Zilker, Sandra; Weinzierl, Sven; Kraus, Mathias; Zschech, Patrick; Matzner, Martin

A machine learning framework for interpretable predictions in patient pathways: The case of predicting ICU admission for patients with symptoms of sepsis

Health Care Management Science

Provided in Cooperation with:
Springer Nature

Suggested Citation: Zilker, Sandra; Weinzierl, Sven; Kraus, Mathias; Zschech, Patrick; Matzner, Martin (2024) : A machine learning framework for interpretable predictions in patient pathways: The case of predicting ICU admission for patients with symptoms of sepsis, Health Care Management Science, ISSN 1572-9389, Springer US, New York, NY, Vol. 27, Iss. 2, pp. 136-167, https://doi.org/10.1007/s10729-024-09673-8

Health Care Management Science (2024) 27:136–167
https://doi.org/10.1007/s10729-024-09673-8

A machine learning framework for interpretable predictions in patient pathways: The case of predicting ICU admission for patients with symptoms of sepsis

Sandra Zilker[1,2] · Sven Weinzierl[2] · Mathias Kraus[3] · Patrick Zschech[4] · Martin Matzner[2]

Received: 8 February 2023 / Accepted: 13 April 2024 / Published online: 21 May 2024

Abstract
Proactive analysis of patient pathways helps healthcare providers anticipate treatment-related risks, identify outcomes, and allocate resources. Machine learning (ML) can leverage a patient's complete health history to make informed decisions about future events. However, previous work has mostly relied on so-called black-box models, which are unintelligible to humans, making it difficult for clinicians to apply such models. Our work introduces PatWay-Net, an ML framework designed for interpretable predictions of admission to the intensive care unit (ICU) for patients with symptoms of sepsis. We propose a novel type of recurrent neural network and combine it with multi-layer perceptrons to process the patient pathways and produce predictive yet interpretable results. We demonstrate its utility through a comprehensive dashboard that visualizes patient health trajectories, predictive outcomes, and associated risks. Our evaluation includes both predictive performance – where PatWay-Net outperforms standard models such as decision trees, random forests, and gradient-boosted decision trees – and clinical utility, validated through structured interviews with clinicians. By providing improved predictive accuracy along with interpretable and actionable insights, PatWay-Net serves as a valuable tool for healthcare decision support in the critical case of patients with symptoms of sepsis.

Keywords Patient pathway · Process prediction · Sepsis · Interpretability · Interpretable machine learning · Interpretation plots · Deep learning

Highlights

- This article proposes PatWay-Net, a novel machine learning framework for predicting critical pathways of patients with sepsis symptoms. Our framework retains patient pathway data in its natural form by combining non-linear multi-layer perceptrons (MLPs) for each static feature (i.e., static module) and an interpretable LSTM (iLSTM) cell for sequential features (i.e., sequential module).
- Our results reveal that our approach outperforms commonly used interpretable machine learning models in our case, such as decision tree and logistic regression by 10.4% and 7.3% in terms of the area under receiver operating characteristic curve, respectively, and non-interpretable models, such as random forest and XGBoost by 4.4% and 1.2%, respectively.
- PatWay-Net provides decision support to clinicians and hospital management in predicting the pathway of a patient accurately while remaining interpretable and can, therefore, help to improve hospital resource management.
- To enhance the model's interpretability and utility for clinical decision-makers, we have developed a comprehensive dashboard that visualizes patient health trajectories, predictive outcomes, and associated risks, facilitating informed clinical and resource allocation decisions.
- The clinical utility of our framework is supported by structured interviews with independent clinicians, confirming its interpretability and actionable insights for healthcare decision support.

1 Introduction

As healthcare organizations face increasing demands and limited resources, the efficiency and compliance of health-

✉ Sandra Zilker
sandra.zilker@th-nuernberg.de

Extended author information available on the last page of the article

care processes are becoming increasingly important [1]. The pandemic has served as a stress test for these processes, revealing several weaknesses, such as gaps in resource allocation, inefficiencies in patient triage, and limitations in data-driven decision-making [2, 3]. As a remedy, advanced decision support systems based on modern machine learning (ML) models can be employed to improve the performance of healthcare processes and provide proactive insights for clinical decision-makers [3–5]. By using large amounts of data that are ubiquitously generated in today's healthcare information systems, such models can learn non-trivial patterns from historical patient trajectories.

A rich source of historical patient data is represented by so-called patient pathways, a timeline of each patient that describes the different departments, measurements, treatments, and transitions that a patient has gone through during a clinical stay [6]. This information can be used to make accurate predictions about future health outcomes, informing the allocation of resources or the focus of medical professionals on specific patients [e.g., 6–9]. In this way, healthcare institutions can derive recommendations for managing and controlling patient pathways early and identify risks and issues before they emerge.

