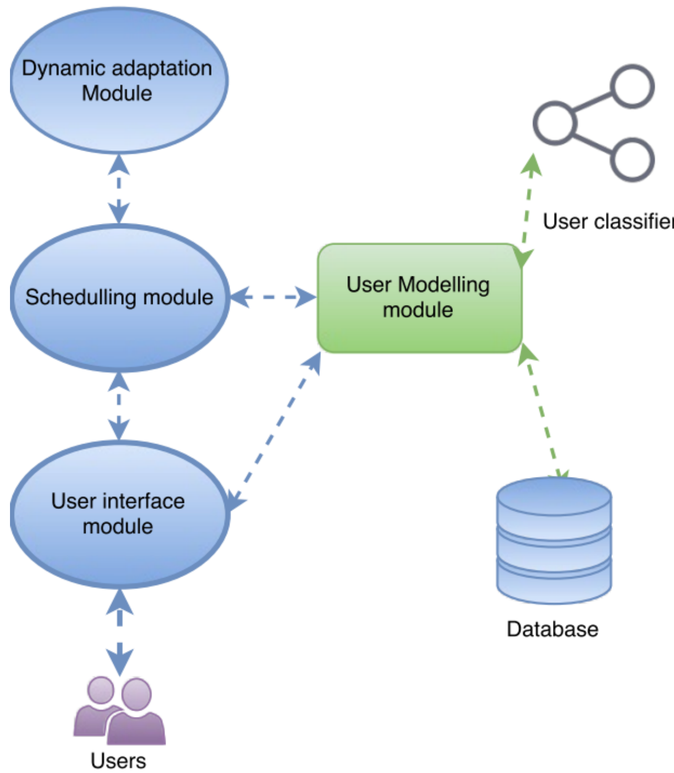

Fig. 1 - ADSyS Architecture with emphasis on the user modelling module.

The Dynamic Adaptation Module employs machine learning methods to forecast the preeminent integration mechanism to use to incorporate new tasks that arrives in a dynamic context [3]. The working flow of this approach is relatively straightforward. When a new scheduling task arrives in ADSyS, the information related to the current scheduling problem and the new task is sent to a Decision Tree based classifier, which then returns the predicted IM (such as earliest due date or greatest priority first, amongst others). After that, the system can use the predicted IM to incorporate the new order in the current scheduling plan.

The user modelling module is responsible for improving the learning curve of new users while boosting the productivity of expert ones. It is composed by several cooperating components that control the characterization of user behaviour and its analysis, using that data to provide a customized experience. This module was built centred on the architecture presented in the following section.

IV. User Modelling on ADSyS

ADSyS follows the architecture for user modelling proposed in [2], [3]. This architecture is comprised of a user information unit, a user classifier and the content adaptation methodology. The user information unit is in charge of encapsulating all relevant interaction between the users and ADSyS. It also includes the database (and its structure definition) which stores that information, to be delivered afterwards to the classifier. The user classifier is a mathematical structure, responsible for distributing them according to their level of expertise into one of three roles: beginner, intermediate or expert. The content adaptation methodology guides how the system information is improved to match the user knowledge level.

Regarding the user classifier two structures are selected: an ANN and a BN. Both structures were selected in order to achieve the goal of this work: to study the differences between using one or the other technique. Both are well established techniques and are commonly used to perform numerous functions (e.g. medical diagnosis [13]). The advantages for selecting these methods instead of others (e.g. Decision Trees) are described in the literature revision, on section II. It is important to note that both classifiers can be used due to the fact that all relevant variables are approximately independent; if not, a Bayesian classifier would not be possible to apply.

Dynamic BN can be implemented using multiple techniques, such as structure variation, CPD changes or even both (full graph variation) [6]. In this paper, the dynamic BN has a probabilistic structure that varies in time. In sum, the probabilities in the CPD table shall be affected to adjustments over time, but the BN graph structure will remain static.

The proposed BN has been designed to classify the users in three different levels, as previously explained. The graph structure [2] is represented in Fig. 2. The BN contains twenty nodes, with thirteen of them being independent. *UserType*, the "result" node – the last descendant and the one that grades the user – has one of three states: beginner, intermediate and advanced. To increase the perceptibility of the graph, logical subgroups were introduced, following the results of preliminary studies [2]. This consists in creating nodes that have the independent nodes as parents and that are, themselves, parents of another node. To keep a well-balanced graph, the proposed limit for a max parent number (for any node) is four; however, the presented BN contains, in its majority, subgroups that only have three parents. Also, for this this rule to be successful, each node (not counting the outcome one) can only have one outgoing edge – although a node can have many parents, every parent only has one child node [2] – and the connections ought to be created based on the associated business logic (e.g. the proposed BN combines the analysis of a single task j completion time, C_j, and the overall completion time, C_{max}).

