ORIGINAL RESEARCH

A robust smart device app assisting medical diagnosis

Vaios Papaioannou*[1], George - Peter K. Economou[2], Athanasios Tsakalidis[2]

[1] *Hellenic Open University, Greece*
[2] *Department of Computer Engineering and Informatics, University of Patras, Greece*

Received: June 4, 2016 **Accepted:** June 30, 2016 **Online Published:** July 19, 2016
DOI: 10.5430/air.v5n2p82 **URL:** http://dx.doi.org/10.5430/air.v5n2p82

ABSTRACT

A Medical Decision Making System (MDMS) has recently been implemented as an app (lication) for Android-based Mobile "Smart" Devices (AMSDs). Feed Forward Artificial Neural Networks (fANNs) is a major element of this MDMS that has been extensively tested in the field of Pulmonary Diseases (PDs). This MDMS fANNs have been taught by using real world patients' clinical data on a powerful Personal Computer; the MDMS modular structure and its fANNs' weights were then ported to AMSDs to be used as a part of wide-range applications software. The Android was chosen as the devices' Operating System (OS), due to its Linux kernel, touchscreen features, and open standards architecture.

Key Words: Mobile "smart" devices, Medical decision support systems, Artificial neural networks, Pulmonary diseases

1. INTRODUCTION

Nowadays a great deal of research and development projects have focused into exploring human understanding, reasoning, and expertise towards merging its efficacy in solving complex problems via standard "algorithmic-based" devices (tele-care medical devices, data mining methods, statistical analysis). Decision Making Systems (DMS)[1,2] can be the platforms of such endeavours; on the other hand, Medical DMS (MDMS) can play an essential role in supporting doctors of medicine (MDs) and also other medical experts or trainees in their diagnoses.[3–5] Additionally, MDMS can be remotely utilized by doctors of medicine, in order to gain accuracy over their judgement upon an unfamiliar disease. Finally, paramedics and related professionals can utilize MDMSs in order to better provide their patients with proper care while transporting them to medical centres.

A working group of medical and software experts developed a creative, modular and very accurate MDMS, in the area of Pulmonary Diseases (PDs) some time ago,[4] and is still being thoroughly tested in a University Hospital Clinic with real world patients. There are specific requirements concerning diagnosis and treatment of human diseases that are fulfilled by guidelines and specific data flows. Because of an increase in incidents of Tuberculosis and Lung Cancer there has been an initial development of a series of reconfigurable MDMSs for PDs.[4,5] The general and flexible structure of the MDMS, composed of layers of different feed forward Artificial Neural Networks (fANNs), is capable of being adjusted to other areas of either medical interest or where human knowledge and reasoning prevails, simply by providing applicable learning patterns.[1]

The MDMS presented in this paper, used actual clinical data to train its layers' fANNs. Through early and advanced experiments it demonstrated its efficiency of correctly classifying symptoms and PDs (reaching 83% out of one thousand potential non learnt PD clinical cases). Past similar wide-range testing showed efficiency up to 98% out of a combination of one hundred non learnt PD cases,[5] suggesting that the ratio

* **Correspondence:** Vaios Papaioannou; Email: vaios@sch.gr; Address: Hellenic Open University, Greece.

will ultimately augment.

This MDMS was ported as a mobile "smart" device app (lication),[6] by means of the wide-spread Android Operating System (OS). The OS choice insures a great variety of industrial devices (AMSDs) available for further porting, due to its huge acceptance and adaptability. Also, since the programming tools, the MDMS ported architecture, and the standpoint behind similar devices all share the same principles, it is surmised that diverse OS MSDs will be fit to host future MDMS porting, too.

Presently, the research team behind the whole project is in the process of contacting Medical and Research Institutions so as to supply a number of "smart" devices, equipped with the MDMS, to PD Clinics throughout Greece. Our hope is to be able to face issues such as geographic and regional incongruities regarding living standards, income, nutrition, and people physical exercise habits. Should this step be completed, the MDMS shall expand to cover another part of the human body diseases.

The contents of the paper are organized as follow: Section 2 deals with ANNs as the building stone of the MDMS; Section 3 presents the organization of the MDMS inputs; the fANNs composition is the subject of Section 4, whereas Section 5 further relates the porting of the MDMS to a "smart" device. Finally, Section 6 hosts conclusions and Section 7 proposed reference work.

2. ANNs AND ARTIFICIAL INTELLIGENCE

MDMS are starting to prove their value as they are used globally by more and more medical centres in actual cases. Artificial Intelligence (AI) techniques play a crucial role in Decision Making Systems (DMS). Based on them they can implement their inference engine or they can provide for the realization a knowledge base along with knowledge base induction rules.[7–9] Moreover, new methods are posed for comparing their performances by giving the necessary evaluation criteria.[8, 10, 11]

Today advances in MDMS based on ANNs are radical and multifaceted.[4, 5, 7, 8, 12–15] ANNs have many advantages such as improved speed factor, dynamic data storage, robustness, parallel searching, and generalization virtues so AI and medical experts utilize ANNs in MDMS implementations. ANNs may be implemented using custom hardware / FPGAs / VLSI chips or microprocessors or in software, general-purpose platforms.[16]

ANNs' accomplished to comply with a large set of applications from domain to general tasks' providing a sound ground of exploitation. ANNs are taught using a variety of learning algorithms that can also run on a Personal Computer using simulations. Lastly, supervised learning techniques were used on the teaching process in order to integrate well known symptoms of PDs to their artificial synapse weights (the way ANNs "learn").