Such recommendations are especially crucial in the context of sepsis symptoms, a complex and time-sensitive condition that demands rapid identification and intervention to improve patient outcomes [10]. By leveraging patient pathway data, healthcare institutions can not only derive timely recommendations to manage and control disease progression but also identify risks and issues, such as early signs of sepsis, before they escalate [11]. Consequently, early detection and treatment of sepsis, facilitated by the analysis of patient pathways, can significantly reduce a patient's deterioration.

ML models represent a promising choice for predicting patient pathways as they can rapidly process large amounts of patient data and find latent patterns that help make informed decisions about patient outcomes. ML models come in various forms and facets. For critical applications, clinical decision-makers typically favor interpretable[1] ML models like decision trees, linear and logistic regression, and generalized additive models (GAMs) [e.g., 12–17]. They have the advantage of providing a clear understanding of how predictions are derived, which is crucial for making informed and accountable decisions. At the same time, however, such interpretable models have the limitation that they cannot handle sequential data structures in their natural form, limiting their prediction capabilities for time-varying patient data.

[1] We make a strict distinction between the terms "interpretation" and "explanation". Interpretation is derived from models designed to be intrinsically interpretable, whereas an explanation can be created by applying a post-hoc analytical explainable-artificial-intelligence approach to a black-box model (cf. Section 3).

In contrast, there is an increasing interest in using more advanced and flexible models, such as bagged and boosted decision trees [e.g., 6, 8, 9, 18] or deep neural networks (DNNs) [e.g., 19–21]. DNNs are of particular interest for predicting patient pathways because of their ability to automatically discover and learn complex patterns in high-dimensional data [22]. This ability also allows them to capture hidden patterns in sequential data structures that are difficult to identify with traditional ML models. However, DNNs generally have the limitation that they lack model interpretability because their internal decision logic is not directly comprehensible by humans [4, 23]. This renders them black boxes for model developers and decision-makers, which is why they are unsuitable for critical healthcare applications.

To address the limitations of both research streams above, we propose PatWay-Net, an innovative ML framework that is designed for both high predictive accuracy and intrinsic interpretability in modeling pathways from patients with symptoms of sepsis. With this framework, we leverage the principle of interpretable ML models while harnessing the flexibility of a DNN architecture. More specifically, our contributions are as follows:

- PatWay-Net is designed to constrain feature interactions, ensuring full model interpretability across the entire DNN architecture.
- The architecture blends non-linear multi-layer perceptrons (MLPs) for static features with an interpretable LSTM (iLSTM) cell for sequential features, preserving the natural data structure of patient pathways.
- A comprehensive dashboard supports PatWay-Net's applicability by enabling clinical decision-makers to interpret PatWay-Net's predictive outcomes and associated risks easily.
- Structured interviews with independent medical experts rigorously validate PatWay-Net's utility and interpretability, attesting to its real-world healthcare applicability.

We evaluate our proposed model using a real-life data set from an emergency department of a Dutch hospital, containing health records of patients with sepsis symptoms [11]. During their stay, patients go through different activities (e.g., changing departments, receiving medications) and develop different trajectories of severity, resulting in individual patient pathways. The data set contains a rich set of static and sequential features, such as socio-demographic data, blood measurements, medical treatments, and diagnoses, which provide a valuable basis for predicting the future behavior of individual pathways. Specifically, we use this information to predict whether a patient will be admitted to the intensive care unit (ICU), which constitutes a highly

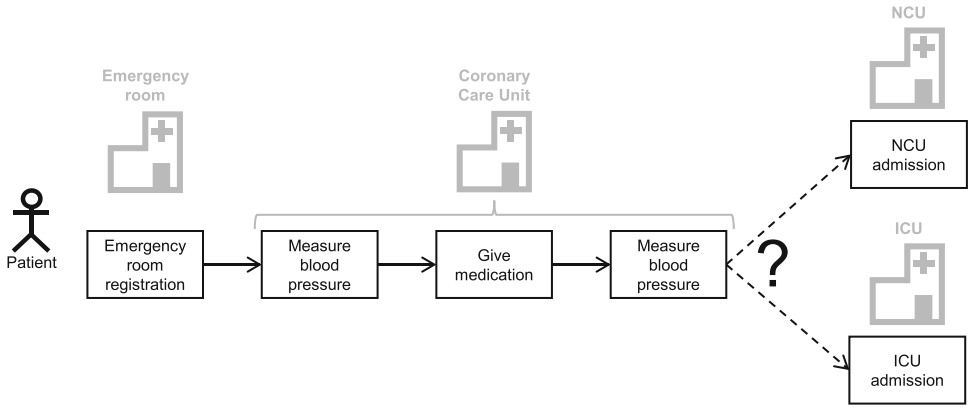


Fig. 1 Illustration of underlying setting. Multiple tasks must be performed when a patient is transferred to a new department or receives a new treatment. Thus, early prediction of the various steps a patient goes through during their hospital stay leads to more efficient operations

relevant prediction task for clinical professionals and administrative staff to support proactive resource allocation [9, 12, 24]. By comparing different types of ML models, we show that PatWay-Net outperforms commonly used interpretable models, such as decision trees or logistic regression, and even non-interpretable models, such as random forest and XGBoost, in terms of area under the receiver operating characteristic curve (AUC_{ROC}) and F1-score. We then use PatWay-Net for interpreting both static and sequential features of the real-world setting to demonstrate its applicability for healthcare decision support.