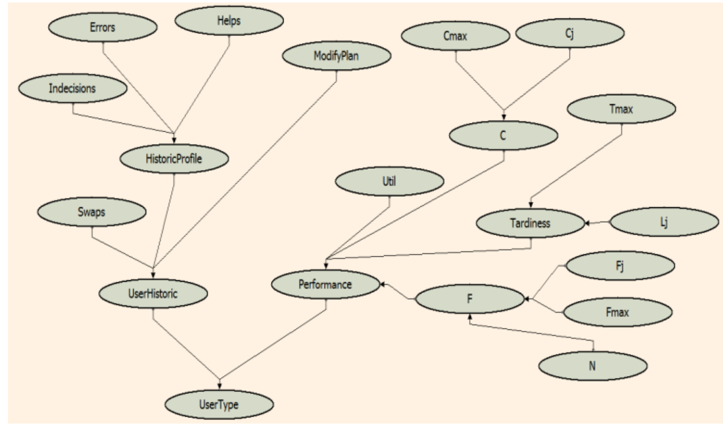

Fig. 2 - The developed BN contains 20 nodes, with 13 being independent.

The thirteen independent nodes are the ones which have their CPD adjusted to mirror the users' knowledge. Each independent node is founded with a tied probability distribution – 0.5 for the 2 Boolean outcomes – and these default values are united to build the initial outcome, which establishes the user as beginner. Nevertheless, it is only after a specific number of interactions that the user classification (from the BN) starts being used. The non-independent nodes – each "child" node – possess a unique, static CPD, defined beforehand in accordance to the expertise outcome expected from the BN. The definition of the static CPD values is a demanding task, as it requires a meticulous analysis of the user information [14] to create, on a trial and error method, proper tables that are able to classify the user correctly. A static CPD table from the developed BN is presented in Fig. 3: The Performance node receives the values from the *Util*, *C*, *F* and *Tardiness* nodes and based on the static definition (the *Yes* or *No* red and yellow columns, respectively) it creates a value, ranging from 0.0 to 1, on the topic of the user's proficiency to improve a scheduling plan.

Parent Node(s)				Performance		
Util	C	Tardiness	F	Yes	No	bar charts
Yes	Yes	Yes	Yes	1,0	0,0	
			No	0,8	0,2	
		No	Yes	0,8	0,2	
			No	0,6	0,4	
	No	Yes	Yes	0,8	0,2	
			No	0,6	0,4	
		No	Yes	0,4	0,6	
			No	0,25	0,75	
No	Yes	Yes	Yes	0,75	0,25	
			No	0,4	0,6	
		No	Yes	0,4	0,6	
			No	0,3	0,7	
	No	Yes	Yes	0,4	0,6	
			No	0,3	0,7	
		No	Yes	0,2	0,8	
			No	0,0	1,0	

Fig. 3 - Performance node (non-independent) static CPD definition

To update the independent nodes CPD, the BN is connected to the database information; pragmatically, whenever the DB has new data, the nodes CPD is adapted to it. Using C_j as example, let us simulate a database that has information containing 76 modified plans and 43 C_j improvements for a specific user. The conditional probability that this user improves the C_j, knowing that he will modify the plan, is 0.605 – the network is adapting itself since the starting 0.5, providing a more precise user projection. This shall happen to every independent node, with the variation being the selected formula to estimate all probabilities (e.g. the node Errors uses the captured information from the interaction with the scheduling diagram, not the number of modified plans - C_j).

Just like the BN, the ANN was developed to classify the user into one of three roles, from beginner to expert. The ANN has three layers (as seen in a sample on Fig. 4), including the input and output. The input layer contains thirteen nodes, one for each stored variable in the DB related to the user's interactions. The middle (hidden) layer, and its number of neurons, is related to the complexity of the problem (the more complex the more neurons shall be needed). This layer supports the rest of the ANN inference. The middle layer is composed by 26 nodes, prescribed to be twice the size of the input layer.

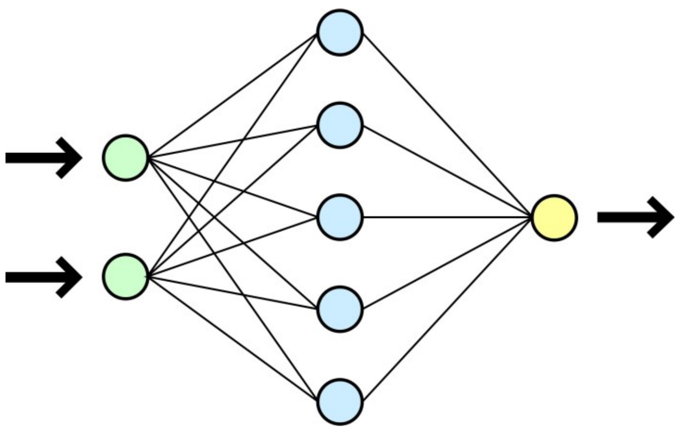

Fig. 4 – Low scale sample of the developed ANN

To be accurate in the user classification, the ANN needs to have well-defined weighs for every node. This is achieved with supervised training, which operates a specific dataset that contains input data and its expected output. The network conducts several runs and adapts node's weighs until it is capable of accurately classifying the original training data.

An important topic of training an ANN is the necessary dimension of the training dataset: if there is not a big enough number of known cases or if they are similar, the neural network will not be able to achieve its objective. Still, operating a large dataset significantly increases the necessary effort to obtain results, but it compensates in a good network definition. To give a better guarantee of proper training, groupings of all DB fields (and the corresponding classification) were made: for each DB variable, an appropriate value was selected and combinations of likely values for the other variables were produced (reproducing real user cases). This process was performed for all DB values and every applicable value that they could have; e.g. C_{max}, one of the DB variables, was set with {0.0, 1.0} and eminent values from real cases, like {0.42, 0.71}. Afterwards, and for all set values, the other variables were fixed with their appropriate values, which allowed the definition of the correct output. Finally, the combinations were inserted in the dataset, which was spread as the ANN input. The node weights were locked (finishing the training phase) after the network presented an accuracy higher than 98% (of correct predictions rate).

The proposed ANN follows a feedforward structure: the information is transferred always in the same direction (input to output nodes, passing via the hidden layer) and the node connections cannot create cycles. The heuristic carefully chosen to train the network was the resilient propagation learning heuristic, created by M. Riedmiller and H. Braun [15]. This allowed for a cost-effective and clear training process. The sigmoid activation function was also used during the training phase.

To infer the user level of expertise, the ANN uses the data kept in the DB. In practice, whenever the system needs the user classification (e.g. after logging in), the ANN retrieves the latest user information from the DB and propagates it to the input nodes. The output node will then present the user level of expertise. To prevent classification errors, a MI threshold value is fixed, defining the least possible total of information needed (e.g. number of sessions; modified scheduling plans) before being able of creating a proper classification (just like the BN).

V. DISCUSSION OF RESULTS

In order to evaluate if there is a categorization disparity between the use of a BN or an ANN to perform user classification, a study was performed; it consisted in obtaining user cases and classifying them via each network and, afterwards, saving the percentage for all user profiles, from beginner to expert. Then, the final user classification from the BN was compared to its respective ANN counterpart, creating the final results shown in Fig. 5 which presents the percentage of users on each level for both networks. The case base used to compare the networks was obtained both from real user cases and via random generation. The random generation consisted in creating arbitrary values for each field used to classify the user. This ensures a substantial case base but introduces some artificial values which would not be found in a real-world scenario. However, this fact is not considered as negative due to the possibility of exploring the capacity of both classifiers to adapt to those extreme and fictional values.

As seen in Fig. 5, the networks present mostly similar classifications. Regarding the number of profiles classified as beginner, the networks present a difference of 1%. Such a minor difference concludes that both networks are well prepared to discover and deal with users that are new to the system, which is both the main focus of the presented architecture and the reality of ADSyS users. On the subject of users assigned to the intermediate and expert levels, the networks present a 13% and 12% disparity, respectively (with the 1% discrepancy being attributed to that same difference on the beginner level). This variation shows that the studied networks have a contrast of classification when users gain expertise, even if only on a small percentage. After a proper analysis of each case with a classification disparity the variation can be attributed to the generated instances: while the ANN was trained and kept adapting itself even to such extreme cases, the static BN is more conservative, prepared to deal properly with real users but not handling well the generated instances – close to 10% of the total cases, near the disparity percentage between networks.

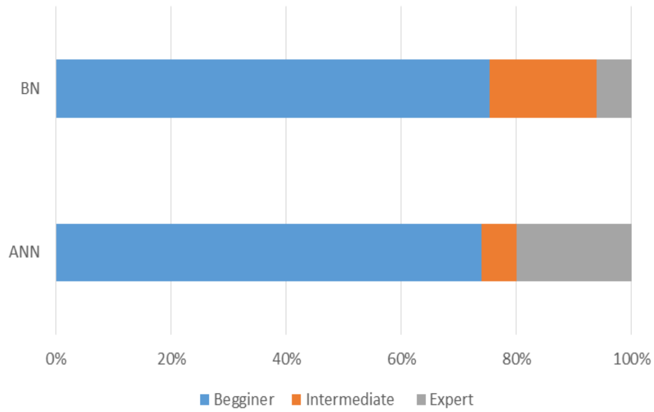

Fig. 5 - Percentage of users on each classification

The result from this study is in accordance with what was expected from the literature and empiric perspective. If properly design and trained, both networks provide accurate classifications and can be used interchangeably. From a computation time position, both networks are very fast (on ADSyS), presenting their classification in a non-noticeable timeframe, so the required time to calculate the user expertise is not a relevant factor.