Thus, the feed forward Artificial Neural Network architecture was elected to be the base of the MDMS due to the simplicity of its configuration, the vastness of proposed teaching methods, and to hardware implementation factors.[1] Also, as a consequence of its various slabs independence while being fed, processing, and outputting data, it is possible to separately employ each Neuron.[4]

3. INPUT ORGANIZATION

Developing knowledge-based environments for supporting MDs, is quite difficult because different doctors handle medical data (MD) in different ways, and the impact of each Medical Data both as a unit and as a whole is essential. Based on individual symptoms, new patients' cases can be built by categorizing and generalizing MD in ANNs synapse weights. It is a requirement by the Clinical Differential Diagnosis Methodology (CDDM) to singly weight all MD, judge intermediate results and process all data from the more broad to the more explicit. This is why CDDM should be followed by any MDMS[7, 11] and ANNs are the optimal implementation for CDDM.[5, 17]

In order to establish the problem domain boundaries, clinical experts in PDs, set a definite number of questions / inputs, the same way that they would be asked by MDs during patient examination. The questions contain findings about PDs' symptoms, *i.e.* Chest Pain, Dyspnoea, Fever, Cough, Sputum, Haemoptysis, and Wheezing. They may also include historical data gathered from physical exams. Additionally a large number of ANNs accepted the MD as input[18–20] in order to relate them to 35 Pulmonary Deceases and 12 major Pulmonary Deceases classes by distinct implementation levels.

Medical Data linked to symptoms findings for potential Pulmonary Diseases were been input to the system. ANNs were taught main elements such as the importance of determining specific pulmonary diseases, multiple pulmonary diseases diagnosis intervention, and higher fitness ordering. When assessing new cases by using appropriate unknown input patterns the whole MDMS was properly trained to exclude or confirm lethal PDs for a patient. Thus this procedure can be exercised for other diseases. In the same way it is possible to generalize main symptoms' classes to potential diseases. Throughout the method the ANNs' setup, remains the same.

4. The proposed fANNs structure

In our research it was proven that fANNs can be the most suitable basis of the proposed MDMS. It consists of three formation layers that follow the time propagation sequence of MD's (see Figure 1). In order to teach the fANN, we forwarded back propagation equations[19] and Kalman filtering of back propagation[20] learning algorithms. The Kalman filtering of back propagation[20] learning algorithm demonstrated better performance by requiring less initialization steps. Also there was quicker learning pace, higher integration rates and data management and improved accuracy. The learning times achieved introduced a spread between 21'-31', *i.e.* two thousand to three thousand learning cycles, and had to do mostly multiplications. floating point arithmetic, Binary inputs, thirty to forty four nodes, about three hundred input patterns, three slabs each (Neural Layers, NL) and an imposed of an average error level of 0.5% on learning accuracy, were the learning parameters implemented in the proposed fANNs.

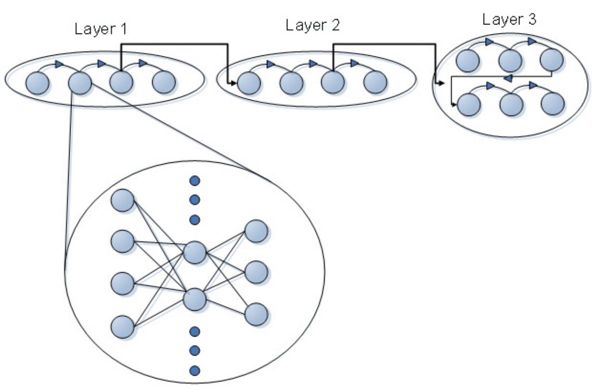

Figure 1. Layers: Composition of ANNs

4.1 First layer

A 4-level, 3-slab structured (*i.e.* having only one hidden NL) formation is used for our fANNs. It was chosen after a significant number of experiments.[4] For every symptom that is important (subjective MD) inputs are fed to the first level separately and in random order. Also, physical and historical examinations data (objective MD) are treated in the same level by two other same structured fANNs. All these fANNs have a number of input, hidden, and output Neurons according to MDs specification. Nevertheless, fANNs' outputs are the generalised classes of all PDs that can possibly match (percentages of similarity and compatibility to patterns they learnt).

In the second level eight 3-slab fANNs weight these outputs. This is achieved by matching pairs of symptoms with data from physical and historical examinations, enforcing the latter ones, as imposed by the appliance of the CDDM. The output is classified to PD classes. The output of the 2nd level is propagated in another 3-slab ANN on the 3rd level, combining all MD and keeping the same output scheme as the previous one.

Table 1 shows the MDs typical findings (*i.e.* inputs) for the "Fever" symptom. Column (Sig) implies the significance this symptom has for each PD class, and it is used only for the purpose of the ANNs' teaching by the MDMS developers and not the end users. The following columns denote Fever's duration ranking as small (SD), medium (MD), and long (LD), its evening (Ev) or daily (Dl) possible occurrence, and its existence after a patient's exposition to particular substances (Ex). Data concerning temperature below (<38°C) or above (>38°C) 38°C, the symptom followed by shivering (Sh), of periodical (Pr) or constant (Cn) nature, are also included.

Table 1. Findings of the Symptom "Fever" related to the PDs' classes

Classes of Pulmonary Diseases	Sig	SD	MD	LD	Ev	Dl	Ex	> 38°C	< 38°C	Sh	Pr	Cn
C.O.P.D.	**		*			*			*		*	
Tuberculosis	***	*	*	*	*	*		*	*		*	*
Interstitial PDs	**		*			*		*	*	*	*	*
Abnormalities of the Diaphragm												
Cancer of the Lungs	*	*	*	*		*			*	*		*
Disorders of the Mediastinum	**	*	*	*		*		*	*	*	*	*
Infection Diseases of the Lungs	***	*	*			*		*	*	*	*	*
Disorders of Pleura	**	*	*	*	*	*		*	*	*	*	*
Bronchial Asthma	*	*	*			*			*			*
Disorders of the Pulmonary Circulation	*		*			*			*			*
Occupational Disorders of the Lungs	*		*				*	*	*	*	*	
Respiratorial Deficiency	*	*	*	*		*		*	*			*
Abnormalities of the Chest Wall												
Non PDs												

The end results show the general tendency of the possible classes of PDs which are present by taking into account all MD and interactively (step by step implementation of the CDDM).

Therefore, clinical examinations are suggested by a two-slab ANN in the fourth level. Figure 2 depicts first Layer.

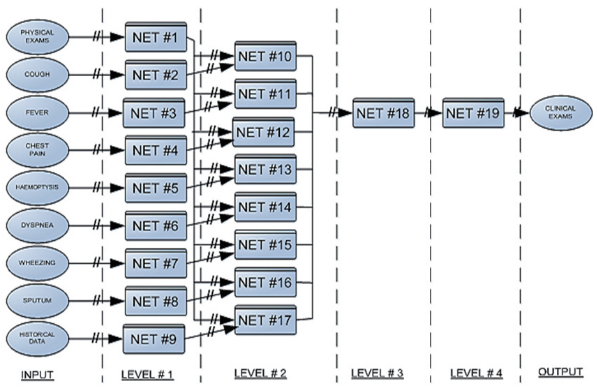

Figure 2. Formation of the first Layer

Based on this architecture it is possible to have transparency of the intermediate results that are generated based on the induction methodology. It is possible for an expert to intervene in every level of the process and according to his own opinion to select the most crucial diagnosis' aspects. The expert can make the final decision by examining all offered percentages of possible PDs' existences. Medical Decision Making Systems can be trained to offer different scenarios of prognoses, forwarding or dismissing more or less compatible diseases.

In addition, input values (findings of the application of the CDDM) can intentionally be left out of the induction process. Those inputs could be non-accurate data, a number of irrelevant symptoms / historical data findings, or even abnormal data, according to an MD training, experience, or personal opinion. Vice-versa, all similar data can be present to obtain the MDMS intermediate and final outputs.

4.2 Second layer

Following the establishment of the scheme for the first layer, there is another four-levelled, three-slab fANNs' formation in the second layer. The difference now lies on the fact that these fANNs do not handle classes of PDs but handle actual PDs with all their outputs, intermediary and final. Also, we have the addition of new inputs. These inputs correlate the final PDs' fitness percentages with the already computed 3rd level of the 1st layer. As a result the final diagnosis is enhanced by exercising a robust positive feedback in the 2nd layer fANNs. This is demonstrated by the improvement in outputs' accuracy (26%-39%) in the preliminary and more elaborated results.

Second layer can be seen as a patient's recurring visit to an MD, bringing along his/her clinical examinations' results. In general a significant, for the development of the disease, time elapses between those visits, giving the chance to the disease to further on its gestation cycle. Hence, the mapping of this procedure is obtained.

4.3 Third layer

All data of the former layers are fed into this layer. Turn-out data, assembled between examinations results' elapsed time and a new evaluation, are also taken into account as inputs. Therapy, medications' dosage, and prescribed timetables are scheduled as the final outputs.

As a result, friendliness, transparency and efficiency are achieved based on the established architecture. MD and clinical experts' interventions guarantee the system's good performance. So far the system has demonstrated results of total performance that reaches success of 83% as the already taught fANNs of the current MDMS generalized their feeds (1,000 new cases in total).

The remaining 17% of new cases fed to the decision making system can by no means be considered to be a failure that results to possible maltreatment due to PDs incorrect classification. Accurate clinical examinations are the first priority of MDs. The proposed MDMS overall performance is expected to be furthermore improved because of the use of the examinations' results.

5. "Smart" device application

Mobile computing and "smart" devices technology is widespread today, hence we designed and implemented an AMSD version of the MDMS: mHealth. As a result "smart" phones, tablets, and other mobile devices can be used by MDs to access diagnosis data. Figure 3 depicts the Android system architecture which is a prime, open source, and entirely customizable operating system for mobile devices programming. It is comprised by multi-purpose software for more basic, middleware, and key applications.

In order to develop such an application, software modules had to be established based on the new AMSDs capabilities and the former C / C++ implementations.

The AMSD includes both stand alone and client - server operation modes and also features an enhancement of the Human Computer Interface, as the old one was a basic window-based implementation.[4] The new interface was developed with compliance to the Android's system UI and framework, and the original code was ported using the Android Studio and it was linked to the newly developed UI structure.

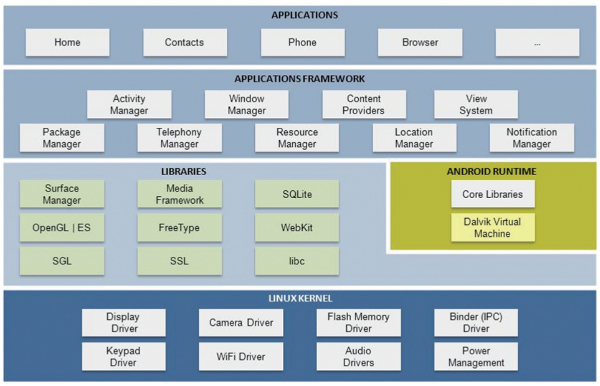

Figure 3. Android System Architecture

5.1 MDMS software architecture

Before the advent of "smart" MSDs, there was no flexibility regarding the implementation of an MDMS. In fact the proposed system was firstly developed on a stand-alone PC. Currently the improved mHealth application has the following options:

A. Stand-alone AMSD-based approach: "smart" devices can be considered mobile computing units with phone operation. mHealth runs solely on the mobile device, it is autonomous, and no network connections are required.

B. Client-Server based approach: the main MDMS application runs on a fast MDMS server; the "smart" device being mostly a mediator uploading MD data to the server in order to perform a diagnosis. Internet connection is required.

C. Ubiquitous Computing approach: the "smart" device is also connected to sensors in order to check a patient's health in real time. The MSD either runs mHealth (option A) or it is connected to a central server (option B).

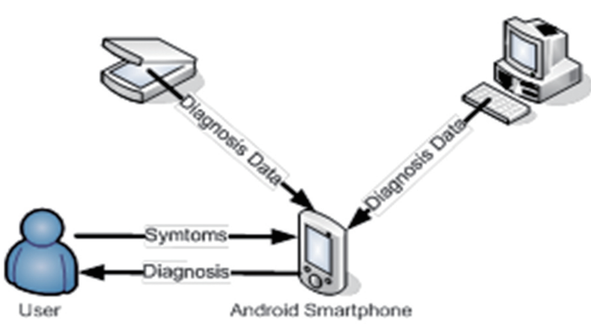

Figure 4. Stand-Alone MDMS Architecture

When using the stand-alone (see Figure 4) version of the android MDMS application, the input may be the patient's MD plus additional symptoms that may occur. The Android "smart" device can subsequently perform the diagnosis using the current symptom datasets combined with stored MDs. The output is the current diagnosis.

When running in Client - Server mode (see Figure 5), MDs can also upload the symptom MDs to a central MDMS server. The server can perform faster and more accurate diagnosis using historical data and up-to-date ANN weights.

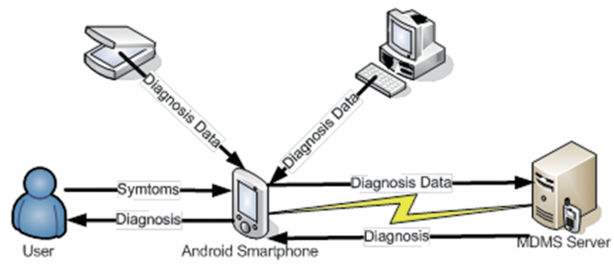

Figure 5. Client-Server MDMS Architecture

In future implementations, the MDMS server will be hosted in a specialized medical center. Expert/specialized MDs could also assess the diagnosis data, update the patient's history and finally suggest treatment for the patient. All this information can be communicated back to the end user MD using the AMSD application (see Figure 6).

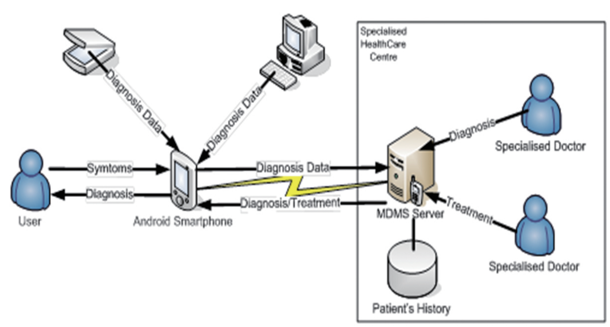

Figure 6. Future MDMS Cloud Implementation

5.2 User (MD) interface

By utilizing the Android OS, design features were kept to a minimal due to compatibility issues. MDs can select symptoms' data sets for loading in order to perform instant diagnosis by using option A or B of operation.[21] Figure 7 shows the Hierarchy Task Diagram for the MDMS MD interface, where the two main tasks are the input of the Diagnosis Data and the Diagnosis itself:

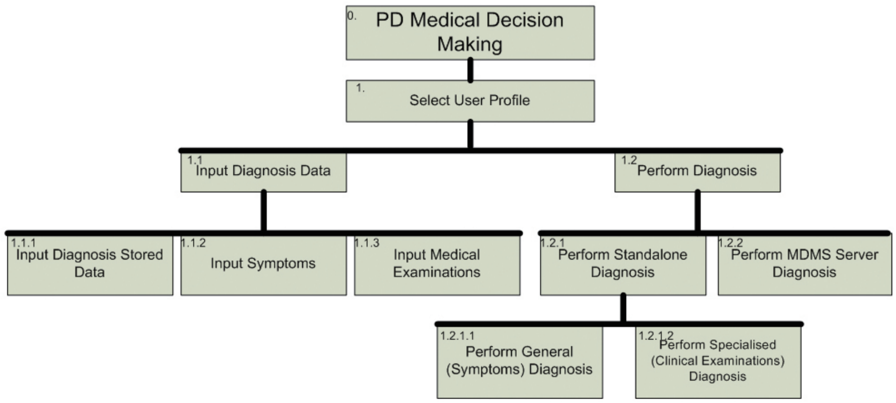

Figure 7. Hierarchical Task Diagram of the MDMS

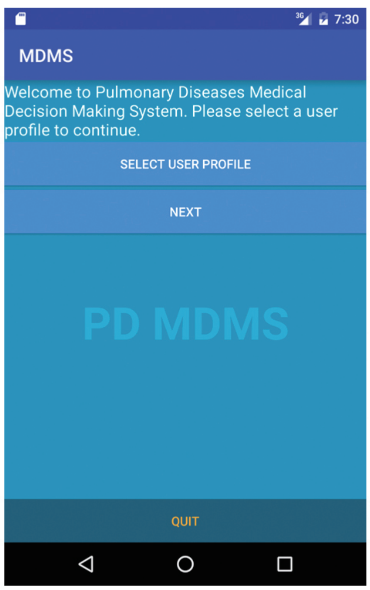

Figure 8. User Profile Selection

Initially an MD must select a user profile, which essentially contains all the required information for the patient (see Figure 8).

The required diagnosis data can be stored in the MSD or they can previously be uploaded to the device from the MD's PC. The MD has also to enter a series of symptoms (see Figure 9).

Thus, the MD is able to perform a general Diagnosis; for a specialized one, the clinical examinations data are also required. The MD can opt for a local or cloud mode of operation depending on performance restrictions (see Figure 10).

Again, the MD can either perform a general diagnosis based mainly on symptoms or a specialized diagnosis using additional clinical examinations data (see Figure 11(A)) so finally, the results are sent back to the MD's "smart" device (see Figure 11(B)).

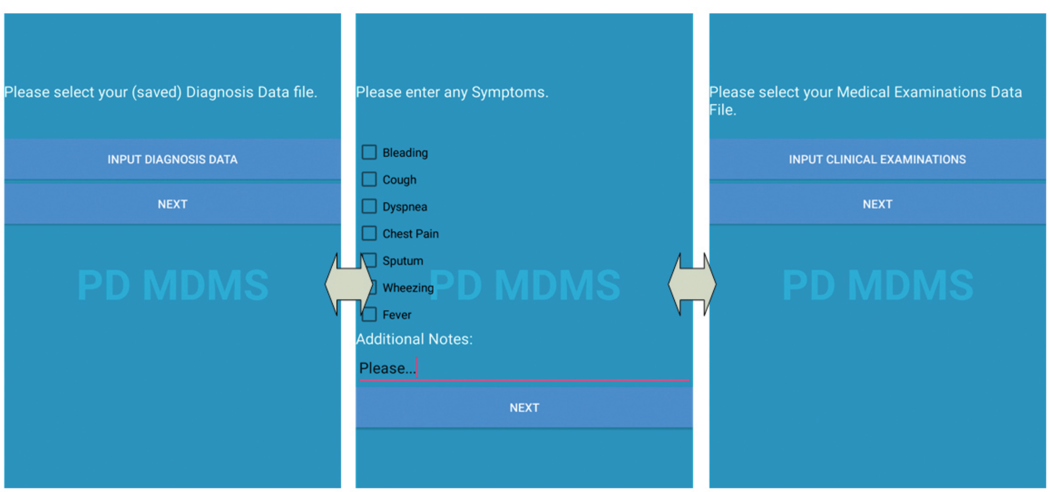

Figure 9. Diagnosis Data Input

The MD may restart the application as many times as he deems necessary in order to verify diagnoses' results. The required diagnosis data and results can be stored so that the user can skip parts of the data input process in the future or even compare conclusions by various CDDM options.

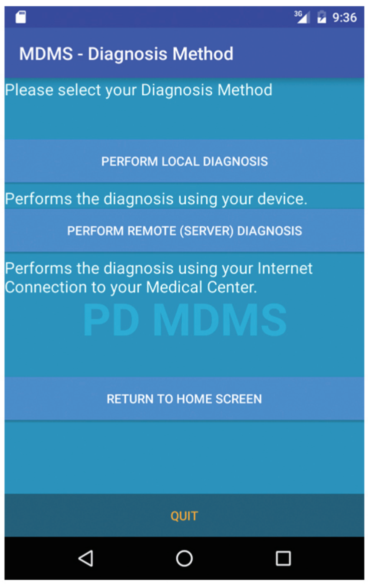

Figure 10. Local/Remote diagnosis Selection

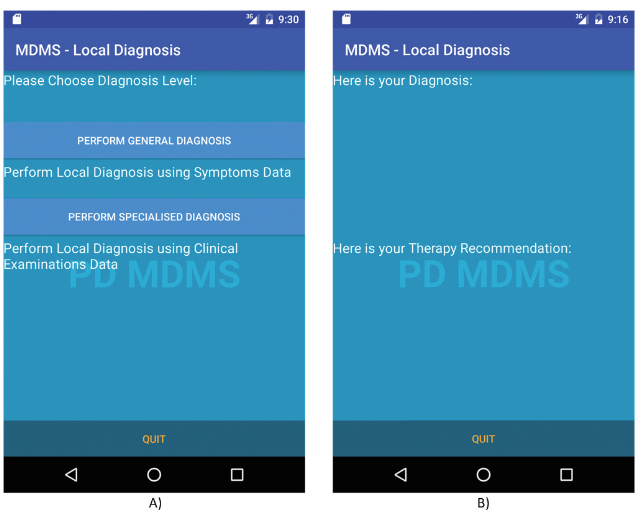

Figure 11. (A) General/Specialized Diagnosis Selection (B) Diagnosis and Treatment Recommendation

5.3 MDMS implementation results

Currently the architecture of the MDMS is ported to the Android OS. The Java programming language is the basis for Android applications, and its components are divided into four types: Content providers, Services, Broadcast receivers, and of course Activities. Activities are the main building blocks, representing a single screen with a user interface. Android Studio (see Figure 12) provides a complete integrated programming environment and necessary SDKs to develop apps and port the MDMS.[22] This SDK provides a compiler, debugger, and a device emulator as well as its own Java Virtual machine (Dalvik Virtual Machine - DVM). Android OS supports 2-D and 3-D graphics using the OpenGL libraries and supports data storage in a SQLite database.

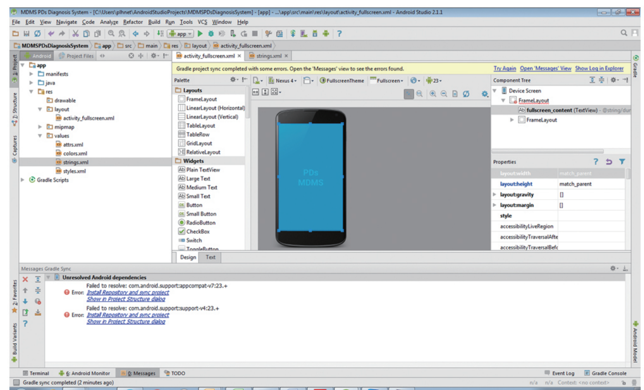

Figure 12. The Android Studio SDK

In order to build an Android application's graphical user interface we use a hierarchy of View and ViewGroup objects. For porting the MDMS, the Android Studio SDK IDE and NDK tools were used.[23] Figure 13 shows this programming hierarchy.

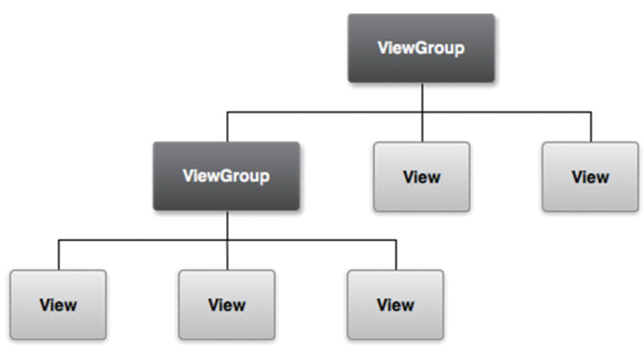

Figure 13. ViewGroup objects branching and containing other View objects

Currently, a large part of the Medical Decision Making System, is programmed in C language and it was ported to android using the Android NDK. This NDK allows a developer to program portable application parts by using C and C++, *i.e.* native code languages. The first version of the implementation included standard C portions of code aligned with the new android interface (see Figure 14).

```
if (f17 == 1)
  {
  for (i=0;i<pattern;i++)
    {
    farfree(*symptoms);
    farfree(*disease);
    symptoms++;
    disease++;
    }
  symptoms -= pattern;
  disease -= pattern;
  farfree(symptoms);
  farfree(disease);
  f17=0;
  }
```

Figure 14. Integration of C code in the Android app

In terms of performance, the original application was developed using something less that an Intel 486 CPU, benchmarked at 0.03 GFLOPS.

Today a typical desktop PC might achieve around 50 GFLOPS (100 GFLOPS for an Intel i7-based PC) whereas a typical medium to high end quad core "smart" device performs above 1GFLOP;[24] more or less 100 times faster than the original system setup. In terms of specifications, most Android devices are equipped with 2 GBs of RAM by using standard flash memory. I/O performance with Class 10 SD cards can reach data rates up to 10 MB/s, while Serial ATA III hard drives can reach up to 6,000 Mbit/s. Therefore, these benchmarks are comparable to many PC configurations and were deemed more than adequate for our purposes.

6. Discussion

In this project, we propose a novel Medical Decision Making System of modular structure which is developed on a basis of a composition of ANNs. This Decision Making System is applied on research areas where human experience is employed for the promotion of decisions. The architecture of this system can be adapted onto the structures of the application field and is arranged according to the classification of the input data. Those data are provided from the particular expert exactly in the form he/she operates on them. The hierarchical structure with which the system processes the input data devises for intermediate decisions, and promotes the final decision from the general to the more specific classes on which the area of interest can be divided. The system's operation eventually prompts for new data to be fed. These new inputs are supplied to the system in succeeding processing levels while time as a discrimination factor for some of the previously input data is being taken account of. The implemented system is being input data that have not undergone pre-processing or symbolic restructuring.

This system was adapted and thoroughly tested on the field of medicine with very good results. Because of the incomplete inputs of this area, a new representation of the learning patterns (used for teaching its ANNs) was created and hereby demonstrated.[4] This representation can be generally arranged during the learning procedure of ANNs and its main characteristic consists of manipulating incomplete input data for the teaching, testing, and employment of these networks.[5] The data components are input separately, their particular value is enhanced and pseudo inputs are set during the ANNs' learning procedure, smoothing their classification onto different output classes. Previous research efforts simply ignored those data or filled them statistically. In addition, the number of hidden layers and artificial neurons were investigated as a factor affecting the performance of ANNs as well as the importance of ANNs' weights. As a consequence, heuristic rules were suggested that consider the specification of their values and intervene to change those before and after the networks' convergence. However, describing again all those features in detail are beyond the scope of this paper.[4]

Finally, major progress was noted in terms of implementation since we have managed to port the desktop ANN architecture into Android-based mobile smart devices. The feed-forward ANN's artificial neurons are approximated and circuits for their integration in C programming are furnished. They cover a large spectrum of specifications, offering some advantages like the retargeting of the system's implementation in mobile devices, processing speed, a generic development platform, independence from the manufacturer and a new parametrical library of leaf-, medium-, and advanced-cells. Based on this architecture it is now feasible to generalise the use of mobile MDMS systems to large number of user groups who can work remotely possibly in cooperation with large medical centres. It may also prove valuable for eHealth applications and wide health care monitoring of sensitive populations. From the doctors point view it was very important to improve usability aspects of the proposed system (see Table 2) with the introduction of the mobile environment which was aligned with the MDs workflows, offering consistence, integration and standardisation of procedures:

Table 2. MDs Application evaluation results

	Evaluation Level (1: Low; 2: High)				
	1	2	3	4	5
Ease of Use	1		5	6	1
Diagnosis Accuracy			5	7	1
Operation Speed			2	9	1
Work Flow		3	4	5	1
Standards			5	6	1

The user interface no longer required the pre-processing of input data by IT specialists. MDs can input the required data directly to the device. Furthermore it has now made possible for health monitor devices to directly feed their data into the system allowing real time monitoring.

For more complex application domains, the client-server architecture was suggested. So far the android app was introduced to MDs of the Regional and University Hospital of Patras as beta testers (see Figure 15) and is demonstrated in IT fairs and industrial exhibitions with success.

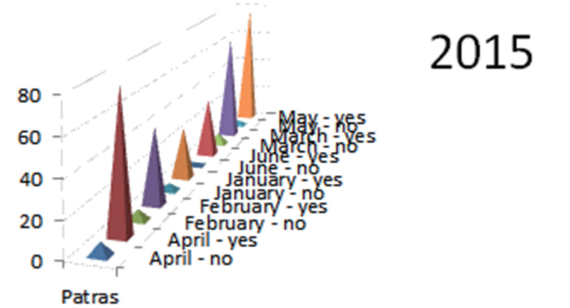

Figure 15. Number of cases diagnosed by the Android app per location per month

7. Conclusions and future work

In this project we have successfully managed to map the proposed ANNs' architecture, into the establishment of a powerful multi-purpose MDMS that was implemented as a "smart" device application. This implementation helped MDs in human diseases' diagnosis using the optimal structuring for an efficient MDMS. Special care was given to achieve great performance results and make it re-targetable.

Preliminary results on the whole project are very encouraging. Future work suggests the intensifying of the MDMS with the augmentation of its learned medical data and the evolution of its inference engine. This will lead to a multi-purpose MDMS as the starting point of other medical diseases induction diagnosis and furthermore to other areas of human expertise. Another area of concern could be the direct scanning of hardcopy medical examinations through the "smart" devices capabilities so as to input them in the form of low-level pictures and make this tool more powerful and autonomous (*e.g.* to fully cover in-home care schemes).

The MDMS "smartphone" device is planned to be distributed to a number of PD Clinics throughout Greece. A major plan is ready to be realized and the MDMS development team is in the process of contacting international Medical and Research Institutions to raise the necessary funds and sponsorships. It is this team belief that different issues concerning the way of living for PD patients will so be properly addressed. "Smart" devices will be the vehicle by which valuable MD will be further gathered and patients' conditions be additionally valuated.

References

[1] Al-Absi HRH, Abdullah A. Hybrid Intelligent System for Disease Diagnosis Based on Artificial Neural Networks, Fuzzy Logic, and Genetic Algorithms. Informatics Engineering and Information Science, Springer - Verlag, Germany. 2011.

[2] House WC. Decision Support Systems: A Data-Based, Model-Oriented, User-Development Discipline, Petrocelli Books Inc., Mc Graw Hill. 1991.

[3] Saxena S, Burse K. A Survey on Neural Network Techniques for Classification of Breast Cancer Data. Int. J. of Eng. and Adv. Tech. 2012; 2(1): 234-237.

[4] Economou GPK, *et al*. Decision Support Systems for Tele-Medicine Applications. Research Studies Press Ltd., Hertfordshire, UK. 2004.

[5] Dimopoulos KG, Sotiriades P, Economou GP, *et al*. Tele-medicine decision support platforms: an application. In Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (pp. 312 - 321), World Scientific and Engineering Academy and Society (WSEAS). 2006.

[6] Available from: `http://en.wikipedia.org/wiki/Smartphone`

[7] Chaudhry B. Computerized Clinical Decision Support: Will it Transform Healthcare? J. Gen. Intern. Med. 2008; 23(1): 85-87. PMid:18095051. `http://dx.doi.org/10.1007/s11606-007-0432-9`

[8] Marozas V, Jurkonis R, Kazla A, *et al*. Development of teleconsultations systems for e-health. Studies in health technology and informatics. 2004; 105: 337-348. PMid:15718622.

[9] Umbaugh SE, Moss RH, Stoecker WV. Applying Artificial Intelligence to the Identification of Variegated Coloring in Skin Tumors. IEEE Eng. in Med. and Biol. 1991; 10(4): 57-62.

[10] Moustafa AA. Performance Evaluation of Artificial Neural Networks for Spatial Data Analysis. Cont. Eng. Sc. 2011; 4: 149-163.

[11] Anooj PKN. Clinical Decision Support Systems: Risk Level Prediction of Heart Disease using Weighted Fuzzy Rules and Decision Tree Rules. Cent. Eur. J. of Comp. Sc. 2011; 1(4): 482-498. `http://dx.doi.org/10.2478/s13537-011-0032-y`

[12] Nilsson E. Neuronal networks involved in low back pain, University of Gothenburg, Sweden. 2012.

[13] Patel JL, Goyal RK. Applications of Artificial Neural Networks in Medical Science. Curr. Clin. Pharm. 2007; 2(3): 217-26. PMid:18690868. `http://dx.doi.org/10.2174/157488407781668811`

[14] O' Leary TJ, Mikel UV, Becker RL. Computer-Assisted Image Interpretation: Use of a NN to Differentiate Tubular Carcinoma from Sclerosing Adenosis. Mod. Path. 1992; 5(4): 402-405.

[15] Poli R, Cagnoni S, Livi R, *et al*. A neural network expert system for diagnosing and treating hypertension. Computer. 1991; 24(3): 64-71. `http://dx.doi.org/10.1109/2.73514`

[16] Watanabe T, Kimura K, Aoki M, *et al*. A Single 1.5-V Digital Chip for a 106-synapse NN. IEEE Trans. on NN. 1993; 4(3): 387-393.

[17] Ahsan MR, Ibrahimy MI, Khalifa OO. A Step towards the Development of VHDL Model for ANN based EMG Signal Classifier. In International Conference on Informatics, Electronics & Vision (ICIEV) (pp. 542 - 547), University of Dhaka, Bangladesh. 2012.

[18] Baptista D, Morgado-Dias F. A Survey of Artificial Neural Network Training Tools. Neural Comput. and Applic. 2013: 609-615.

[19] Lippmann RP. An Introduction to Computing with NN. IEEE ASSP Mag. 1987; 4(2): 4-22. http://dx.doi.org/10.1109/MASSP.1987.1165576

[20] Scalero RS, Tepedelenlioglu N. A Fast New Algorithm for Training Feedforward NN. IEEE Trans. on Sig. Proc. 1992; 40(1): 202-210.

[21] Available from: http://en.wikipedia.org/wiki/Cloud_computing

[22] Chang G, Tan C, Li G, *et al*. Developing mobile applications on the Android platform. In Mobile multimedia processing (pp. 264 - 286), Springer-Velag, Germany. 2010.

[23] Available from: http://developer.android.com

[24] Dongarra JJ, Luszczek P, Petitet A. The LINPACK Benchmark: past, present and future. Concurrency and Computation: practice and experience. 2003; 15(9): 803-820. http://dx.doi.org/10.1002/cpe.728