Our paper is organized as follows: Section 2 motivates the task of predicting critical patient pathways from a clinical point of view. Section 3 presents relevant background and related work. Section 4 introduces our proposed ML framework for interpretable patient pathway prediction, PatWay-Net. Section 5 outlines the evaluation and application results based on the real-life use case for predicting ICU admission for patients with symptoms of sepsis. Section 6 summarizes our work by drawing implications for research and practice, reflecting on limitations, and providing an outlook for future work.

2 Clinical relevance

2.1 Patient pathways and clinical decision support

Healthcare processes are generally concerned with all activities related to diagnosing, treating, and preventing diseases to improve well-being [25]. This includes patient-related activities organized in patient pathways and administrative activities that support clinical tasks [26]. Patient pathways are directly linked to a patient's diagnostic–therapeutic cycle and, therefore, do not constitute strictly standardized processes. However, accurate prediction of patient pathways is crucial for optimizing resource allocation, improving patient outcomes, and facilitating timely clinical interventions, thus making it an essential tool for enhancing healthcare efficiency and effectiveness.

Figure 1 illustrates a patient's hospital stay at multiple departments. In each department, various tasks must be performed to ensure a safe and well-organized patient transition. In this example, the patient was transferred from the emergency room to the coronary care unit. Depending on the patient's condition, the patient may be transferred to the ICU or the normal care unit (NCU). Therefore, both departments must be prepared for patients. By using a decision support system that accurately predicts the next station, resources for one of the departments can be saved.

Technically, a patient visiting the hospital produces a patient pathway. A set of multiple patient pathways is then stored as an event log. Table 1 presents an example representation of an event log. Here, one visit of a patient is represented by a patient pathway (ID = 1). In the beginning, the patient registered at the emergency room at *2024-02-20 12:11:01*. Also, the gender of the patient is registered as male (M). In the next patient activity, the blood pressure is measured at 180. Later, medication is administered before the

Table 1 Example event log with a single patient pathway following the scenario in Fig. 1

Patient pathway ID	Patient activity	Timestamp	Blood pressure	Gender
1	Emergency room registration	2024-02-20 12:11:01	-	M
1	Measure blood pressure	2024-02-20 13:11:27	180	M
1	Give medication	2024-02-20 14:30:27	-	M
1	Measure blood pressure	2024-02-20 15:45:55	195	M
1	ICU Admission	2024-02-20 16:12:02	-	M

blood pressure is measured for a second time at 195. The next activity then describes the patient being transferred to the ICU.

As shown in Table 1, the patient information in an event log is not structured to be easily processed by prediction models. Therefore, careful processing of static and sequential patient information is necessary to predict following patient activities accurately. In addition, experts generally have to ensure that the model learns meaningful patterns from the data, which constrains the model to be intrinsically interpretable. Both points are addressed in this work.

In the development of decision support for healthcare applications, the involvement of medical experts is inevitable [27]. Their insights can ensure that the proposed approach aligns with the complexities and problems of clinical practice. In this work, a comprehensive dashboard serves as a translational interface, bridging the gap between high-level computational outputs and real-world clinical decisions. It provides a demonstration of a model's potential for real-world applicability, ensuring that its capabilities are both understandable and useful to healthcare practitioners.

2.2 The case of sepsis

Sepsis results from the body's overwhelming response to an infection and can be life-threatening [28]. Therefore, sepsis is a time-sensitive issue that needs clinicians' attention as early as possible to enable the best possible outcome for each patient [10]. Based on this, it is critical to predict this outcome during an ongoing patient pathway to provide timely recommendations for controlling the disease's progression [11, 29]. However, the importance of sepsis lies not only in the urgency of its treatment but also in its complex and variable nature that can be detected in the resulting patient pathways [30]. While its symptoms and, thus, underlying medical indicators, can progress or change rapidly, treatment needs to be adapted dynamically which influences the patient pathway [29]. Ultimately, in the context of developing interpretable ML models for predicting patient pathways, the focus on patients with sepsis symptoms is crucial, given the imperative to enhance clinical decision-making, resource allocation, and ultimately, patient outcomes in this high-stakes domain.