Pragmatically, when there is a desire to introduce user classification into any system, a BN should be used if there is no previous work done as it only requires the definition of the CPD table, feasible after identifying the user information. If a case base of information on how users interact with the system (and the associated proficiency) already exists, it might be better to implement an ANN, as it can be enhanced to recognize any type of user without the need to manually define the node weighs.

VI. Conclusions and Future work

A computational study was presented in order to compare the approaches based on Bayesian and Artificial Neural Networks to model users' behaviour and profiles on ADSyS. As measured, Bayesian Networks are much simpler and can easily be update; however, they can only work when variables are independent and most of the definition has to be done by the developer, particularly during the CPD table definition. A BN should be advantageous when starting from scratch, with no previous work. An ANN is faster, easier and the most appropriate method to implement if there is already a proper case base to perform the network training. However, there will always be exception to this theoretical rule, hence the need for a case by case analysis and the impossibility of stating one method as superior to the other.

It is suggested as future work the application of the proposed architecture on other types of systems, preferably on certain complex areas that benefit from content adaptation (such as medicine or education); a study on how the scale of the system and, specifically, the number of users would affect the performance of the user classifiers; and the development of a solution that would allow the cooperation between systems that use the proposed architecture in order to better identify its users.

Acknowledgment

This work was supported by FEDER Funds through the "Programa Operacional Factores de Competitividade - COMPETE" program and by National Funds through FCT "Fundação para a Ciência e a Tecnologia" under the project: PEst-OE/EEI/UI0760/2014.

References

[1] A. Jameson, "User modeling meets usability goals," in International Conference on User Modeling, 2005, pp. 1–3.

[2] A. Madureira, B. Cunha, J. P. Pereira, I. Pereira, and S. Gomes, "An Architecture for User Modeling on Intelligent and Adaptive Scheduling Systems," in Sixth World Congress on Nature and Biologically Inspired Computing (NaBIC), 2014.

[3] B. Cunha, A. Madureira, and J. P. Pereira, "User modelling in scheduling system with artificial neural networks," Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on. pp. 1–6, 2015.

[4] A. Madureira, S. Gomes, B. Cunha, J. P. Pereira, J. M. Santos, and I. Pereira, "Prototype of an Adaptive Decision Support System for Interactive Scheduling with MetaCognition and User Modeling Experience," in Sixth World Congress on Nature and Biologically Inspired Computing (NaBIC), 2014.

[5] F. Karray, M. Alemzadeh, J. Abou, and S. M. N. Arab, "Human-Computer Interaction: Overview on State of the Art," Int. J. Smart Sens. Intell. Syst., vol. 1, no. 1, 2008.

[6] K. P. Baclawski, "Bayesian network development," in New Trends in Software Methodologies, Tools, and Techniques, IOS Press, 2004, pp. 18–48.

[7] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," Journal of Pharmaceutical and Biomedical Analysis, vol. 22, no. 5. pp. 717–727, 2000.

[8] J. E. Dayhoff and J. M. DeLeo, "Artificial neural networks: opening the black box.," Cancer, vol. 91, no. 8 Suppl, pp. 1615–35, Apr. 2001.

[9] A. Krenker, A. Kos, and J. Bešter, Introduction to the Artificial Neural Networks. Ljubljana, Slovenia: INTECH Open Access Publisher, 2011.

[10] B. Shneiderman, Designing the User Interface: Strategies for Effective Human-Computer Interaction, 1st ed., vol. 3rd. Reading, MA: Addison-Wesley, 1998.

[11] K. Al-Omar and D. Rigas, "Comparison of adaptive, adaptable and mixed-initiative menus," in 2009 International Conference on CyberWorlds, CW '09, 2009, pp. 292–297.

[12] E. Horvitz, "Principles of mixed-initiative user interfaces," in CHI '99 Proceedings of the SIGCHI conference on Human Factors in Computing Systems, 1999, pp. 159–166.

[13] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl, J. Havel, A. Lopez, E. M. Peña-Méndez, P. Vaňhara, A. Hampl, and J. Havel, "Artificial neural networks in medical diagnosis," J. Appl. Biomed., vol. 11, no. 2, pp. 47–58, 2013.

[14] A. Madureira, B. Cunha, J. P. Pereira, S. Gomes, I. Pereira, Jorge M. Santos, and A. Abraham, "Using Personas for Supporting User Modeling on Scheduling Systems," in Proceedings of the International Conference on Hybrid Intelligent Systems, 2014, pp. 279–284.

[15] M. Riedmiller and H. Braun, "RPROP - A Fast Adaptive Learning Algorithm," in Proceedings of the International Symposium on Computer and Information Science VII, 1992.