

Article

# “PhysIt”—A Diagnosis and Troubleshooting Tool for Physiotherapists in Training

Reuth Mirsky <sup>1,2,\*</sup> , Shay Hibah <sup>1</sup>, Moshe Hadad <sup>1</sup>, Ariel Gorenstein <sup>1</sup> and Meir Kalech <sup>1</sup> 

<sup>1</sup> Department of Software and Information Systems Engineering, Ben Gurion University, Negev 84105, Israel; shayhibah@gmail.com (S.H.); moshh@post.bgu.ac.il (M.H.); gorenсар@post.bgu.ac.il (A.G.); kalech@bgu.ac.il (M.K.)

<sup>2</sup> Computer Science Department, University of Texas at Austin, Austin, TX 78712, USA

\* Correspondence: dekelr@post.bgu.ac.il; Tel.: +1-512-8501984

Received: 2 January 2020; Accepted: 25 January 2020; Published: 28 January 2020

**Abstract:** Many physiotherapy treatments begin with a diagnosis process. The patient describes symptoms, upon which the physiotherapist decides which tests to perform until a final diagnosis is reached. The relationships between the anatomical components are too complex to keep in mind and the possible actions are abundant. A trainee physiotherapist with little experience naively applies multiple tests to reach the root cause of the symptoms, which is a highly inefficient process. This work proposes to assist students in this challenge by presenting three main contributions: (1) A compilation of the neuromuscular system as components of a system in a Model-Based Diagnosis problem; (2) The *PhysIt* is an AI-based tool that enables an interactive visualization and diagnosis to assist trainee physiotherapists; and (3) An empirical evaluation that comprehends performance analysis and a user study. The performance analysis is based on evaluation of simulated cases and common scenarios taken from anatomy exams. The user study evaluates the efficacy of the system to assist students in the beginning of the clinical studies. The results show that our system significantly decreases the number of candidate diagnoses, without discarding the correct diagnosis, and that students in their clinical studies find *PhysIt* helpful in the diagnosis process.

**Keywords:** model based diagnosis; applications; diagnosis; physiotherapy

## 1. Introduction

When a patient contacts a physiotherapist (PT) regarding a problem in the peripheral nervous system or muscular system, the usual cues are either in terms of motion or sensory abilities. The patient can report some difficulty in performing a specific movement or a sensory problem such as numbness or tingling. A weakened motion is indicated by an observation on the muscles, while a defected sensation is indicated by an observation on the dermatomes. These reports are the symptoms of the patient. Based on the reported symptoms, the PT hypothesizes the possible reasons that could explain the patient’s complaints. These reasons are called diagnoses. To discriminate the root cause among the possible diagnoses, a troubleshooting process is executed in which the PT performs a series of tests that are meant to disambiguate between the correct diagnosis and the rest. This approach is usually time consuming and can be ineffective, especially in the case of trainee PTs with little experience. For example, some clinicians move back and forth between their original and revised hypotheses to come up with a final diagnosis [1].

This paper presents an AI-based tool—*PhysIt*—that enables an interactive visualization and diagnosis to assist trainee physiotherapists (The system can be viewed using the following link: <http://www.ise.bgu.ac.il/PHYSIOTHERAPY/Homepage.aspx>).

The first step to create such a tool is a compilation of the neuromuscular system as components of a system in a Model-Based Diagnosis problem. Our approach is based on modeling the physiotherapy

diagnosis process as a model-based diagnosis (MBD) problem [2–5]. MBD relies on a model of the diagnosed system, which is utilized to simulate the expected behaviour of the system given the operational context (typically, the system inputs). The resulting simulated behaviour (typically, the system outputs) are compared to the observed behaviour of the system to detect discrepancies that indicate failures. The model is then used to pinpoint possible failing components within the system. In the physiotherapy domain, the observed system behaviour is the patient's weakened motion or defected sensation. The system model is a model of the human body, such as the nervous system, the muscles, the dermatomes etc., as well as the connections between them. A diagnosis is the human body component(s) that does not function well. Modeling this problem as an MBD enables solving it by applying off-the-shelf MBD algorithms. This compilation is the first main contribution of this paper.

Second, the *Physit* system computes the diagnoses based on the observations and then it operates a troubleshooting algorithm to assist the PT to choose informative tests and finally identify the root cause of the patient's complains. This is the second contribution of this paper and it can be decomposed into the different features of the *Physit* system:

The first feature of *PhysIt* is an interactive graphical model of anatomical entities. To this aim, we used expert knowledge to define the important entities that are required to clinically diagnose patients. In particular the tool focuses on nerve roots, nerves, muscles and dermatomes. Using this domain representation, we implemented an interactive inference to visually present the relationships between the entities.

The second feature of *PhysIt* is a framework to assist a trainee PT with the diagnosis process. This framework proposes the next recommended test to perform given the current state and observations given from the patient and previous tests. This framework utilizes the MBD algorithm described earlier.

The third feature of *PhysIt* is a troubleshooting process in which the root cause of the symptom is recognized. This is done by adapting an iterative probing process from the MBD literature [6], in which tests are iteratively proposed to the PT in order to eliminate redundant diagnoses.

There is a huge research on diagnosis in medicine. Most of the works propose frameworks and algorithms utilizing different diagnosis approaches, such as knowledge-based [7], data driven [8] and model-based [9]. Many of the works even run experiments on specific medical problems. As far as we know, no previous work presents a comprehensive tool for trainee PTs that includes visualisation, diagnosis and troubleshooting. Our work does not present new diagnosis or troubleshooting methods, but it utilizes previous model-based methods to present a tool that helps trainee PTs in the diagnosis process, by applying anatomical model visualization, diagnosis and troubleshooting.

The third contribution of this paper is a comprehensive performance analysis and a user study to evaluate *PhysIt*. The performance analysis was performed on both simulated cases and scenarios depicted by the domain experts, that are common cases in anatomy exams. We examined the diagnosis process in terms of accuracy, precision, waste costs and the AUC of the health state [10,11]. Our results show that the tool always finds the correct diagnosis and that the troubleshooting process can significantly decrease the number of candidate diagnoses, and thus facilitates trainee PTs. Our user study included simulations of a physiotherapy diagnosis process performed by physiotherapy students. The students were given different levels of access to our system, and were then requested to answer a questionnaire in order to evaluate the experience with the system. The study shows that our system was perceived as helpful in choosing the tests to perform and in improving the diagnosis process.

To summarize, this paper contains an exploratory research on the application domain of physiotherapy diagnosis. It consists of a new theoretical representation, based on MBD literature, to model a PT diagnosis process, and then utilizes this representation to propose an educational tool for PTs in the beginning of their clinical studies. This research proposes a thorough analysis of the proposed tool, both by testing its mathematical and theoretical capabilities, and by testing it empirically on use-cases used in physiotherapy exams and in a user study with PTs in training.

The flow of the paper is as follows: in the next section we detail the related work, then in Section 3 the architecture and interface of the tool will be presented. Section 4 describes technical details about the different parts of *PhysIt*: the model, the diagnosis algorithm and the troubleshooting process. In Section 5 the diagnosis and the troubleshooting processes will be evaluated and in Section 6 the user study will be presented. Section 7 concludes this work.

## 2. Related Work

In Section 2.1 we present the main approaches for diagnosis and specifically model-based diagnosis methods in medicine. Then in Section 2.2 we depict troubleshooting approaches. Finally in Section 2.3 the contributions of our work are presented in the light of previous work.

### 2.1. Diagnosis

Diagnosis approaches are typically divided into three categories: data-driven, model-based, and knowledge-based. Data-driven approaches are model free. The online monitored data is used to differentiate a potential fault symptom from historically observed expected behaviour, e.g., via Principle Component Analysis [12]. Model-based approaches [13–15] typically use reasoning algorithms to detect and diagnose faults. The correct/incorrect behaviour of each component in the system is modeled as well as the connections between them, and the expected output is compared to the observed output. A discrepancy between them is exploited to infer the faulty components. Knowledge-based [16] approaches typically use experts to associate recognized behaviours with predefined known faults and diagnoses. A similar partition is proposed by Wagholikar et al. [17], which survey paradigms in medical diagnostic decision support, dividing most works into probabilistic models (Bayesian models, fuzzy set theory, etc.), data driven (SVM and ANN) and expert-based (rule-based, heuristic, decision analysis, etc.).

The decision on the best approach is obviously dependent on the domain knowledge. If we have enough data on past processes of the system then probably we would prefer to use data driven approaches, on the other hand if the system can be represented by rules, designed by experts, then a knowledge-based approach is preferred. Finally, if we can formally model the system, then a model-based approach will be appropriate. Next, we present relevant research and elaborate on MBD approaches within the context of medical systems.

In this paper we focus on a model-based approach, since we used expert physiotherapists which helped us to model the upper part of the human body which is innervated by the nerve roots C-3 to T-1. For a survey of knowledge-based approaches in medicine we refer the reader to [7]. There are additional surveys that address knowledge-based approaches in specific medical fields as breast cancer diagnosis [18] and medical expert systems for diabetes diagnosis [19]. Data driven approaches are very common in medicine, Patel et al. [8] and Tomar et al. [20] survey many of these approaches, specifically Kourou et al. survey machine learning approaches for cancer prognosis [21]. Data Mining techniques are used to label specific conditions such as Parkinson Disease [22] or Diabetes [23].

There are several approaches in MBD. All are relevant also to diagnosis in medicine [9,24]. They differ in the way the domain knowledge is represented. Obviously, in many cases the model is determined by the type of knowledge we have. Consistency-Based Diagnosis (CBD) assumes a model of the normal behaviour of the system [2,3]. Causality models describe a cause-effect relationships. There are two diagnosis approaches to deal with causality models, set-covering theory of diagnosis [25] and abductive diagnosis [26,27]. A third way to model a system is by a bayesian network [28], where the relations between the components are represented by conditional probability tables. Given evidence, an inferring process is run and produces a diagnosis with some probability. We survey each one of these approaches next.

General Diagnostic Engine (GDE) is an algorithm to solve the CBD problem [2]. This algorithm proceeds in two steps: (1) First, it finds conflicts in the system by using assumption-based truth maintenance system (ATMS) [29]. A conflict is a set of components, which when assumed healthy the

system theory is inconsistent with the observation. (2) Then the GDE computes the hitting sets of the conflicts, where each hitting set is actually a diagnosis. Downing [30] proposes a system which extends GDE to deal with the physiological domain. For this, Downing extends the GDE to cope with (1) dynamic models by dividing the time to slices and solve the diagnosis problem for each slice, and with (2) continuous variables by representing the variables qualitatively. He gives some examples from the physiological domain such as diagnosing the stages of acidosis regulation. Also Gamper and Nejdil [31] cope with the temporal and continuous behaviour of medical domain. They propose to represent the temporal relationships between qualitative events in first-order logic and then, given observations, they run CBD algorithm to diagnose the system. They run experiments on a set of real hepatitis B data samples.

CASNET [32] is one of the pioneer causal models in medicine. It describes pathophysiological processes of disease in terms of cause and effect relationships. The relationships between the pathophysiological states are associated also with likelihood to direct the diagnosis. CASENT even links a therapy recommendation to the diagnostic conclusion. INKBLOT [33] is an automated system which utilizes neuroanatomical knowledge for diagnosis purposes. The model includes hierarchy anatomical model of the central nervous system where the cause effect relationships describe the connections between and damages and manifestations. Also Wainer et al. [34] describe a cause-effect model where the causes are disorders and effects are the manifestations. They extend the diagnostic reasoning, using Parsimonious Covering Theory (PCT) [35], to deal with temporal information and necessary and possible causal relationships between disorders and manifestations. They demonstrate their new algorithm on diagnosis of food-borne diseases.

The problem of diagnosis, often shown as a classic example of abductive reasoning, is highly relevant to the medical domain [36]. As shown in previous papers [27,37], abduction with a model of abnormal behaviour is much better way than consistency-based to deal with medical diagnosis. However, not always such knowledge is easy to obtain, since it requires experts to model not only the normal behaviour, but also how a component behaves in each one of its abnormal cases. Obviously, this knowledge helps to focus on more meaningful diagnoses, but it is difficult to obtain. Pukancová et al. [38] focus on a practical diagnostic problem from a medical domain, the diagnosis of diabetes mellitus. They formalize this problem, using information from clinical guidelines, in description logic in such a way that the expected diagnoses are abductively derived. The importance of taking into consideration temporal information in medicine has been previously recognized. Console and Torasso [39] discuss the types of temporal information which can be represented by causal networks, and they use a hybrid approach to combine abductive and temporal reasoning for the diagnosis process.

Bayesian networks (BN) is a probabilistic model using for diagnosis in various domains such as vehicles [40], electrical power systems [41] and network systems [42,43]. BN describes conditional probabilities between the components; given evidence (observations), an inference algorithm is used to compute the probability of each healthy component to propagate the evidence. A classical work in the medical domain is the Pathfinder, which is designed to diagnose lymphatic diseases using Bayesian belief networks. It begins with a set of initial histological features and suggests the user additional features to examine in order to differentiate between diagnoses [44,45]. Velikova et al. [46] presents a decision support system that can detect breast cancer based on breast images, the patient's history and clinical information. To address this goal, they integrate the three approaches to model the knowledge: consistence-based, causal relationships and Bayesian network. MUNIN is a causal probabilistic network for diagnosing muscle and nerve diseases through analysis of bioelectrical signals, with extensions to handle multiple diseases [47,48].

## 2.2. Troubleshooting

Mcilraith [49] presented the theoretical foundation for sequential diagnosis, where a probe is a special case of a *truth test*, which is a test checking if a given grounded fluent is true. This process

is similar to clinical evaluation, where the PT performs tests to discriminate between diagnoses. Physiotherapy clinical evaluation is also similar to the active diagnosis problem [50,51], which is the problem of how to place *sensors* in a discrete event system to verify that it is diagnosable, given a set of observations. A very similar problem is the sensor minimization problem [52], where *observers* are placed on particular events to make sure the system is diagnosable and the number of observers is minimized [53]. None of these works reasons about scenarios in which the true state of a component can be masked by other components to return inconsistent values upon probing. Mirsky et al. [54] discuss a similar problem, where the presence of a component in the true hypothesis can be inferred by probes, but they do not reason about a scenario where a specific probe returns one value, while its true state is the opposite value, as discussed in our work.

To reduce the number of hypotheses, McSherry et al. [55] propose a mechanism for independence Bayesian framework. The strategy they propose searches for lower and upper bounds for the probability of the leading hypothesis as the result of each test is obtained. Rather than a myopic minimum entropy strategy they propose efficient techniques for increasing the efficiency of a search for the true upper or lower bound for the probability of a diagnostic hypothesis.

Algorithms for minimizing troubleshooting costs have been proposed in the past. Heckerman et al. [56] proposed the decision theoretic troubleshooting (DTT) algorithm. Probing and testing are well-studied diagnostic actions that are often part of a troubleshooting process. Probes enable the output of internal components to be observed, and tests enable further interaction (e.g., providing additional inputs) with the diagnosed system, providing additional observations (e.g., observing the system outputs). Placing probes and performing tests can be costly, and thus the challenge is where to place probes and which tests to fix the system while minimizing these costs. The intelligent placement of probes and the choice of informative tests have been addressed by many researchers over the years [6,57–62] using a range of techniques including greedy heuristics and information gain, which is calculated by comparing the entropy of the hypothesis set before and after a probe is placed [57]. This approach allows for a clear and straightforward mathematical representation of complex systems that can be analyzed to provide completeness, soundness and other computational guarantees. Due to this reason, in this paper we follow the information gain approach and adapting it to handle hidden fault states of the components in the system.

### 2.3. Summary and Our Contribution

In the light of previous work we can see that medical diagnosis is a highly researched area. Most of the previous works can be divided into three approaches: model-based, data-driven and knowledge-based. The main model-based approaches are consistency-based, causal reasoning and Bayesian networks. In many cases the diagnosis method depends on the information available to the researcher. Not always experts exist to help in designing a rule-based system or a model, nor there is enough historical data which can be exploited to generate a classifier or to learn probabilities.

In this work we used expert PTs to generate a model of the the upper human body which is innervated by the nerve roots C-3 to T-1. Unfortunately, we did not have historical data to learn the probabilities of each component to damage nor the conditional probabilities between components. As far as we know, this knowledge is not modeled for neuro-muscular diagnosis in physiotherapy for this part of the body. Therefore, our diagnosis and troubleshooting algorithms assume uniform distribution. Obviously, this can be easily changed given probabilistic knowledge.

The main contribution of this paper is a consistency-based diagnosis and troubleshooting tool, especially for trainee PTs, that includes: (1) An interactive visual model, which helps a PT to see the connections between the nerve roots, nerves, muscles and dermatomes. (2) A diagnosis process which assists the PT to generate hypotheses, given the patient's symptoms. (3) A troubleshooting process that proposes the PT a sequence of tests to discriminate the hypotheses and focus on the correct one. To the best of our knowledge, this is the first tool that combines these components to assist trainee PTs.

### 3. Architecture and Interface

The system is constructed of several components in a client-server framework, which is designed to allow high usability and applicability for PTs in their clinical evaluations. These components are depicted in Figure 1. A relational database (DB) is implemented using MSSQL to store the connections between the different entities. The server side is ASP.NET and it connects directly to the DB. After a connection is established, an Entity Framework is used to map the tables into objects, to allow easier and faster manipulations on the data. Finally, the client side is implemented using HTML, Javascript and JSON. The system’s home page is web-based, which allows the user to navigate to one of the following modules:

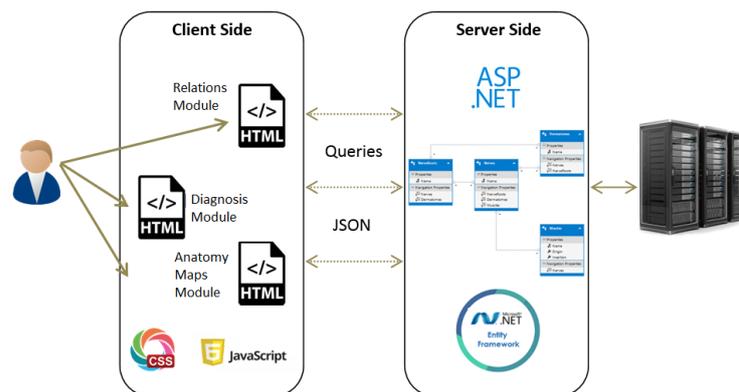


Figure 1. Framework description of the system.

**Maps** The purpose of this module is to provide visualization of the anatomical entities in the human body, while allowing to focus on different structures. This module contains an inner navigation bar, to choose between one of several views: root nerves, nerves, muscles, dermatomes and relations. All maps but the latest focus on different component types and present the names of the relevant components on an illustration. The relations map is a hierarchical representation of the connections between the different entities. It is similar to the relationships graph in the relationships module, but its visualization focuses only on a specific component at a time. An example of this representation is shown in Figure 2. Clicking on one of the nodes constructs a graph of the dependencies of this node.

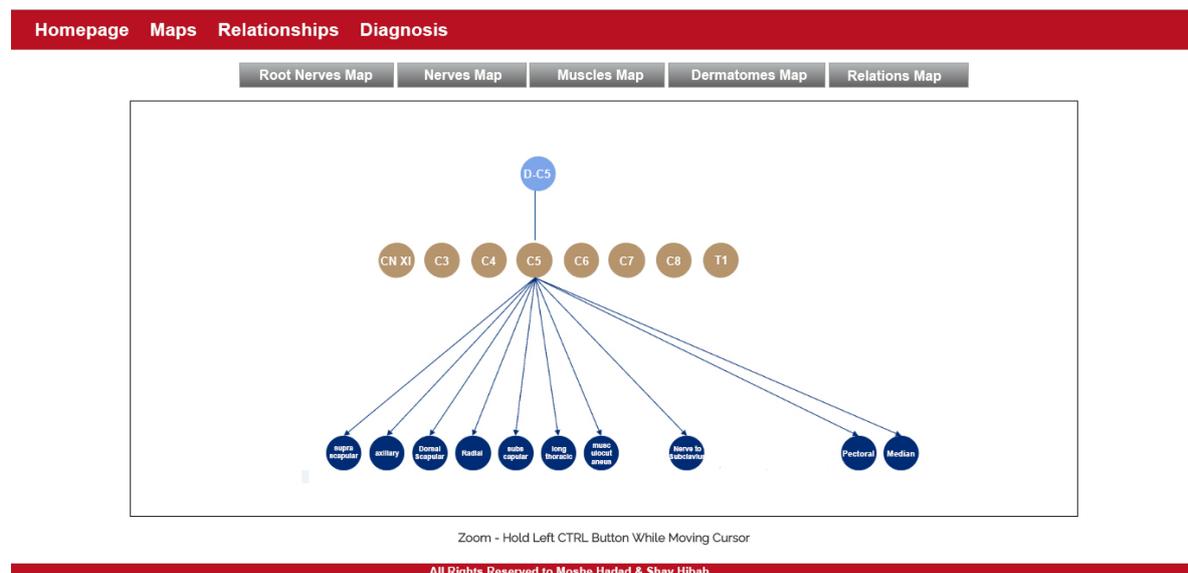


Figure 2. The maps module.

**Relationships** The purpose of this module is to allow a thorough investigation of the relations between the different components of the body. The navigation through the different components can be performed either by using a drop-down list and choosing a specific item from it, or by clicking directly on a node in the graph. The complete relationship graph is presented in Figure 3. This module enables to dynamically navigate from one node to another, a feature which allows the PT to investigate causal connections.

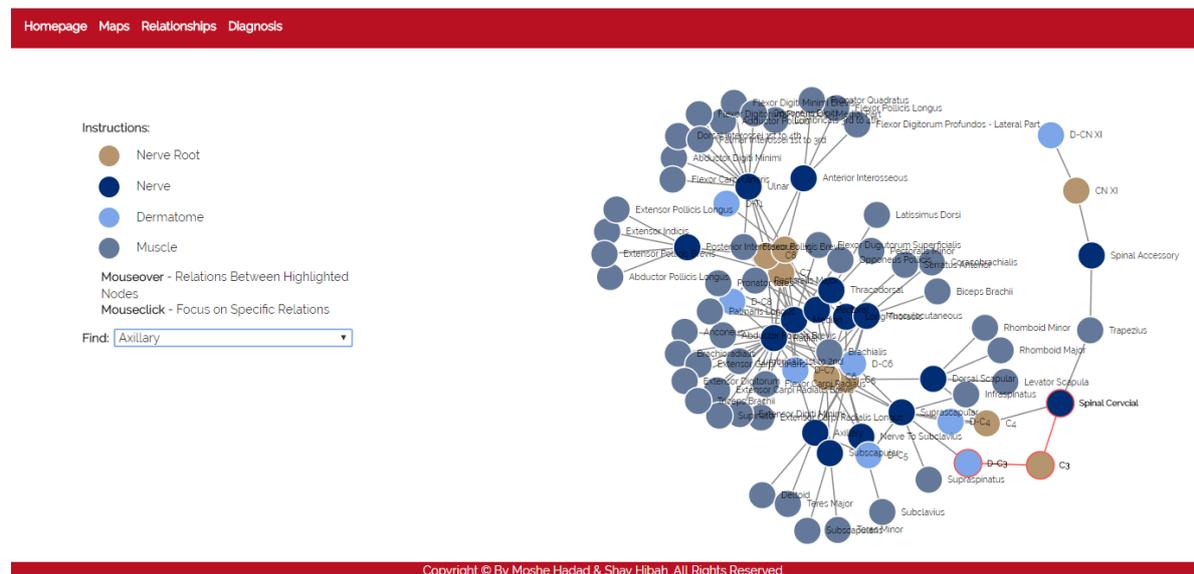


Figure 3. The relationships module.

**Diagnosis** The purpose of this module is to diagnose the patient, given a list of symptoms. The initial screen of this module is shown in Figure 4. This screen contains two lists of possible symptoms—muscles and dermatomes—which can be added by the PT. When the PT finishes adding initial symptoms, a click on the “Diagnose” button will trigger a recommendation for the next component to check, and then the system requests the PT to update whether the test passed or failed (the component works as expected or not). At any point, the PT can choose to stop this process and receive a list of the remaining diagnoses.

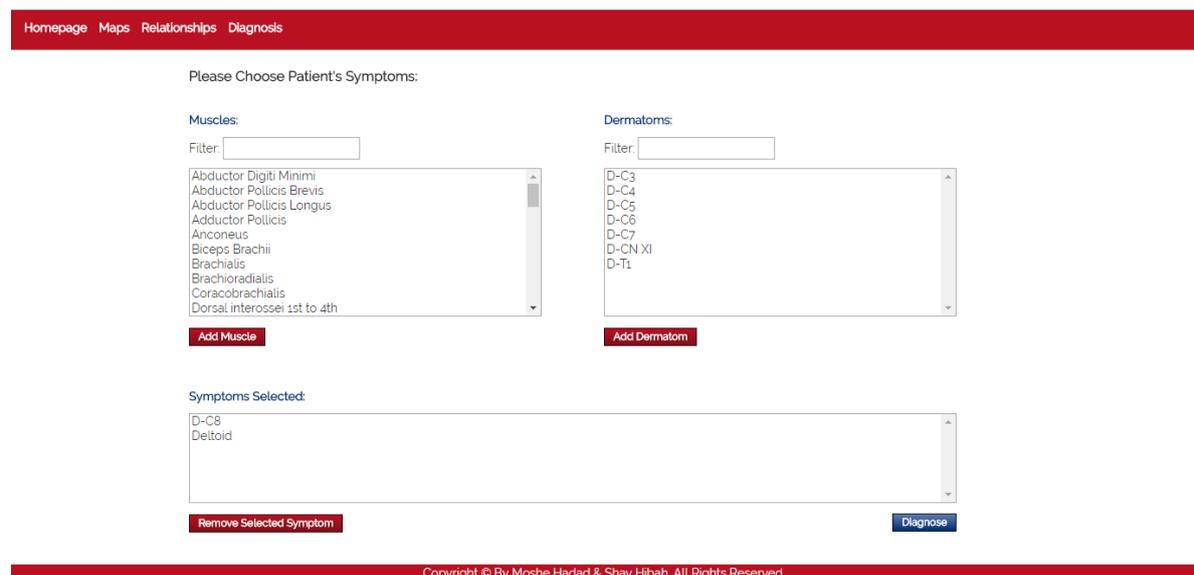


Figure 4. The diagnosis module.

#### 4. Technical Description

In this section we will describe technical details about the different parts of *PhysIt*. Specifically, we will describe the model we used (Section 4.1), the diagnosis algorithm (Section 4.2) and the troubleshooting process (Section 4.3).

##### 4.1. Model Description

The first feature of *PhysIt* is a model of the entities involved in a physiotherapy diagnosis. We elicited a model of the upper human body which is innervated by the nerve roots C-3 to T-1, or from head to the upper part of the torso. We acquired the information through interviews with senior PTs and data gathering from physiotherapy graduate students. The entities we modeled are *Nerve roots*, *nerves*, *muscles* and *dermatomes*. The relations between the different entities are described in Figure 5:

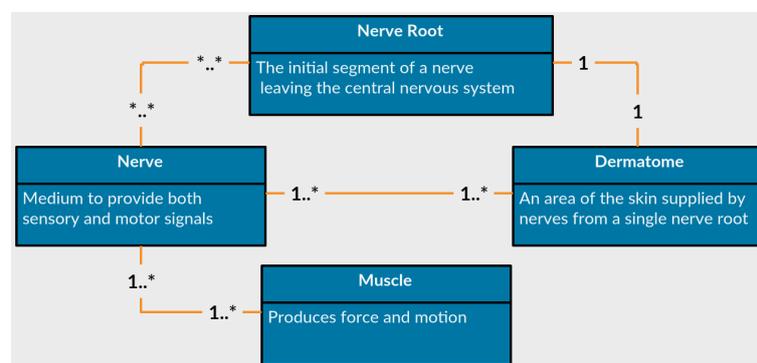


Figure 5. Anatomical entities represented in the diagnosis models.

**Nerves** are the common pathway for messages to be transmitted to peripheral organs. A damaged nerve can cause paralysis, pain or numbness in the innervated organs.

**Nerve Roots** are the initial segments of a nerve affected by the central nervous system. They are located between the vertebrae and process all signals from the nerves. A damaged nerve root can cause paralysis, weakened movement, pain or numbness in vast areas of the body.

**Muscles** are soft tissues that produce force and movement in the body. A damaged muscle can cause weakness, reduced mobility and pain.

**Dermatomes** are sensory areas along the skin, which are traditionally divided according the relevant nerve roots that stimulate them. A damaged dermatome is usually caused by a scar or burn and can cause pain, numbness or lack of sense.

As can be observed from the list of entities, some of the symptoms overlap each other. Tingling sensation at the tip of the index finger can be related either to a problem in a nerve root labeled C-7, to a burn in the relevant dermatome DC-7, or to a problem in a median nerve. Since this work only focuses on damages to the peripheral nervous system or muscular system, we assume that a symptom that is expressed in a dermatome is a signal to a damage in either a nerve root or a nerve. Moreover, the tingling sensation is a cue related to a dermatome, but the dermatome itself is assumed to be healthy. We will elaborate more on this issue later.

The anatomical data for creating this model was elicited by us using physiotherapy students and approved by faculty members with clinical experience. We mapped the relations between all pairs of entities in terms of functionality. A fragment of the elicited relational model is presented in Figure 6. The nodes represent the different components, the colors indicate their type and an edge indicates that one node influences or influenced by the other node associated to it.



**Definition 2** (Diagnosis). *A set of components  $\Delta$  is a diagnosis if*

$$SD \wedge \bigwedge_{c \in \Delta} \neg h(c) \wedge \bigwedge_{c \notin \Delta} h(c) \wedge OBS \not\vdash \perp$$

There may be multiple diagnoses for a given problem. A common way to prioritize diagnoses is to prefer *minimal diagnoses*, where a diagnosis  $\Delta$  is said to be *minimal* if no proper subset  $\Delta' \subset \Delta$  is a diagnosis. In this work we will focus on finding minimal diagnoses. Let us formalize the neuro-muscular diagnosis in physiotherapy in terms of a MBD problem.

#### 4.2.1. COMPS

In our model, *COMPS* is a set of all nerve roots, nerves, muscles and dermatomes. Each  $c \in COMPS$  has a health state described by  $h(c) \in \{True, False\}$ . However, since the physiotherapy clinical evaluation only discusses the neuro-muscular systems rather than other pathologies such as skin burns, the dermatomes are assumed to be healthy components that are only used for testing other components. This means that for each dermatome  $d \in COMPS$ , it holds that  $h(d) = True$ .

#### 4.2.2. OBS

The observations, *OBS* in our model, are the patient's weakened motions or defected sensations. Typically, a patient is not connected to sensors that measure the weakened motion or defected sensation. Instead, the PT stimulates the component, for instance a muscle, and observes whether it is defected. To formalize the observation, let us define a test of a component. Given a component  $c$ , we define the predicate  $testOK(c) \in \{True, False\}$ , where  $testOK(c) = True$  indicates that the test successfully passed, meaning, the motion or the sensation are not defected. Consequently,  $OBS \subseteq \{testOK(c) \mid c \in COMPS\}$ .

#### 4.2.3. SD

*SD* represents the behaviour of the components as well as the influence of each component on the others. Obviously, it is very hard to formalize the behaviour, even for experts. For example, a problem in the radial nerve might cause pain in the shoulder area, but it can also cause numbness, weakened movement or none of these symptoms. Nevertheless, it is possible to formalize that once the inputs of a component are proper and the component is healthy, then we expect to get proper outputs. Let  $in(c)$  and  $out(c)$  be the input and output of a component, respectively. We define the predicate  $ok(in(c))$ , where  $ok(in(c)) = True$  indicates that the input of component  $c$  is proper. In the same way we define the predicate  $ok(out(c))$ . If a component has more than a single input (output) we will add the index to the input (output),  $in_i(c)$  ( $out_i(c)$ ). Also, assume  $c_n$  and  $c_m$  represent the number of inputs and outputs of component  $c$ , respectively. Then the next formula states the behaviour of a component:

$$\forall c \in COMPS : \left( \bigwedge_{i \in \{1, \dots, c_n\}} ok(in_i(c)) \wedge h(c) \right) \rightarrow \bigwedge_{i \in \{1, \dots, c_m\}} ok(out_i(c))$$

In addition, we formalize how a proper output influences a test. Intuitively, proper outputs entails that a test passed successfully. Thus we add the following formula:

$$\forall c \in COMPS : \left( \bigwedge_{i \in \{1, \dots, c_m\}} ok(out_i(c)) \right) \rightarrow testOK(c)$$

Finally, to formalize the connections between the components, we use the inputs and outputs of the components. If, for instance, the first output of component  $c_i$  is the first input of  $c_j$  we add a next equality:  $out_1(c_i) = in_1(c_j)$ .

We would like to draw the attention of the reader to two conclusions arising from this model:

1. **Transitivity:** for a given component  $c$ , if (1)  $h(c) = True$  and (2) every component  $c'$  that affects  $c$  ( $out(c') = in(c)$ ) is healthy ( $h(c') = True$ ) and (3) the inputs of  $c'$  are proper ( $ok(in(c'))$ ), then it must hold that  $testOK(c) = True$ .
2. **Weak Fault Model (WFM):** in this model we describe only the healthy behaviour of a component rather than its faulty modes. Thus, we cannot conclude anything about the success of a test ( $testOK(c)$ ) in case the component is faulty ( $h(c) = False$ ). In addition, in case a test passed successfully, we cannot conclude that the component checked by this test is healthy. Only in case that a test failed, we can conclude that the tested component or one of its antecedents is faulty.

Once we formalized the problem in terms of an MBD, we can use any off-the-shelf MBD algorithms. MBD algorithms can be roughly classified into two classes of algorithms: conflict-directed and diagnosis-directed [63]. A classical conflict-directed MBD algorithm finds diagnoses in a two-stage process. First, it identifies conflict sets, each of which includes at least one fault. Then, it applies a hitting set algorithm to compute sets of multiple faults that explain the observation [2,4,64]. These methods guarantee sound diagnoses (i.e., they return only valid diagnoses), and some of them are even complete (i.e., all diagnoses are returned). However, they tend to fail for large systems due to infeasible runtime or space requirements [5].

Diagnosis-directed MBD algorithms directly search for diagnoses. This can be done by compiling the system model into some representation that allows fast inference of diagnoses, such as Binary Decision Diagrams [65] or Decomposable Negation Normal Form [66]. The limitation of this approach is that there is no guarantee that the size of the compiled representation will not be exponential in the number of system components. Another approach is SATbD, a compilation-based MBD algorithm that compiles MBD into Boolean satisfiability problem (SAT) [5,67], and then uses state-of-the-art SAT solvers to find the possible diagnoses. We follow a similar line of work here, but instead of a classical SAT solver we use a conflict-directed algorithm, which allows us to find conflicts in polynomial time in our domain by using a Logic-based Truth Maintaining System [68]. The number of conflicts and their size, in our domain, are not so big and enable a standard hitting set algorithm to compute the diagnoses in a reasonable time.

### 4.3. The Troubleshooting Process

While the diagnoses computation is feasible, the diagnosis process may still produce a large set of possible diagnoses. To assist the PT to disambiguate between the diagnoses and focus on the root cause of the pain, the third feature of *PhysIt* enables a troubleshooting process. The challenge in troubleshooting is which test(s) to choose. This process iteratively proposes tests that can discard incorrect diagnoses and focus on the root cause. We adopt the information gain approach to choose the tests to perform [6,57,60–62].

Algorithm 1 presents this process. After running the diagnosis algorithm, it creates a list of possible tests (probes) which include all the components in the diagnosis sets (line 2). It then chooses the probe that gives us the highest information gain (line 6). In practice, we broke ties randomly.

After querying about the best probe, the algorithm updates the diagnosis set: if the test successfully passed (probe's output was true), there is nothing to update (since the model is a weak fault model). Otherwise, it means that either the probed component or one of its affecting components is faulty. Hence, the algorithm removes all the diagnoses that do not contain the tested component or one of its inputs. Lastly, it updates the list of the remaining probes accordingly. This process continues until the diagnosis set  $D$  is not shrunk by the probes anymore. At the end of the process, the algorithm returns a list of the remaining diagnoses.

**Algorithm 1:** Probing Process

---

```

Input:  $\langle COMPS, OBS, SD \rangle$ 
Output:  $D$ : a set of diagnoses.
1  $D \leftarrow \text{DIAGNOSER}(\langle COMPS, OBS, SD \rangle)$ 
2  $probes_{new} \leftarrow \cup D$ 
3  $probes \leftarrow \emptyset$ 
4 while  $probs \neq probes_{new}$  do
5    $probes \leftarrow probes_{new}$ 
6    $\text{argmax}_{c \in probes} IG(c, D)$ 
7   if  $!testOK(c)$  then
8      $D \leftarrow \text{remove}(D, c)$ 
9      $probes_{new} \leftarrow \cup D$ 
10 return  $(D)$ 

```

---

The information gain calculation is a standard metric for quantifying the amount of information gained by testing a component [69]. This can be achieved by comparing between the entropy of the diagnosis set before and after the test. The entropy of the diagnosis set  $D$  is defined as

$$Ent(D) = - \sum_{\Delta \in D} P(\Delta) \cdot \log(P(\Delta)) \quad (1)$$

where  $P(\Delta)$  is the probability of the diagnosis  $\Delta$ . If the components fail independently of each other, then  $P(\Delta) = \prod_{c \in \Delta} P(c)$ , where  $P(c)$  is the probability of component  $c$  to fail. Without prior information, a common assumption is a uniform distribution of the components to fail [10,11]. The information gain from a probe is the difference between the entropy of the set  $D$  before the test of  $c$  and the entropy of the set  $D'$  remains after the test:  $IG(D|c) = Ent(D) - Ent(D')$ .

## 5. Performance Analysis

We evaluated the diagnosis correctness and the troubleshooting performance in *PhysIt* using empirical analysis of the outputted diagnoses, based on metrics from information retrieval and diagnostics. These metrics were evaluated both on simulated scenarios, and on case studies representing common scenarios we received from PTs. We first present the methodology of the scenario generation (Section 5.1) and the results on these scenarios (Section 5.2). Then we present the results on scenarios based on real-world clinical experience (Section 5.3).

### 5.1. Scenario Simulator

In order to evaluate the system, we built a simulator that checks the system's accuracy and efficiency using different metrics. The simulator has several steps in the fault injection and observation process. At first, the simulator chooses 1 to 5 faulty components, randomly. These components are used, at the end of the diagnosis process, as a ground truth to check the correctness of the diagnoses outputted by our diagnosis algorithm. We name these injected faulty components as "the real diagnosis".

Next, the simulator collects all components that can be relevant to the real diagnosis: This set includes all the components that were injected as faulty, and the set of components that can be affected by them. For example, nerve root C-6 is connected directly to Radial, Median and other nerves and connected indirectly to Brachialis, Extensor Carpi Ulnaris and other muscles. In this case, the root nerve C-6 is above all in the hierarchy, meaning that any of the components found below it can be affected by it.

Then, the simulator labels these potentially affected components with a value of  $!testOK$  with a probability of 0.5. This labeling simulates the answer of a real TP, if the component will be tested in the troubleshooting process. All other components automatically get the value  $testOK$  for their test.

The simulator makes sure that every component in the real diagnosis has at least one symptom that explains its presence and sets the value of this symptom to `!testOK`. This step is designed to make sure the completeness of the diagnosis process and that it will not miss the real diagnosis.

At last, out of the set of the symptoms labeled with `!testOK`, the simulator chooses symptoms that will form the observation set of the real diagnosis. We set the number of observations to be blocked from above by the cardinality of the number of faulty components. For example, in case of four faulty components, the range of the observation set size is between 1 to 4.

## 5.2. Results

We modeled 75 components in the system. We ran the simulator on all possible faults with a single component, and randomly created additional 150 instances per fault cardinality for cases with 2–6 components. In total, we got 825 instances. Out of these instances, 270 diagnoses contained two or more faulty components with a shared affecting component. We discarded these cases, since they cannot be considered under the assumption of minimal cardinality. Thus, the simulator finally outputted 555 different cases. We analyzed the results with several metrics:

### 5.2.1. Diagnosis Set Size

This metric measures the outputted set of diagnoses before and after the troubleshooting process. As seen in Figure 7, the number of diagnoses grows exponentially with the number of reported faulty components. Blue and green bars refer to the diagnosis set before and after the troubleshooting process, thus it can be seen that the troubleshooting process succeeds in decreasing the number of diagnoses even by a half. The more faulty components the more effective the troubleshooting algorithm is in reducing the number of diagnoses.

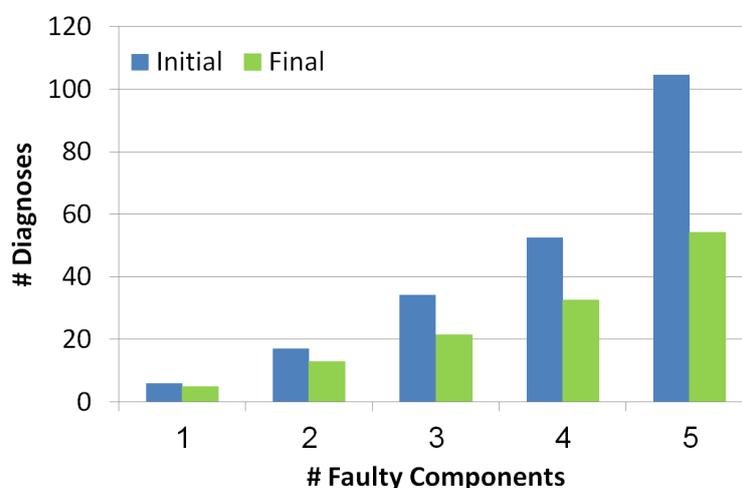


Figure 7. Number of diagnoses before and after the troubleshooting process.

### 5.2.2. False Positive Rate (FPR)

This metric measures the FPR of the outputted set of diagnoses before and after the troubleshooting process. *FPR* is measured for each diagnosis separately. The formula of this metric is:  $FPR = FP/N = FP/(FP + TN)$ , where *FP* is the number of components in the diagnosis that are not really faulty and *TN* is the number of components that are not in the diagnosis and are healthy. To compute the *FPR* of the whole set of diagnoses, we computed the weighted *FPR*, by multiplying the *FPR* of each diagnosis by its probability. Since the probabilities of the diagnoses are normalized the computation of the weighted *FPR* is correct.

The *x*-axis in Figure 8 refers to the number of faulty components while the *y*-axis refers to the *FPR* value. Blue and green bars refer to the diagnosis set before and after the troubleshooting process,

correspondingly. The lower *FPR* the better. There is a positive correlation between the number of faulty components and the *FPR* value, since the more faulty components the more diagnoses contain false positive components. Nevertheless, we can see two positive results: (1) the *FPR* is low even when the faulty components number increases, (2) the troubleshooting process reduces the *FPR*.

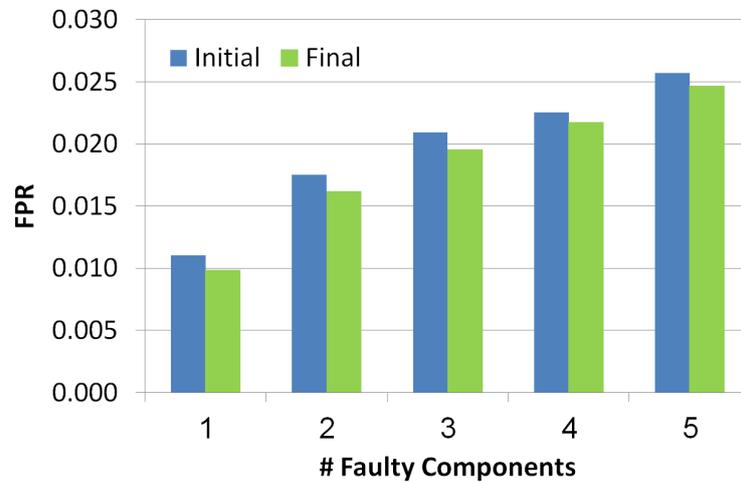


Figure 8. False positive rate of the simulated scenarios.

### 5.2.3. Area Under the Curve (AUC)

To explain this metric we should define first the term **Health State**, which has recently proposed by Stern et al. [10,11]. The health state indicates the probability of each component to be faulty, given a set of diagnoses *D* and a probability function over them *p*:

$$H(c) = \sum_{\Delta \in D} p(\Delta) \cdot \mathbb{1}_{c \in \Delta} \tag{2}$$

where  $\mathbb{1}_{c \in \Delta}$  is the indicator function defined as:

$$\mathbb{1}_{c \in \Delta} = \begin{cases} 1 & c \in \Delta \\ 0 & \text{otherwise} \end{cases}$$

Based on the health state, Stern et al. propose the AUC metric. The AUC is usually used in classification analysis to determine if the model predicts the classes well. In order to calculate the AUC value, we calculate the *FPR* and *TPR* of 11 thresholds values, 0 to 1 in hops of 0.1. Each threshold value creates a pair of values (*FPR* and *TPR*) which eventually becomes a point on the Receiver Operating Characteristic curve (ROC). The AUC is the area under the ROC curve. The higher the AUC the more accurate health state. Each threshold determines the set of components for which the *FPR* and *TPR* are calculated. All components have a higher health state than the threshold are taken into consideration.

As seen in Figure 9, the *x*-axis refers to the number of the faulty components while *y*-axis refers to AUC value. Blue and green bars refer to the diagnosis set before and after the troubleshooting process, correspondingly. There is a negative correlation between the number of faulty components and the AUC, since the number of diagnoses grows with the number of faulty components and thus the health state is less accurate. Furthermore, the AUC of the health state computed for the set of diagnoses before the troubleshooting process is lower than the AUC calculated after the troubleshooting process. This shows the benefit of the troubleshooting process.

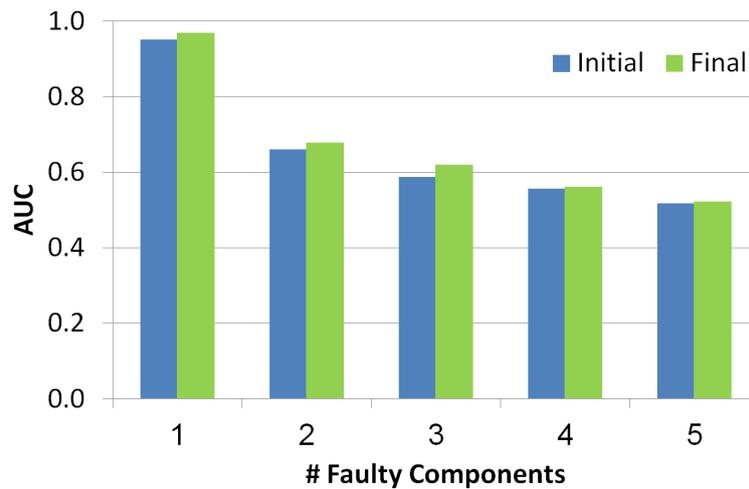


Figure 9. Area under the curve of the simulated scenarios.

#### 5.2.4. Top-K

This metric is known in the information retrieval literature. It checks whether the real diagnosis exists in the top- $K$  diagnoses returned by the algorithm, where  $K$  is a number between 1 to 5. The diagnoses are ranked in a decreasing order of their probability. As seen in Figure 10, the  $x$ -axis refers to the  $K$  value while the  $y$ -axis refers to the ratio of instances that had the faulty components in the top- $K$  diagnoses. Blue bars refer to initial diagnosis, while final diagnosis are presented by green bars. As the value of  $K$  increases, the chance to be in the top  $K$  increases too. It is clear that the final set of diagnoses shows better results than the initial set which means that the troubleshooting algorithm is indeed a helpful tool to reduce the size of the diagnosis set while improving the localization of the real diagnosis.

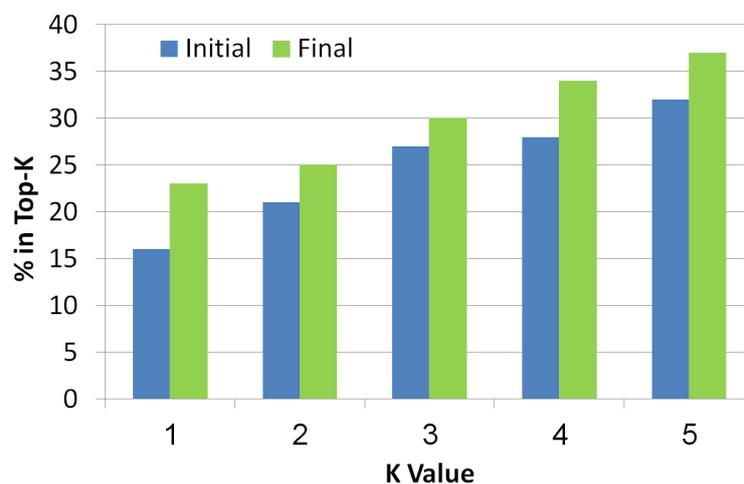


Figure 10. Top-K of the simulated scenarios.

All of the above experiments were conducted under the strict assumption that a faulty component may be assigned *!testOK* with a probability of 0.5. In practice, this probability is expected to be closer to 1 than to 0.5. Therefore, all experiments were repeated such that the simulator always assigns *!testOK* to a faulty components and the components it affects. Table 1 summarizes the results of the evaluated metrics so far, using this relaxed assumption, in order to show the real potential improvement of using this system. The rows represent the metrics and the columns represent the number of faulty components. For each metric and cardinality, we compared the initial and final values and present the improvement in the metric in percentage. This table emphasizes that the bigger the cardinality,

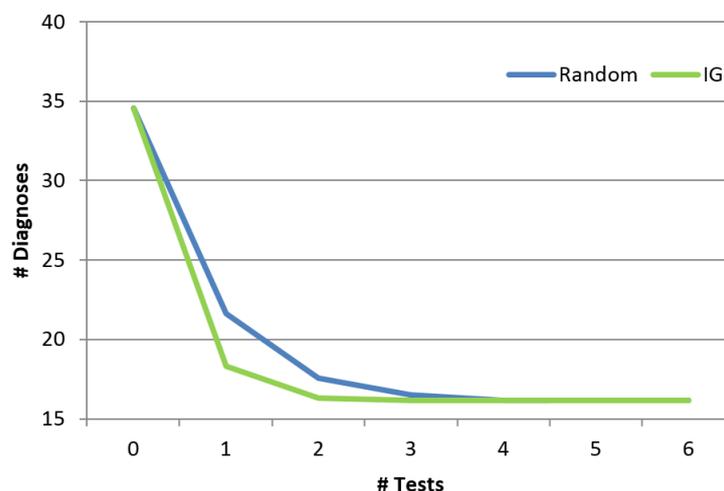
the more difficult the problem is to solve. However, the benefit of using the troubleshooting process is clear: the process manages to remove irrelevant diagnoses (according to the improvement in the wasted cost and top-5 metrics), without hindering the correctness of the results (since the FPR only improves). Moreover, the improvement of the troubleshooting becomes greater as the number of faulty components increases.

**Table 1.** Improvements in metrics per number of faulty components. \*—initial value was 0. \*\*—initial and final values were both 0.

Metric	1	2	3	4	5
FPR	0.11	0.08	0.06	0.03	0.04
AUC	0.01	0.05	0.05	0.01	0.03
Wasted Effort	0.15	0.25	0.42	0.44	0.54
Top-5	0.05	0.24	0.67	1.00 *	0.00 **

### 5.2.5. Comparing to Random

Finally, we show the benefit of the troubleshooting algorithm comparing to a random approach. The random approach chooses randomly the next component to test from a set which includes the union of all the diagnoses. Obviously, both the information gain algorithm as well as the random algorithm will finally invoke the same set of tests and the final set of diagnoses will be the same. However, the order of invoking the tests is different between the two algorithms, and might affect how fast the diagnosis set is reduced. Figure 11 shows the influence of the order of the tests ( $x$ -axis represents the number of tests) on the number of diagnoses. As shown, the troubleshooting algorithm which uses the information gain reduces the size of the diagnosis set faster than random. Even after using a single probe, the random algorithm reduces the number of diagnoses by 38%, and the information gain algorithm manages to reduce it by 47%. This is a significant difference across the examined cases ( $p < 0.01$ ). We repeated this experiment for different cardinalities (number of faulty components), and the reduction trends remain the same for all cardinalities (1 to 5).



**Figure 11.** Reduction of the diagnosis set.

### 5.3. Real-World Scenarios

With the help of experts from the Physiotherapy Department in Ben-Gurion University of the Negev, we modeled 17 representative scenarios of common cases, which are in use in physiotherapy anatomy exams. As these are written scenarios and not clinical evaluation performed on real patients, the value of some of the components is unknown, and the results of any test performed in order to reduce the possible diagnosis set will have to be simulated. Simulating test results for this

lack of values will not benefit new insights beyond the ones already received from the simulated cases. Instead, we focus this evaluation on the correctness of the outputted diagnosis set before the troubleshooting process.

In 16 out of the 17 cases investigated, the outputted diagnosis set contained the real diagnosis as reported by the PTs. In a single case, the real diagnosis was not a minimal one - but a combination of two nerve roots C-5 and C-6. According to the constructed model, all the symptoms could be explained exclusively by C-6, so the diagnosis {C-5, C-6} is redundant. Since our diagnosis algorithm searches for minimal subset diagnoses it missed this diagnosis.

Due to the completeness property of our troubleshooting process, in 16 out of the 17 cases the system managed to decrease the size of the diagnosis set without removing the correct diagnosis. These results show that even in realistic scenarios conducted by experts *PhysIt* found sound diagnoses and succeeded to reduce the diagnosis set without missing the real diagnosis.

## 6. User Study

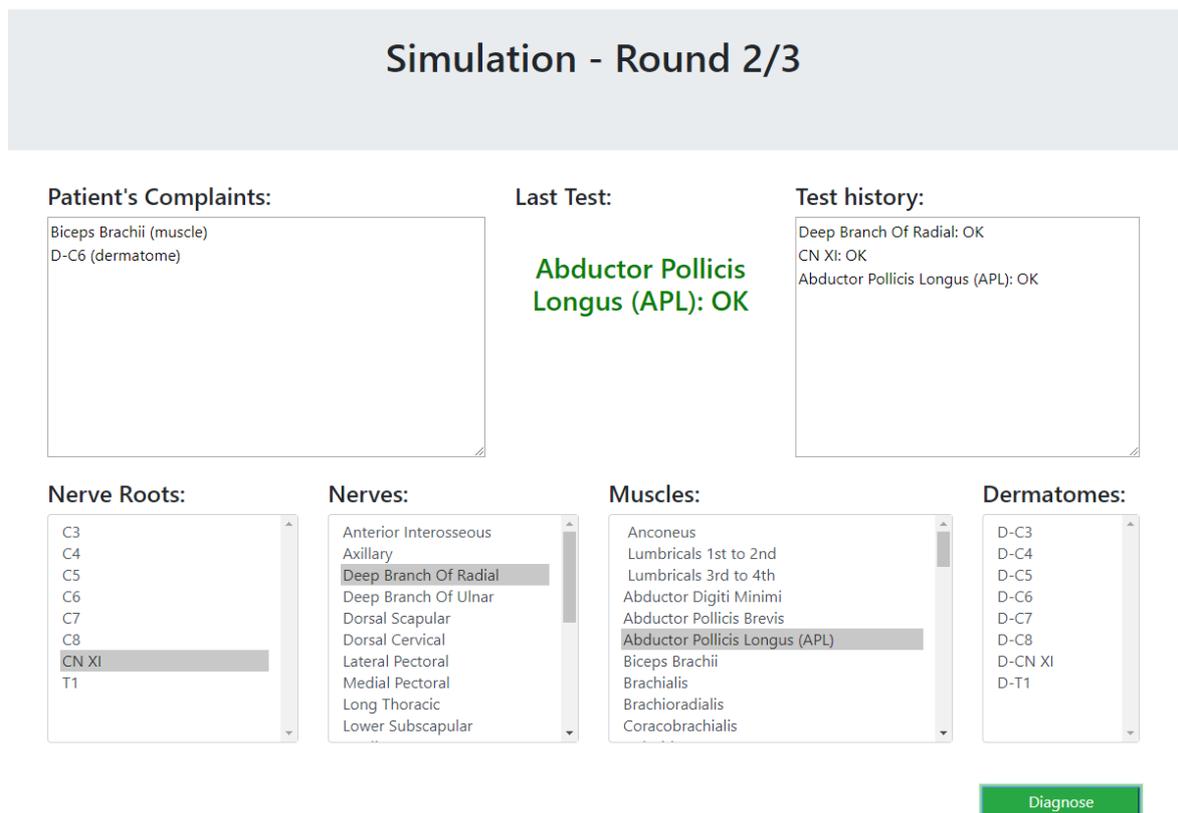
The promising results of the diagnosis system both on simulated and real scenarios, encouraged us to test the system in a human study, in order to show its ability to assist students in their physiotherapy studies. There is a variety of books and atlases that teach students anatomy [70–72]. However, to the best of our knowledge, no system is in use to assist physiotherapy students in the beginning of their clinical studies. For this reason, we devised a user study to evaluate the usefulness of *PhysIt* specifically for students in an advanced stage of their physiotherapy studies.

### *Experimental Setup*

The experiment consists of simulations of clinical diagnoses with and without the various modules of the *PhysIt* system (maps, relationships and diagnosis), following by a questionnaire to evaluate the students' experience with the system. We constructed a wrapper to our system with a landing page that can direct the user to the three different modules of *PhysIt* and to a simulator that imitates the diagnosis process.

The simulator begins with a list of symptoms that represent the patient's complaints at the beginning of a diagnosis process. Then, the participant (the experimenter) could choose a test from a list of dermatomes, muscles, nerves and nerve roots. The simulator simulates the test of the selected component by the physiotherapist and returns whether the test passed successfully (the selected component is healthy) or unsuccessfully. This process is done as long as the experimenter wishes to perform tests. The cases that were chosen for the simulator are based on the 17 expert case studies. As these cases do not elaborate the results of all possible tests, the results of unknown tests were chosen as follows. For a component that is clearly unrelated to the patient's symptoms, the relevant test returns that the component is healthy; for a component that is clearly related to the patient's symptoms, the test returns that the component is not healthy; and for a component that might be connected to one of the symptoms, the test result will be chosen at random. The simulated scenario ends when the participant decides on a diagnosis. The participants were not informed with the correctness of their responses, so it will not affect their answers about their experience with the system. A screenshot of the simulator is presented in Figure 12.

The three modules of *PhysIt* that were evaluated are: maps, relationships and diagnosis (see Section 3 for details). The participants were divided into three groups, such that each one of them had an access to a different subset of the system modules. The first group could only use the maps module; the second could use the maps and the relationships modules; and the third could use all of the three modules.



**Figure 12.** A snapshot of the user study simulator.

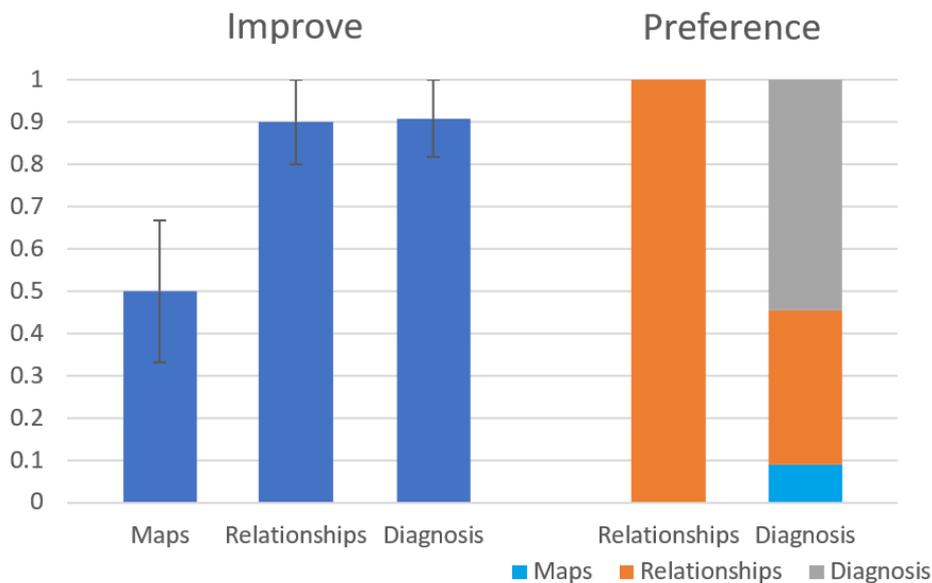
In addition to the simulations and recorded test sequences and diagnoses, the participants were requested to answer a questionnaire about their experience with the system. The questionnaire consists of the following questions:

1. Improve: Did the system improve your choice of tests to perform? (yes/no)
2. Clear: Was the system easy to understand? (5-point scale)
3. Use: Was the system easy to use? (5-point scale)
4. Preference: Which of the components did you use the most? (choice between available components)
5. Open: In your opinion, was there something that was missing in the system? (open question)

Thirty one participants in the third year of their physiotherapy studies were divided into three groups: The first group consisted of 10 student and received access to the maps module of the *PhysIt* system (the Maps group); the second consisted of 10 students and received access to both the maps and the relationships module (the Relationships group); and the third group consisted of 11 students and received access to all components of the *PhysIt* system (the Diagnosis group).

Figure 13 shows the results of the first question (Improve) and the fourth question (Preference). As seen on the left side of the figure, the Relationships and the Diagnosis modules are considered by the subjects to improve their diagnosis process significantly more than the Maps module ( $p = 0.027$  and  $p = 0.012$  respectively). The Fleiss' Kappa agreement between the subjects is 81% in the Relationships group and 66.3% in the Diagnosis group. As seen on the right side of the figure, out of the participants in the Diagnosis group, 55% preferred the diagnosis module over the other modules of the system.

Out of the students in the Relationships group, all students preferred the Relationships module over the Maps module. The results of the other general questions (Clear and Use) seem to be a slight preference to the diagnosis module over the other modules but they this preference is statistically insignificant. We have also calculated precision and recall for the diagnoses returned by the students compared to the root problem, but these results were insignificant as well.



**Figure 13.** Results for Improve and Preference questions from the user study.

For the Open question about what is missing in the system, the most common answer was that the system is missing a preliminary layer where patients can describe their symptoms (e.g., “The patient will complain on a tingling sensation, numbness, pain or weakness, not on a NOT-OK deltoid”). The patient’s complaints from this preliminary layer might later be connected to other components. Another reoccurring answer complements that the system lacks more detailed diagnoses (“e.g., the root cause of a problem is Tennis elbow rather than a NOT-OK Extensor Carpi Radialis Brevis” and “It would be nice to add to the diagnosis whether this is a chronic or acute condition”). Overall, it seems like the participants felt that the system over-simplified the diagnosis process, but was still considered useful as an educational tool.

## 7. Conclusions and Future Work

In this work, we presented *PhysIt*, a tool for diagnosis and troubleshooting for physiotherapists. We managed to apply an MBD approach in the real world, using a physiotherapy-related domain. We applied a classical MBD algorithm to compute diagnoses given some symptoms and showed that a troubleshooting process can significantly decrease the number of candidate diagnoses, without discarding the correct diagnosis. Experiments on synthetic scenarios show the benefit of the troubleshooting algorithm. Additional experiments on real scenarios show the potential benefit of *PhysIt* to reduce the set of diagnoses without hindering completeness. A user study conducted with students shows that the system could potentially be in use for physiotherapy studies in the beginning of clinical training.

From discussing this work with many PTs who are familiar with clinical evaluation and diagnosis, it seems that several desired properties are necessary in the future:

1. A malfunction in the muscle is usually reported by the patient as a mobility issue. Identifying the relevant muscle based on motion disability or pain is part of the clinical evaluation, which is not

- presented in our model. We intend to extend the system to include “movement” entities and their relations to muscles and nerves.
- In practice, most tests do not output a binary result and a component can have more states rather than *testOK* and *!testOK*. We wish to augment probabilities in our model - both to represent a degree of “faultiness” and to be able to evaluate the impact of batches of tests.
  - As shown in previous papers, abduction with a model of abnormal behaviour is a much better way to deal with medical diagnosis. To this aim we plan to achieve more information about the abnormal behaviour of components and integrate it in our model in order to discard redundant diagnoses.

**Author Contributions:** Conceptualization, R.M. and M.K.; methodology, R.M., S.H. and M.H.; software, S.H., M.H. and A.G.; validation, R.M. and M.K.; formal analysis, R.M. and M.K.; investigation, S.H., M.H. and A.G.; writing—original draft preparation, R.M. and M.K.; writing—review and editing, R.M. and M.K.; visualization, S.H., M.H. and A.G.; supervision, R.M. and M.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** This research was conducted with assistance from the physiotherapy department in Ben-Gurion University and was funded by ISF grant No. 1716/17. The user study in this work was approved for use of humans as subjects in empirical study by the ethical committee in the department of Software and System Information Engineering in Ben-Gurion University.

**Conflicts of Interest:** There are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

## References

- Jones, M.A.; Jensen, G.; Edwards, I. Clinical reasoning in physiotherapy. *Clin. Reason. Health Prof.* **2008**, *3*, 117–127.
- de Kleer, J.; Williams, B.C. Diagnosing Multiple faults. *Artif. Intell.* **1987**, *32*, 97–130. [[CrossRef](#)]
- Reiter, R. A theory of diagnosis from first principles. *Artif. Intell.* **1987**, *32*, 57–95. [[CrossRef](#)]
- Stern, R.T.; Kalech, M.; Feldman, A.; Provan, G.M. Exploring the Duality in Conflict-Directed Model-Based Diagnosis. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012.
- Metodi, A.; Stern, R.; Kalech, M.; Codish, M. A Novel SAT-Based Approach to Model Based Diagnosis. *J. Artif. Intell. Res. (JAIR)* **2014**, *51*, 377–411. [[CrossRef](#)]
- Feldman, A.; Provan, G.; van Gemund, A. A model-based active testing approach to sequential diagnosis. *J. Artif. Intell. Res. (JAIR)* **2010**, *39*, 301. [[CrossRef](#)]
- Wagner, W.P. Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies. *Expert Syst. Appl.* **2017**, *76*, 85–96. [[CrossRef](#)]
- Patel, S.; Patel, H. Survey of data mining techniques used in healthcare domain. *Int. J. Inf.* **2016**, *6*, 53–60. [[CrossRef](#)]
- Lucas, P.J.F.; Orihuela-Espina, F.; Representing Knowledge for Clinical Diagnostic Reasoning. In *Foundations of Biomedical Knowledge Representation: Methods and Applications*; Hommersom, A., Lucas, P.J., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 35–45. [[CrossRef](#)]
- Stern, R.; Kalech, M.; Rogov, S.; Feldman, A. How Many Diagnoses Do We Need? In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI’15, Austin, TX, USA, 25–30 January 2015; pp. 1618–1624.
- Stern, R.; Kalech, M.; Rogov, S.; Feldman, A. How Many Diagnoses Do We Need? *Artif. Intell.* **2017**, *248*, 26–45. [[CrossRef](#)]
- Jolliffe, I. *Principal Component Analysis*; Wiley Online Library: Hoboken, NJ, USA, 2005. [[CrossRef](#)]
- Isermann, R. Model-based fault detection and diagnosis: Status and applications. In Proceedings of the 16th IFAC Symposium on Automatic Control in Aerospace, St. Petersburg, Russia, 14–18 June 2004; pp. 71–85.
- Economou, G.P.; Lymberopoulos, D.; Karavatselou, E.; Chassomeris, C. A new concept toward computer-aided medical diagnosis—a prototype implementation addressing pulmonary diseases. *IEEE Trans. Inf. Technol. Biomed.* **2001**, *5*, 55–65. [[CrossRef](#)]

15. Sánchez-Garzón, I.; González-Ferrer, A.; Fernández-Olivares, J. A knowledge-based architecture for the management of patient-focused care pathways. *Appl. Intell.* **2014**, *40*, 497–524. [[CrossRef](#)]
16. Akerkar, R.A.; Sajja, P.S. *Knowledge-Based Systems*; Jones and Bartlett Publishers: Burlington, MA, USA, 2010.
17. Waghlikar, K.B.; Sundararajan, V.; Deshpande, A.W. Modeling paradigms for medical diagnostic decision support: A survey and future directions. *J. Med. Syst.* **2012**, *36*, 3029–3049. [[CrossRef](#)]
18. Arya, C.; Tiwari, R. Expert system for breast cancer diagnosis: A survey. In Proceedings of the 2016 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 7–9 January 2016; pp. 1–9. [[CrossRef](#)]
19. Ambilwade, R.; Manza, R.; Gaikwad, B.P. Medical expert systems for diabetes diagnosis: A survey. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2014**, *4*, 647–652.
20. Tomar, D.; Agarwal, S. A survey on Data Mining approaches for Healthcare. *Int. J. Bio-Sci. Bio-Technol.* **2013**, *5*, 241–266. [[CrossRef](#)]
21. Kourou, K.; Exarchos, T.P.; Exarchos, K.P.; Karamouzis, M.V.; Fotiadis, D.I. Machine learning applications in cancer prognosis and prediction. *Comput. Struct. Biotechnol. J.* **2015**, *13*, 8–17. [[CrossRef](#)] [[PubMed](#)]
22. Bonato, P.; Sherrill, D.M.; Standaert, D.G.; Salles, S.S.; Akay, M. Data mining techniques to detect motor fluctuations in Parkinson's disease. In Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Francisco, CA, USA, 1–5 September 2004; Volume 2, pp. 4766–4769.
23. Zorman, M.; Masuda, G.; Kokol, P.; Yamamoto, R.; Stiglic, B. Mining diabetes database with decision trees and association rules. In Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems (CBMS 2002), Maribor, Slovenia, 4–7 June 2002; pp. 134–139.
24. Lucas, P.J. Model-based diagnosis in medicine. *Artif. Intell. Med. Spec. Issues* **1997**, *10*, 201–208. [[CrossRef](#)]
25. Reggia, J.A.; Nau, D.S.; Wang, P.Y. Diagnostic expert systems based on a set covering model. *Int. J. Man-Mach. Stud.* **1983**, *19*, 437–460. [[CrossRef](#)]
26. Cruz, J.; Barahona, P. A causal-functional model applied to EMG diagnosis. In *Conference on Artificial Intelligence in Medicine in Europe*; Springer: Berlin/Heidelberg, Germany, 1997; pp. 247–260.
27. Console, L.; Dupré, D.T.; Torasso, P. A Theory of Diagnosis for Incomplete Causal Models. In Proceedings of the Eleventh International Joint Conference (IJCAI-89), Detroit, MI, USA, 20–25 August 1989; pp. 1311–1317.
28. Sebastiani, P.; Abad, M.M.; Ramoni, M.F. Bayesian Networks. In *Data Mining and Knowledge Discovery Handbook*; Maimon, O., Rokach, L., Eds.; Springer: Boston, MA, USA, 2005; pp. 193–230. [[CrossRef](#)]
29. de Kleer, J. An assumption-based truth maintenance system. *Artif. Intell.* **1986**, *28*, 127–162. [[CrossRef](#)]
30. Downing, K.L. Physiological applications of consistency-based diagnosis. *Artif. Intell. Med.* **1993**, *5*, 9–30. [[CrossRef](#)]
31. Gamper, J.; Nejd, W. Abstract temporal diagnosis in medical domains. *Artif. Intell. Med.* **1997**, *10*, 209–234. [[CrossRef](#)]
32. Weiss, S.M.; Kulikowski, C.A.; Amarel, S.; Safir, A. A model-based method for computer-aided medical decision-making. *Artif. Intell.* **1978**, *11*, 145–172. [[CrossRef](#)]
33. Citro, G.; Banks, G.; Cooper, G. INKBLOT: A neurological diagnostic decision support system integrating causal and anatomical knowledge. *Artif. Intell. Med.* **1997**, *10*, 257–267. [[CrossRef](#)]
34. Wainer, J.; de Melo Rezende, A. A temporal extension to the parsimonious covering theory. *Artif. Intell. Med.* **1997**, *10*, 235–255. [[CrossRef](#)]
35. Peng, Y.; Reggia, J.A. *Abductive Inference Models for Diagnostic Problem-Solving*; Springer Science & Business Media: New York, NY, USA, 2012.
36. Patel, V.L.; Arocha, J.F.; Zhang, J. Thinking and reasoning in medicine. *Camb. Handb. Think. Reason.* **2005**, *14*, 727–750.
37. Brusoni, V.; Console, L.; Terenziani, P.; Dupré, D.T. A spectrum of definitions for temporal model-based diagnosis. *Artif. Intell.* **1998**, *102*, 39–79. [[CrossRef](#)]
38. Pukancová, J.; Homola, M. Abductive Reasoning with Description Logics: Use Case in Medical Diagnosis. In *Description Logics*; Journal of Artificial Intelligence Research: El Segundo, CA, USA, 2015.
39. Console, L.; Torasso, P. On the co-operation between abductive and temporal reasoning in medical diagnosis. *Artif. Intell. Med.* **1991**, *3*, 291–311. [[CrossRef](#)]
40. Huang, Y.; McMurrin, R.; Dhadyalla, G.; Jones, R.P. Probability based vehicle fault diagnosis: Bayesian network method. *J. Intell. Manuf.* **2008**, *19*, 301–311. [[CrossRef](#)]

41. Mengshoel, O.J.; Chavira, M.; Cascio, K.; Poll, S.; Darwiche, A.; Uckun, S. Probabilistic model-based diagnosis: An electrical power system case study. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2010**, *40*, 874–885. [[CrossRef](#)]
42. Steinder, M.; Sethi, A.S. Probabilistic fault localization in communication systems using belief networks. *IEEE/ACM Trans. Netw.* **2004**, *12*, 809–822. [[CrossRef](#)]
43. Hood, C.S.; Ji, C. Probabilistic network fault detection. In Proceedings of the GLOBECOM'96, 1996 IEEE Global Telecommunications Conference, Communications: The Key to Global Prosperity, London, UK, 18–28 November 1996; Volume 3, pp. 1872–1876.
44. Schvaneveldt, R.W. *Pathfinder Associative Networks: Studies in Knowledge Organization*; Ablex Publishing: New York, NY, USA, 1990.
45. Nathwani, B.N.; Clarke, K.; Lincoln, T.; Berard, C.; Taylor, C.; Ng, K.; Patil, R.; Pike, M.C.; Azen, S.P. Evaluation of an expert system on lymph node pathology. *Hum. Pathol.* **1997**, *28*, 1097–1110. [[CrossRef](#)]
46. Velikova, M.; Dutra, I.; Burnside, E.S. Automated Diagnosis of Breast Cancer on Medical Images. In *Foundations of Biomedical Knowledge Representation: Methods and Applications*; Hommersom, A., Lucas, P.J., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 47–67. [[CrossRef](#)]
47. Andreassen, S.; Woldbye, M.; Falck, B.; Andersen, S.K. MUNIN: A causal probabilistic network for interpretation of electromyographic findings. In Proceedings of the 10th International Joint Conference on Artificial Intelligence, Milan, Italy, 23–29 August 1987; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1987; Volume 1, pp. 366–372.
48. Suojanen, M.; Andreassen, S.; Olesen, K.G. A method for diagnosing multiple diseases in MUNIN. *IEEE Trans. Biomed. Eng.* **2001**, *48*, 522–532. [[CrossRef](#)]
49. Mcilraith, S.A. Towards a Theory of Diagnosis, Testing and Repair. In *Proceedings of the Fifth International Workshop on Principles of Diagnosis*; Morgan Kaufmann: New Paltz, NY, USA, 1994; pp. 185–192.
50. Sampath, M.; Lafortune, S.; Teneketzis, D. Active diagnosis of discrete-event systems. *IEEE Trans. Autom. Control.* **1998**, *43*, 908–929. [[CrossRef](#)]
51. Haar, S.; Haddad, S.; Melliti, T.; Schwoon, S. Optimal Constructions for Active Diagnosis. In Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS, Guwahati, India, 12–14 December 2013; pp. 527–539.
52. Cassez, F.; Tripakis, S. Fault Diagnosis with Static and Dynamic Observers. *Fundam. Inform.* **2008**, *88*, 497–540.
53. Debouk, R.; Lafortune, S.; Teneketzis, D. On an optimization problem in sensor selection. *Discret. Event Dyn. Syst.* **2002**, *12*, 417–445. [[CrossRef](#)]
54. Mirsky, R.; Stern, R.; Gal, Y.; Kalech, M. Sequential Plan Recognition. In Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI), New York, NY, USA, 9–15 July 2016.
55. McSherry, D. Sequential Diagnosis in the Independence Bayesian Framework. In *Soft-Ware 2002: Computing in an Imperfect World: First International Conference, Soft-Ware 2002 Belfast, Northern Ireland, April 8–10, 2002 Proceedings*; Bustard, D., Liu, W., Sterritt, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 217–231. [[CrossRef](#)]
56. Heckerman, D.; Breese, J.S.; Rommelse, K. Decision-theoretic troubleshooting. *Commun. ACM* **1995**, *38*, 49–57. [[CrossRef](#)]
57. de Kleer, J.; Raiman, O. Trading off the Costs of Inference vs. Probing in Diagnosis. In Proceedings of the IJCAI 1995, Montreal, QC, Canada, 20–25 August 1995; pp. 1736–1741.
58. McSherry, D. Interactive case-based reasoning in sequential diagnosis. *Appl. Intell.* **2001**, *14*, 65–76. [[CrossRef](#)]
59. Xudong, W.; Biswas, G.; Weinberg, J. MDS: An integrated architecture for associational and model-based diagnosis. *Appl. Intell.* **2001**, *14*, 179–195.
60. Brodie, M.; Rich, I.; Ma, S. Intelligence Probing: A Cost-Effective Approach to Fault Diagnosis Computer Networks. *IBM Syst. J.* **2002**, *41*, 372–385. [[CrossRef](#)]
61. Zamir, T.; Stern, R.; Kalech, M. Using Model-Based Diagnosis to Improve Software Testing. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014.
62. Landi, C.; van Gemund, A.; Zanella, M. Heuristics to Increase Observability in Spectrum-based Fault Localization. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Prague, Czech Republic, 18–22 August 2014; pp. 1053–1054.

63. Nica, I.; Pill, I.; Quaritsch, T.; Wotawa, F. The Route to Success: A Performance Comparison of Diagnosis Algorithms. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI '13, Beijing, China, 3–9 August 2013; pp. 1039–1045.
64. Williams, B.C.; Ragno, R.J. Conflict-directed A\* and its role in model-based embedded systems. *Discret. Appl. Math.* **2007**, *155*, 1562–1595. [[CrossRef](#)]
65. Torta, G.; Torasso, P. The Role of OBDDs in Controlling the Complexity of Model Based Diagnosis. In Proceedings of the 15th International Workshop on Principles of Diagnosis (DX04), Carcassonne, France, 23–25 June 2004; pp. 9–14.
66. Darwiche, A. Decomposable Negation Normal Form. *J. ACM* **2001**, *48*, 608–647. [[CrossRef](#)]
67. Metodi, A.; Stern, R.; Kalech, M.; Codish, M. Compiling Model-Based Diagnosis to Boolean Satisfaction. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012.
68. Rutenburg, V. Propositional truth maintenance systems: Classification and complexity analysis. *Ann. Math. Artif. Intell.* **1994**, *10*, 207–231. [[CrossRef](#)]
69. Shannon, C.E. A mathematical theory of communication. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2001**, *5*, 3–55. [[CrossRef](#)]
70. Netter, F.H.; Colacino, S.; others. *Atlas of Human Anatomy*; Ciba-Geigy Corporation: Basel, Switzerland, 1989.
71. Netter's 3D Anatomy. 2018. Available online: <http://netter3danatomy.com/> (accessed on 28 January 2020).
72. Healthline Human Body Maps. 2005. Available online: <https://www.healthline.com/human-body-maps> (accessed on 28 January 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).