3 Methodological background and related work

ML models are increasingly being integrated into clinical applications to assist healthcare professionals in diagnosing diseases, predicting patient outcomes, and making treatment decisions [7–9]. While the predictive power of these models is often decisive, it is also essential that they provide comprehensible outputs due to the critical nature of healthcare decisions. Comprehensible outputs promote transparency, reduce the risk of unintended biases, and ensure the reliability of the model results, ultimately contributing to safer and more effective patient care [31–33].

From a methodological point of view, there are generally two distinct streams of research dealing with comprehending ML models. Table 2 provides an overview of both streams with exemplary approaches, which can be further classified according to the type of input features they support.

3.1 Explainable machine learning

The first stream of research refers to the concept of explainable ML. It promotes the use of flexible ML models with high predictive power, which subsequently require *post-hoc explanation methods* to convert their complex mathematical functions into easier-to-understand explanations [23, 32]. Common representatives of flexible ML models for static features are bagged and boosted decision trees such as random forest [42] and XGBoost [43]. Such models excel at handling static tabular data because they can capture complex interactions between features, allowing them to achieve high predictive performance [6, 8, 9, 18]. In this work, we include both models as strong baseline approaches in our evaluation section. However, the construction of high-level interactions creates a lack of transparency because the individual feature effects are no longer understandable by humans and therefore require additional explanation methods.

For sequential features, the field has increasingly focused on DNNs in recent years [22]. Their multi-layered network architecture allows them to automatically discover and learn complex patterns in high-dimensional data structures that are relevant for the prediction task [4, 5]. Of particular interest are recurrent neural networks and long short-term memory (LSTM) networks because they can capture temporal patterns and therefore offer superior predictive performance compared to traditional approaches in dynamic and complex healthcare process environments [e.g., 19, 20]. Furthermore, such network architectures have the advantage that they can be modified to capture static and sequential features simultaneously [e.g., 21, 41]. Nevertheless, the nested, multi-layered structure of DNNs also creates a lack of transparency, because it is not directly observable what information in the input data drives the models to generate their prediction, rendering them black boxes for model users. In our work, we adopt the overall idea of an LSTM network [44] but propose a modification to ensure full model transparency.

To turn the internal decision logic of black-box models into comprehensible results, the field of explainable ML has proposed a variety of post-hoc explanation methods [6, 45]. Some of these methods are model-specific. That is, they are designed for specific types of models and derive explanations by examining internal model structures and parameters

Table 2 Positioning of our work with respect to related fields from a methodological perspective

	Explainable machine learning	Interpretable machine learning
Definition	Refers to methods that aim to simplify (approximate) the decision logic of ML models that are not directly understandable to human users (known as black-box models).	Refers to ML models that are designed to be inherently understandable to human users.
Main focus	Encourages the use of flexible ML models with high predictive power that require post-hoc explanations to convert complex mathematical functions into a more understandable form for clinical model validation.	Encourages the use of ML models that ensure a complete understanding and validation of the decision logic for fully transparent clinical decision support without the need for additional explanation methods.
Static features	Involves a scenario where a flexible black-box ML model is provided with a fixed-length feature vector (e.g., age, weight, vital signs of a patient), and the model's response is analyzed after prediction using model-specific explanation methods such as layer-wise relevance propagation [e.g., 6] or model-agnostic explanation methods such as Shapley additive explanations [e.g., 18].	Interpretable ML models limit interactions between features to reduce complexity, allowing for comprehensive validation of the model's performance. Typical interpretable ML models are linear models [e.g., 12, 14, 15], decision trees [e.g., 17], or generalized additive models [e.g., 13, 16], as well as typical risk charts, such as the well-known simplified acute physiology score (SAPS) at the intensive care unit [34].
Sequential features	A black-box sequential ML model is provided with temporal patient data (e.g., a trend of vital signs over a period), and the model's response is analyzed post-hoc [e.g., 31, 35, 36]. Typical sequential ML models with high predictive power are recurrent neural networks like long short-term memory (LSTM) networks [e.g., 19, 20].	An interpretable ML model that processes temporal patient data with full transparency to medical professionals. Interpretable models allow for complete validation of model behavior. Examples include probabilistic finite automatons [e.g., 37], hidden Markov models [e.g., 38], and certain advances in neural networks [e.g., 39, 40].
Static + sequential features	A black-box model consists of two parts: One that can process static features and one that can process sequential features [e.g., 21, 41]. The information about the patient from the two sources is then combined to compute the model output.	*Our research:* A fully interpretable ML model that can process both static and time-varying patient data.

(e.g., layer-wise relevance propagation for DNNs [6, 36]). Other methods are model-agnostic and, therefore, broadly applicable to different ML models. One of the most widely used model-agnostic methods is Shapley additive explanations (SHAP) [46]. SHAP uses a game-theoretic approach to explain the output of any ML model. It has been applied, for example, to mortality prediction in ICUs [31] and to process prediction models based on general event logs [35]. An overview of existing post-hoc explanation methods is given by Loh et al. [32]. Overall, post-hoc explanation methods have the advantage of providing a high degree of flexibility while encouraging the use of models with high predictive performance. Furthermore, they can lead to valuable insights, especially for exploratory analysis purposes [47].

However, post-hoc explanation methods must also be viewed with caution. They generally attempt to reconstruct the cause of a generated prediction by approximation. As a result, they can never fully explain the entire black-box model without losing information, which may lead to unreliable results. Similarly, explanations are provided only *after* a model's prediction, making it impossible to fully validate the functioning of the model for all inputs before model deployment. This issue becomes particularly critical when the distribution of input data changes over time, and the model may need to handle input feature ranges that were not encountered during its training phase. Overall, such deficiencies can lead to misleading conclusions and potentially harmful results [33, 48]. For this reason, we refrain from pursuing this general research stream in this paper.

3.2 Interpretable machine learning

The second stream of research refers to the field of interpretable ML, which promotes the development of *intrinsically interpretable models* [23, 49]. In this research stream, the structure of an ML model is constrained, such that the resulting model allows for a better understanding of how predictions are generated. Traditional representatives are linear models and decision trees, which are easy to comprehend and therefore often remain the preferred choice in critical healthcare applications [e.g., 12, 14, 15, 17]. At the same time, however, they are generally too restricted to capture more complex relationships.

A more advanced class of intrinsically interpretable ML models are GAMs [23, 49]. In GAMs, input features are modeled independently in a non-linear way to generate univariate shape functions that can capture arbitrary patterns but remain fully interpretable. The resulting shape functions for each feature are summed up afterward to produce the final model output. Thus, GAMs include additive model constraints yet drop the linearity constraint of a simple logistic/ linear regres-

sion model. This structure is simply interpretable as it allows users to verify the importance of each feature. That is, the fitted shape functions directly reveal how each feature affects the predicted output without the need for additional explanation.

In recent years, a wide variety of GAM variants have been proposed that can learn specific types of shape functions depending on the underlying learning procedure, for example, based on splines [50], decision trees [51, 52], or even neural networks [53–55]. However, all of these approaches have in common that they primarily focus on processing static features and, therefore, cannot handle sequential data structures in their natural form [49]. As a consequence, their application in the healthcare domain is usually limited to preprocessed features in a static and aggregated form [e.g., 13, 16]. In this work, we adopt the general idea of GAMs to capture non-linear effects of individual features and propagate this idea not only to static features but also to sequential features to obtain a powerful yet fully interpretable model.

Apart from that, there are also interpretable ML models that are specifically designed to capture sequential patterns. Traditional approaches include probabilistic finite automatons [37] or hidden Markov models [38]. Such models have the drawback that they require explicit knowledge about the form of an underlying process model [56], which is challenging to discover or reconstruct from complex event data in dynamic healthcare environments [11, 57]. Therefore, recent approaches increasingly pursue the idea of constraining the structure of DNN architectures to obtain models that can process sequential features in their natural form while remaining intrinsically interpretable. To date, however, little work exists in this area and current approaches often do not distinguish between sequential and static features [e.g., 39, 40].

In summary, only a limited amount of approaches deal with the development of intrinsically interpretable models for transparent patient pathway prediction. In particular, it lacks an innovative approach that can capture non-linear relationships in the form of flexible shape functions for static as well as sequential patient features while providing comprehensible model outputs that visualize the different feature effects for transparent decision support. Likewise, to the best of our knowledge, none of the existing approaches can automatically detect and integrate (sequential) feature interactions to control the model's flexibility for improved predictive performance. As a remedy, we propose PatWay-Net, a novel ML framework that combines all these aspects within a single approach.

4 PatWay-Net

This section describes PatWay-Net, an interpretable ML framework building on a DNN model with an architecture that transfers the ideas of GAMs into a novel, intrinsically interpretable LSTM module for sequential features, and intrinsically interpretable MLPs, for static features.[2] We apply this proposed DNN architecture of PatWay-Net to the problem of patient pathway prediction but want to emphasize that our proposed architecture is universal and can be applied to a variety of problem sets that combine sequential and static data (see also Appendix D for evaluations on other use cases).

In the following, we first describe the underlying problem of patient pathway prediction (Section 4.1), before mathematically describing the architecture (Section 4.2) and the training process (Section 4.3) of PatWay-Net's DNN model. Subsequently, we describe the different interpretation plots that can be derived from the intrinsically interpretable architectural design of PatWay-Net's DNN model (Section 4.4).

4.1 Problem statement

An ML model $f \in \mathcal{F}$ should map patient pathways to a target of interest, with $\mathcal{F}$ denoting the so-called hypothesis space. Patient pathways comprise two sets of information, one set describes static information about the patient, and one set describes dynamic or sequential information about the patient.

Definition 1 (Patient Pathways) Mathematically, the information that describes a set of patients can be expressed as a tuple

$$\left(\mathbf{X}_{static},\, \mathsf{X}_{seq}\right), \tag{1}$$

where $\mathbf{X}_{static} \in \mathbb{R}^{s \times q}$ is the static patient data, and $\mathsf{X}_{seq} \in \mathbb{R}^{s \times T \times p}$ is the sequential patient data. The dimension s denotes the number of patient pathways, q indicates the number of static variables that describe a patient (e.g., one-time diagnoses or gender), and T and p describe the number of time steps that we recorded for the sequential information and the number of features tracked in each time step, respectively. A single patient's patient pathway i is denoted by the static information $\mathbf{X}^{(i)}_{static}$ and the sequential data $\mathsf{X}^{(i)}_{seq}$.

The objective of this work is to find a prediction model $f \in \mathcal{F}$ that maps the static information $\mathbf{X}_{static}$ and sequential information X_{seq} about patients to target outcomes $\mathbf{y} = (y_1, \ldots, y_s)$, that is

$$f : \left(\mathbf{X}_{static},\, \mathsf{X}_{seq}\right) \rightarrow \mathbf{y}. \tag{2}$$

The target outcomes $\mathbf{y}$ can thereby represent various patient activities in the future, such as ICU admission.

[2] For reproducibility, all developed and used material can be found here: https://github.com/fau-is/patway-net

A timely prediction of the future occurrence of an activity is crucial, as it can prevent the worsening of the patient's condition and initiate successful treatment by medical experts. Therefore, a prediction model should not only make predictions once the full patient pathway is present but should make predictions already at earlier stages, that is, with less information included in the patient pathways. Thus, we define the patient pathway prefix in the following.

Definition 2 (Patient Pathway Prefix) Given patient pathway i with static information $\mathbf{X}_{static}^{(i)} \in \mathbb{R}^q$ and sequential information $\mathsf{X}_{seq}^{(i)} \in \mathbb{R}^{T \times p}$, the patient pathway prefix of length t^* is defined as a tuple

$$\left(\mathbf{X}_{static}^{(i)},\ \mathsf{X}_{seq}^{(i)}[:t^*]\right), \tag{3}$$

where $\mathsf{X}_{seq}^{(i)}[:t^*] \in \mathbb{R}^{t^* \times p}$ denotes the first t^* time steps of the patient's sequential information.

4.2 Architecture of the DNN model

The proposed interpretable architecture of PatWay-Net is shown in Fig. 2. It contains a static, a sequential, and a connection module. While the first two modules naturally model the event log data, the connection module maps the outputs of these modules onto predictions of patient activities (in our case ICU admission).

4.2.1 Static module

The static module resembles a GAM [50], yet combines the underlying idea with the power of DNNs [54]. By making this architectural choice, we allow our proposed model to remain fully transparent. That is, the effect of each input feature on the model output can be fully assessed after training the model. This is achieved by mapping the input features separately to output values (i.e., there are no interactions between input features). This separation naturally constrains this DNN but, on the other hand, allows the visual inspection of the effect each static feature has on the network's output. Consequently, although our proposed model is derived from the field of DNNs, we make careful choices about our architecture to allow for a fully transparent white-box model (in contrast to the black-box behavior of general DNNs).

Mathematically, for q static input features $\mathbf{X}_{static}[1], \ldots, \mathbf{X}_{static}[q]$, the static module maps the input features to outputs $o^1, \ldots, o^q$ through

$$o^l = f_{MLP}^l(\mathbf{X}_{static}[l]), \qquad \text{with } l \in \{1, \ldots, q\}, \tag{4}$$

where each f_{MLP}^l denotes a neural network and $o^l \in \mathbb{R}$ indicates a single scalar. The neural networks of the architecture are trained in individual sub-modules so that the weights of the different neural networks are trained independently from each other (cf. the boxes around the neural networks of the static module in Fig. 2). With this architecture, we can later compute the outputs o^l for various input values for each neural network f_{MLP}^l and, thereby, visually inspect the effect that the input has on the output.

4.2.2 Sequential module

The sequential module extends the previous idea of our static module to a sequential setting. For this, we propose a novel interpretable LSTM (iLSTM) layer to encode the values of each sequential feature $\mathsf{X}_{seq}[j]$ into a vector $\mathbf{h}_t^j \in \mathbb{R}^m$, with $j \in \{1, 2, \ldots, p\}$, where m denotes the hidden size for a single sequential feature in the iLSTM cell. To ensure intrinsic interpretability of the iLSTM, each sequential feature has its corridor throughout the gates and state vectors of the origi-

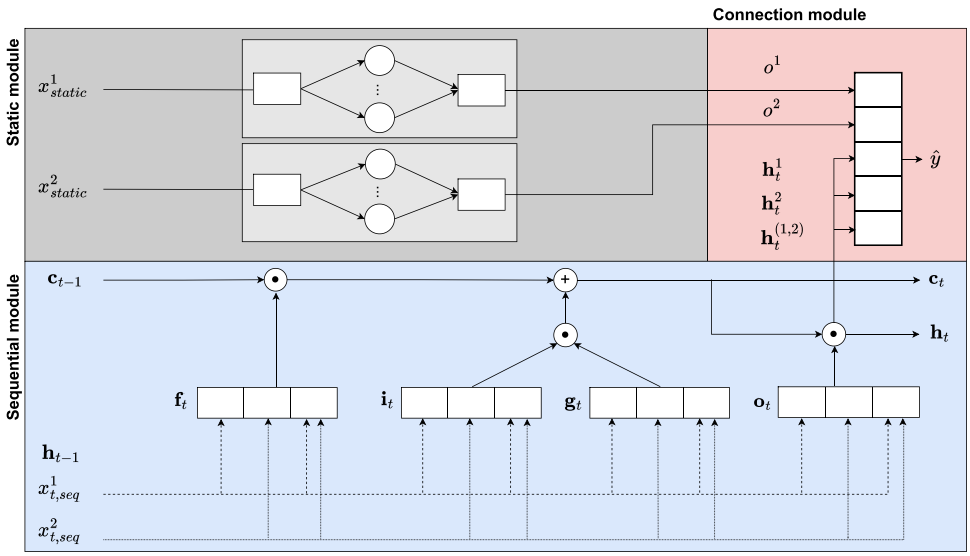


Fig. 2 Illustration of the architecture of PatWay-Net's DNN model consisting of a sequential, a static, and a connection module. Here, two static and two sequential features are shown, which run through their modules and are then connected

nal LSTM [44], without the possibility to interact with any other feature (similar to the previous static module, in which each static feature went through a separate neural network). Such feature corridors in the iLSTM layer have a specific size, defined by the internal element size of the corresponding sequential feature m, defining how much vector space is reserved for the sequential feature value computation, from the gates to the hidden state.

Similar to a vanilla LSTM [44], the iLSTM uses a forget gate, an input gate, and an output gate, as well as a candidate state, resulting in the vectors $\mathbf{f}_t, \mathbf{i}_t, \mathbf{o}_t$, and $\tilde{\mathbf{c}}_t$, respectively. The information of the sequence is then stored in a cell state $\mathbf{c}_t$, and a hidden state $\mathbf{h}_t$. Technically, this restriction, to not allow uncontrolled interactions, is realized by multiplying weight matrices with masking matrices. A masking matrix includes only 0 or 1 values. If an element of a weight matrix should be considered, the corresponding element in the masking matrix is set to 1, else it has the value 0. Mathematically, the iLSTM can be formalized as

$$\mathbf{f}_t = \sigma\left(\mathbf{x}_t \otimes (\mathbf{U}_f * \mathbf{U}_m) + \mathbf{h}_t \otimes (\mathbf{V}_f * \mathbf{V}_m) + \mathbf{b}_f\right), \quad (5)$$

$$\mathbf{i}_t = \sigma\left(\mathbf{x}_t \otimes (\mathbf{U}_i * \mathbf{U}_m) + \mathbf{h}_t \otimes (\mathbf{V}_i * \mathbf{V}_m) + \mathbf{b}_i\right), \quad (6)$$

$$\mathbf{o}_t = \sigma\left(\mathbf{x}_t \otimes (\mathbf{U}_o * \mathbf{U}_m) + \mathbf{h}_t \otimes (\mathbf{V}_o * \mathbf{V}_m) + \mathbf{b}_o\right), \quad (7)$$

$$\tilde{\mathbf{c}}_t = tanh\left(\mathbf{x}_t \otimes (\mathbf{U}_{\tilde{c}} * \mathbf{U}_m) + \mathbf{h}_t \otimes (\mathbf{V}_{\tilde{c}} * \mathbf{V}_m) + \mathbf{b}_{\tilde{c}}\right), \quad (8)$$

$$\mathbf{c}_{t+1} = \mathbf{f}_t * \mathbf{c}_t + \mathbf{i}_t * \tilde{\mathbf{c}}_t, \quad (9)$$

$$\mathbf{h}_{t+1} = \mathbf{o}_t * tanh\left(\mathbf{c}_{t+1}\right). \quad (10)$$

Here, σ denotes the sigmoid activation, $\otimes$ is the matrix multiplication, and $*$ denotes the element-wise multiplication. $\mathbf{U}_m$ and $\mathbf{V}_m$ are masking matrices that ensure that the individual features are computed independently using values from their corridor and, therefore, omitting interactions between sequential features. By contrast, a traditional, non-interpretable LSTM [44] does not use such masking matrices and, therefore, allows any interaction between features for which values are to be computed. As output, the iLSTM layer returns for each sequential feature $\mathsf{X}_{seq}[j]$ the vector $\mathbf{h}^j \in \mathbb{R}^m$, that is, the last hidden state of the iLSTM for the sequential feature j. Let $f^j_{iLSTM} \in \mathbb{R} \to \mathbb{R}^m$ denote this function, which maps the j-th sequential feature onto the corresponding hidden state, and let $f_{iLSTM} \in \mathbb{R}^p \to \mathbb{R}^{p*m}$ denote the function that maps all sequential features to the complete hidden state vector.

Beyond single sequential features, the iLSTM layer can encode values of a pairwise sequential feature interaction (j, k) in $(1, \ldots, p) \times (1, \ldots, p)$ into a vector $\mathbf{h}^{j,k} \in \mathbb{R}^m$. Mathematically, the iLSTM computes such interactions as an additional sequential feature that does not interact with other features. The interactions to be used in PatWay-Net can be chosen manually or can be detected automatically using heuristics. We describe such a heuristic in Appendix A.

4.2.3 Connection module

In the connection module, the information from the static module and the sequential module are then combined to compute the estimations $\hat{\mathbf{y}}$ for the target outcomes $\mathbf{y}$. Mathematically, we use the hidden outcome values $o^1, \ldots, o^q$ for static features, the hidden state values $\mathbf{h}^1, \ldots, \mathbf{h}^p$ for sequential features, and potentially $\mathbf{h}^{j,k}$ for interacting sequential features $j, k \in (1, \ldots, p) \times (1, \ldots, p)$. These values are then concatenated and mapped onto the output neuron to provide the estimations $\hat{\mathbf{y}}$. The mapping is performed using a single feed-forward layer with sigmoid activation, as the prediction of patient activities (in our case ICU admission) is defined as a binary classification task.

4.3 Parameter optimization of the DNN model

All parameters from the three modules are combined into one DNN model in which these are optimized simultaneously. Let $f_{\text{PatWay-Net}}$ denote this DNN model with parameters β. Depending on the task, the fit of $f_{\text{PatWay-Net}}$ to the target outcomes $\mathbf{y}$ is then measured by a loss function $\mathcal{L}$. In our real-world data application, we use binary cross-entropy, as ICU admission represents a binary decision. Overall, we minimize the empirical risk, that is

$$\beta^* = \arg\min_{\beta} \sum_{i=1}^{s} \sum_{t=1}^{T} \mathcal{L}\left(f_{\text{PatWay-Net}}\left(\mathbf{X}^{(i)}_{static}, \mathsf{X}^{(i)}_{seq}[:\, t]; \beta\right), y_i\right), \quad (11)$$

where we iterate over the patient pathways s and over the prefixes for each patient pathway T.

We address this optimization problem using an adaptive moment estimation (Adam) optimizer [58] with default hyperparameters. For every epoch, we perform a mini-batch gradient descent to optimize the internal parameters batch-wise efficiently.

4.4 Interpretations of the DNN model

Based on the architectural design of PatWay-Net's DNN model, different interpretation plots can be created, allowing an interpretation of how the model input affects the model output. The interpretation plots are part of a comprehensive dashboard, that serves as a decision support tool for clinical decision-makers (cf. Section 5.4). Table 3 provides an overview of the four interpretation plots that we propose in this paper, including plot names, the underlying equations, and short descriptions of the plots' purposes.

In the real-life data application that follows, we prefer the designation *(medical) indicator* over *feature* because it is more comprehensible for decision-makers in the medical domain. Accordingly, we name our four interpretation plots

Table 3 Overview of PatWay-Net's interpretation plots

Plot name	Underlying equation	Description
Medical indicator importance	Equations 12 and 13	This plot shows the medical indicator importance on the x-axis and the medical indicator name on the y-axis for all static and sequential data. It provides a quick overview of which medical indicators are most relevant for the model.
Medical indicator shape	Equations 12 and 13	This plot shows the medical indicator value on the x-axis and the effect on model output on the y-axis for static (Eq. 12) or sequential (Eq. 13) data. It allows medical experts to get a detailed look into the model behavior for single points in time.
Medical indicator transition	Equation 14	This plot shows the effect that a transition of a sequential medical indicator has from a value at time step $t-1$ to another value at time step t. It depicts the change of effect on the z-axis. Thereby, medical experts can observe how changes in indicator values over time (e.g., vital signs) affect the model.
Medical indicator development	Equation 15	This plot shows time steps on the x-axis and respective effect values on the y-axis for sequential data. It provides the trajectory as well as the effect that each value had on the model.

medical indicator importance, *medical indicator shape*, *medical indicator transition*, and *medical indicator development* (see Table 3). The importance, shape, and transition plots are based on so-called shape functions [52]. Traditionally, shape functions are only computed for static features [e.g., 13, 16, 51–55]. However, one of our paper's contributions is that we also extend their idea to sequential features to obtain interpretable model results for sequential features.

In general, shape functions describe the effect on the model output for various values of a single indicator. