MEDINFO 2010
C. Safran et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-588-4-218

MEDAL: Measuring of Emergency Departments' Adaptive Load

Edward Vitkin[a,b], Boaz Carmeli[a], Ohad Greenshpan[a], Dorit Baras[a], Yariv Marmor[b]

[a] *IBM Haifa Research Lab, Haifa University Campus, Haifa 31905, Israel*
[b] *Technion - Israel Institute of Technology, Technion City, Haifa 32000, Israel*

Abstract

We propose an innovative approach for measuring real-time operational load within emergency departments. Medical informatics, operations researchers, and other decision makers in the health care field have yet to come to an agreement regarding standardized matrices for measuring operational load within emergency departments. As a result, it is difficult to develop methods and approaches for reducing operational load. We propose a flexible framework based on neural networks. These networks can calculate user-tuned load value, based on a set of well-defined operational and clinical indicators. The operational load value is calculated by learning the weights of the raw operational indicators within a particular emergency department.

Keywords:

Operations research, Workload, Emergency Department, Machine learning, Neural networks (computer)

Introduction

The rising cost of healthcare services has been a subject of mounting importance and much discussion worldwide. Ample explanations have been proposed, yet regardless of their cause, rising costs impose pressures on healthcare providers to improve the management of quality, efficiency, and economics for their organizations. Significant attention has been given to the question of how to reduce this cost in the healthcare domain Hospitals are one of the major players in the provisioning of health services and within hospitals, emergency department (ED) overcrowding has been perhaps the most urgent operational problem [1, 2, 3]. Overcrowding in hospital EDs leads to excessive waiting times and repellent environments, which in turn cause: (1) poor service quality (clinical, operational); (2) unnecessary pain and anxiety for patients; (3) negative emotions (in patients and escorts) that sometimes lead to violence against staff; (4) increased risk of clinical deterioration; (5) ambulance diversion; (6) patients leaving without being seen (LWBS); (7) inflated staff workload; and more [4].

In order to reduce the occurrence of overcrowding in hospital EDs and optimize ED operations, we need to understand what the current crowding load level is. This means that it is necessary to decide how load on various resources should be defined, to whom it should be presented, and how it should be demonstrated. This task is difficult for several reasons. First, establishing which parameters contribute to the load is complex and subjective. Second, even once the parameters are established, assigning a level of contribution to each one is difficult, due to the varying conditions in each hospital, and to the perceptions of different management teams. Third, the ED is a complex environment that involves various types of entities (e.g., physicians, nurses, patients, executives); each of whom may define the load function differently. Clearly, load has to be a usage-dependent function. Fourth, the definition of load changes from time to time and needs to be updated periodically. Fifth, due to the large number of quickly-changing events and factors that are critical to saving lives, displaying a real-time snapshot of the load in the system is critical. This snapshot must be displayed in such a way that decisions based on current load can be made quickly and easily.

Our Contribution

In this paper, we present Measuring the Emergency Department Adaptive Load (MEDAL), a flexible and adaptive system that enables the calculation of user-tuned load in emergency departments, using various types of inputs and defined load functions. Tuning for the needs of the user makes the system user-specific, resulting in a load score that directly reflects the high level of input. The system is based on artificial neural networks [5] and enables the following: (1) a static mechanism for an explicit definition of load functions; and (2) a dynamic learning mechanism that enables the system to adapt to the perceptions of users with no explicit load function definition.

The dynamic learning mechanism allows the system to present different load values for the same objective situation. This is particularly useful for understanding the difference in operational load perception by physicians, nurses, patients, and the ED management.

MEDAL is a highly adaptable load measurement tool that, by scarifying rigid definition for high flexibility, allows users to analytically compare various situations, and to reach informed decisions regarding the appropriate steps to take in order to reduce the ED load.

The paper is organized as follows: the Methods section describes the main idea of our proposed solution, along with the technical details on how the system was built. In the Results section, we describe our main results achieved from implementing load profiles. The Discussion section deals with the benefits of using our system in diverse and dynamic environments for process optimization and planning. We then conclude with a summary of the approach, the benefits of system use, and directions for future research and development.

Methods

The calculation of operational load in emergency departments and its presentation are highly important, yet difficult tasks, for improving efficiency. To that end, we developed MEDAL, a flexible framework for measuring subjective load, using iterative user feedback. MEDAL can be adapted for any user preference and view of the load in specific environments. We describe the main idea on which the design of the system was based and developed, and then elaborate on the technical implementation details.

Main Idea

MEDAL is a flexible and configurable framework for measuring ED load. By default, our framework receives an extensive set of raw indicators as input. These indicators were reported in the literature [6] as a consensus for the set of measures that are important for the calculation of ED load. Moreover, this set of indicators can easily be modified according to user needs. Also, our framework receives operational events from the existing ED infrastructure, processes them, and calculates the time-specific input indicator values. The core of our framework is a learning neural network mechanism that enables the following main features: (1) Users can modify the set of basic indicators collected from the ED infrastructure. (2) Users can configure the system with any kind of load function. (3) In cases where the load function cannot be defined explicitly, our framework provides an adaptive mechanism that learns the desired load function autonomously. This is done by learning from the user feedback on the calculated load, while viewing snapshots of ED states and the corresponding calculated load. (4) The system also offers advanced capabilities for tracking the origin of the load status and for understanding its cause at different levels of granularity. Moreover, the system can provide specific alerts regarding the high load values at various predefined internal points, even if the total load in the system is low. (5) The system enables comparison among various calculations of load, according to the perception of different roles in the ED. We further demonstrate and discuss these features in the Results section.

We now elaborate on the technical details that served as a basis for the development of the proposed system.

Implementation Details

Load Function Definition

Our framework provides a simple way to define any explicit load function, based on canonical indicators and the display of that function's behavior during different time frames. This process is described below and shown in Figure 2.

Learning Unknown Load Functions with Neural Networks

Due to the complexity of the ED environment, explicitly defining the load function is often not useful. Therefore, machine learning techniques should be harnessed to solve these issues. We chose to use the artificial neural networks [5] as our basic mechanism. These systems are flexible for composition, adaptive over time, meaningful for the user, and enable the definition of complex relationships (e.g., nonlinear) between inputs and outputs.

We first provide a theoretical background on neural networks and then explain how these were harnessed to solve the problem raised in this paper.

Neural Networks – Theoretical Background

Artificial neural networks [5] are mathematical representations of complex mathematical functions. They are composed of units named perceptrons (Figure 1a), and arranged as a multi-layered feed-forward network (Figure 1b), in which the outputs of one layer are the inputs of the next layer. This type of learning machine was inspired by the brain structure. These machines are successfully used in many applications, such as pattern classification, dimensionality reduction, and function approximation [7, 8, 9]. Because of the origins of the machines' design, the nodes in such networks are often called *neurons.* The machines' greatest advantage is their simplicity (in both representation and learning). In addition, the number of required training examples (that is relative to the network structure) is not high compared to other machine learning solutions.

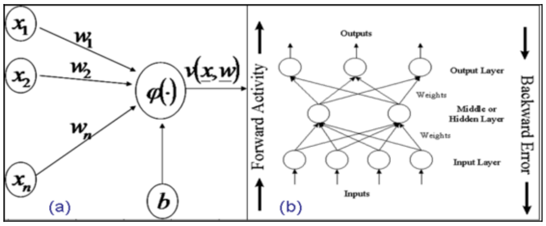

Figure 1- (a) single perceptron; (b) Multi-layer network

Each perceptron is composed of n inputs, $x_1, x_2, ..., x_n$, n weights $w_1, w_2, ..., w_n$ and an activation function $\varphi(\cdot)$. The output of the unit is $v(\underline{x}, \underline{w}) = \varphi(\underline{x}^T \underline{w})$, where $\underline{x} = (1, x_1, ..., x_n), \underline{w} = (b, w_1, ..., w_n)$. Examples of activation functions are sign ($\varphi(u) = sign(u)$), linear function ($\varphi(u) = u$), and logistic function ($\varphi(u) = 1/(1 + e^{-u})$). The type of activation function affects the ability of the network to learn and is application-dependent. The units in different layers are connected in a feed-forward style to determine the network structure (see Figure 1b). The exact structure is also application-dependent, and in many cases, domain knowledge can help to determine this structure.

Given a training set of the form $(\underline{X}_i, y_i)_{i=1}^{l}$ in which $\underline{X}_i \in \Re^n$ is the input to the network and $y_i \in \Re$ is the expected output (or target function) of the network, the back propagation algorithm [5] can be used to find a set of weights that minimizes the mean square error (MSE) between the expected output and the current calculated output. There are two types of learning—offline (or batch) learning, and online learning. In offline learning, the entire training set is given in advance. In each iteration of the back propagation algorithm, all of the examples are taken into account when updating the weights. In online learning, the examples are given one after the other, and each learning iteration depends on the current example only. Online learning is typically used when the environment changes over time, and when the network is trained to fit those changes.

Harnessing Neural Networks to Calculate Load in the ED

To demonstrate the advantages of our methodology, we built a neural network that represents the collective knowledge presented in the exhaustive indicators review paper [6]. We used it as our basic network for the load calculation (see Figure 2).

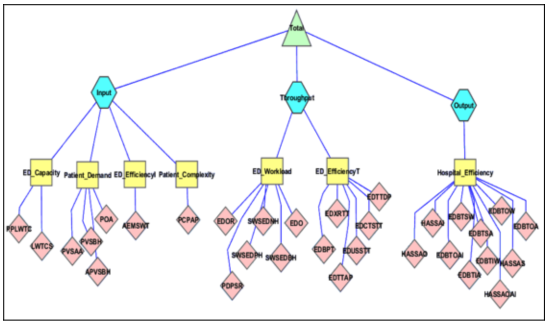

Figure 2- ED Neural Network View: The triangle is a total neuron, hexagons are the stage neurons, rectangles are the concept neurons, and diamonds are input indicator neurons

For the set of inputs, we took the set of indicators that were defined in the review paper and combined and ranked them in a hierarchical manner. The hierarchy in the network consists of four main layers: 1) ***Indicators Layer:*** This layer is the set of inputs, and consists of 31 nodes corresponding to 20 indicators that appeared in the review paper [6]. Several indicators were spliced to match their definitions. For example, indicator "*ED Throughput time*" was spliced into two nodes: one for admitted patients and one for discharged patients. Others were omitted due to the lack of appropriate input simulation data. We made minor modifications to the network, as suggested in the paper, based on additional data we could collect by using a simulator presented by Sinreich and Marmor [1]. This simulator, also used by Wasserkrug et al [10], is based on a canonical ED model, which was generated based on real-life observations within numerous hospitals. 2) ***Concepts Layer:*** The 31 indicators in the indicators layer are connected (each indicator to a single concept) to the following six concepts: patient demand, patient complexity, ED capacity, ED efficiency, ED workload, and hospital efficiency. The hospital capacity concept was omitted due to the lack of appropriate data. The *ED efficiency* concept was divided into two sub-concepts to serve the input and the throughput separately. This modification was made to keep the tree-like structure of the network. We will explain the importance of that structure later in this paper. 3) ***Operational Stages Layer:*** The seven concepts are connected (one concept to each operational stage) to the following three operational stages: input, throughput, and output. 4) ***Load Score Layer:*** This layer is the output layer and consists of a single node representing the total load function score.

As described above, we can learn the target function using either the offline or online method. Both approaches require knowledge of the true target function values on some set of input vectors. This means that we have to present each such vector to the expert user and receive the desired function value in return. However, this flow cannot be used for two main reasons. First, input vectors are often too long for human perception and embedding. Second, The desired value of the target function cannot be explicitly calculated. In the following paragraph we describe how we handle both of these challenges.

Instead of presenting the input vector itself, we present current ED status using a patient-centric dashboard (Figure 3). The patient-centric dashboard provides ED staff with information about the status of each patient in the ED. Patient status is usually presented as a row within a table-like view. This approach is commonly used in EDs around the world. By working with the patient-centric dashboard, the ED staff gains total insight regarding the operational ED load. We use this insight to enter feedback into the network, thus enabling it to learn the required load function. The user provides feedback by using the dedicated feedback buttons. For example, if the user *feels* that the represented load is far below the desired value, he pushes the larger "+" button, increasing the current load by 10% increments. A similar update process (+1%, -1%, -10%) works for other feedback buttons. Clearly, such a feedback system can be easily implemented in any existing dashboard without significant changes.

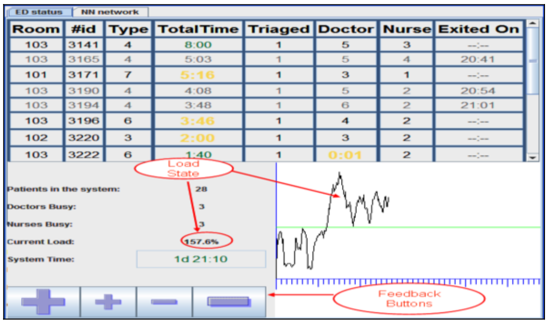

Figure 3- Dashboard snapshot. Load value (black line) is calculated as the percent of the average (green line)

Tracking Load and Bottlenecks

In our system, every neuron has an explicit operational meaning. Our neuron network design keeps the tree-like neuron hierarchy instead of the usual all-to-all connections. This allows each neuron to preserve its operational meaning during the learning process. Conserving the tree-like structure allows the user to track the current load back into the network and to gain a deeper understanding of the current load status (Figure 4). Moreover, we can get an alert from any hierarchy level in the system if a certain neuron becomes overloaded. For example, if the current system load is only 40% of the average but the CT room is overcrowded due to lack of personnel, the appropriate neuron's status will reach the high mark, and can alert the user, provided the neuron was preconfigured accordingly. As a result, the ED manager may react by temporarily adding to the CT room staff.

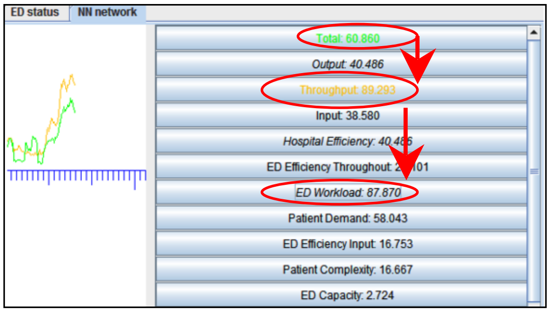

Figure 4- Tracing load: Green line indicates total load, orange line indicates throughput; the rise in the total load was clearly caused by increasing the throughput input neuron. We can trace it further and deduce that the peak in throughput was caused by elevation of the ED workload concept neuron

Comparing Different Views on ED Load

Our framework allows dynamic learning based on feedback from different user groups. We can calculate and present several different load views for the same objective situation. For example, measuring the current ED load as perceived by doctors, nurses, and patients, or even by a single individual such as the ED manager, could have interesting applications. Having an option for defining a subjective load function that best reflects the actual load experienced by a given user group could be very useful, as well. In the next section, we investigate and demonstrate the differences in load perception for three user groups: ED Doctor, ED Nurse, and ED Patient, as an example of this application.

Results

To demonstrate the system's ability to reflect subjective load, we identified three possible group types: nurse, doctor, and patient. The user profile reflects operational load as it is being experienced by a given group, or even by a specific individual within the ED. User profiles can be statically defined by fixing weights on relevant neurons, or preferably, by dynamically learning the user profile. Dynamic learning involves capturing user feedback from a specific user or user group associated with a relevant profile. To achieve this, we generated operational data using a simulator tool [1] and defined subjective target load functions for three group types. Nurse and doctor target load functions were defined as the average occupation ratio during a time period. Patient target load function was defined as the ratio of a patient's waiting time to the patient's total staying time in the ED (Figure 5).

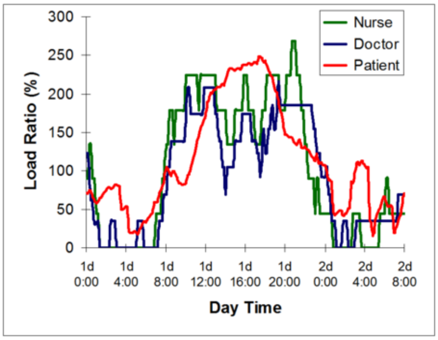

Figure 5- Simulated nurse, doctor, and patient profile behavior, where 100% is the average daily load

Figure 6 presents results from running the system multiple times and dynamically learning the three groups' profiles. Each profile is comprised of the weights of major neurons learned by the system.

Raw Indicator \ Profile	Nurse	Doctor	Patient
Patient Volume standardized for Bed Hours	42%		76%
Summary Workload standardized for ED Bed Hours	18%	45%	
ED Bed Placement Time	12%	30%	8%
ED CT service Turnaround Time	28%	25%	16%

Figure 6- User profile comprised of major raw indicator weights learned by the system

We can see that all three profiles show reasonable behavior when comparing time of day and when comparing the profiles to one another. When the system is overloaded, all users feel it. However, the load experience is different for each group. For example, doctors need to stay later than nurses at the end of the day and to close all open cases. Thus, the load on them decreases later than it does for nurses. On the other hand, triage, served by nurses, is the first station in the patient flow. Hence, the nurses' operational load starts earlier. These examples demonstrate that there are indeed different weights on the neurons, emphasizing the need for subjective load scores for the same objective ED state.

Discussion

In this paper, we introduce MEDAL, a flexible system for the estimation and measurement of load in hospital emergency departments. The development of MEDAL stemmed from the understanding that ED load is influenced by a large number of factors that are difficult for both humans and machines to consider together. Solberg et al., [6] gathered 74 experts to collect a set of 113 measures affecting the load in EDs, of which 38 were selected through a discussion and rating process. This set of measures was divided into three categories (input, throughput, and output) and seven concepts (patient demand, ED capacity, patient complexity, ED efficiency, ED workload, hospital efficiency, and hospital capacity). While Solberg provides a comprehensive set of load measures and concept categories, he does not suggest a straightforward way to use these measurements in real-life scenarios. Our experience shows that establishing a standard operational load model that fits all EDs is not practical, due to the inherent differences among them.

MEDAL is a flexible and adaptive framework for load calculation that comes with a set of initial measures (the measures that appear in the abovementioned paper) and a set of load functions. The framework can be easily enhanced with additional measures and load functions. Moreover, the system includes a mechanism to learn the load function from the user, in cases where this could not be explicitly defined. This learning mechanism is used to define user profiles, each with its own view and requirements, from the calculation of load in the ED environment. Figure 5 shows the load calculated for three possible profiles. It shows that all measures estimate the load quite similarly, meaning that when there is a burden on the hospital, all relevant entities (e.g., staff members, patients) are affected. However, this system allows users to distinguish between the load pressures felt by these entities. Measuring load is a crucial step for optimizing ED operation. This measure of load can be used in various ways. First, it can be used to optimize the staff routing inside the ED at times of high load levels. This could of course reduce the burden on hospital departments. Using the offline method, load calculation can be used to provide detailed planning for different staff members in the ED (e.g., nurses, physicians). The MEDAL approach may also be applicable to other hospital wards and even to other industries. Providing a highly adaptable measuring tool, may prove useful in many situations in which global agreement about measured indicators is out of reach. Our work on this issue is beyond the scope of the current paper.

Conclusion

In this paper, we presented MEDAL, a novel, flexible, adaptive framework for user-specific load definition and calculation. The major advantages of the presented system are its flexibility to fit specific user needs, and its ability to handle and learn from highly undefined data that represents subconscious user perceptions. The implementation of the method is straightforward and can be easily integrated in any current ED dashboard system. The received user-specific load score can also be used for operation optimization and for providing advice to ED personnel. Moreover, measuring operational load, while taking into account user-specificity, is an interesting research direction in and of itself.

References

[1] Sinreich D, and Marmor YN. 2005. Emergency department operations: the basis for developing a simulation tool. IIE Transactions 37:233–245.

[2] Hall RW. Patient Flow: Reducing Delay in Healthcare Delivery. Springer. 2006

[3] Green LV. Using Operations Research to reduce delays for healthcare. In Tutorials in Operations Research, eds. Zhi-Long Chen and S. Raghavan, 2008, 1–16. Hanover, MD: INFORMS.

[4] Derlet RW, and Richards JR. Overcrowding in the nation's emergency departments: complex causes and disturbing effects. Annals of Emergency Medicine 2000 35:63–68.

[5] Haykin S. Neural Networks: A Comprehensive Foundation, 1999 2nd edn, Upper Saddle River, NJ Prentice Hall, Inc.

[6] Solberg LI, Asplin BR, Weinick RM, Magid DJ. Emergency department crowding: consensus development of potential measures. Ann Emerg Med. 2003 Dec;42(6):824-34.

[7] Khan J et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine 2001, 7, 673-679.

[8] Hinton GE and Salakhutdinov RR. Reducing the Dimensionality of Data with Neural Networks. Science 2006, 313 (5786), 504.

[9] Hornik K, Stinchcombe M, and White H. Multilayer feedforward networks are universal approximators. Neural Netw. 2, 5 (Jul. 1989), 359-366.

[10] Wasserkrug S, Greenshpan O, Marmor YN, Carmeli B, Vortman P, Basis F, Schwartz D, and Mandelbaum A. InEDvance: Advanced IT in Support of Emergency Department Management. in NGITS 2009.

Address for correspondence

Boaz Carmeli: boazc@il.ibm.com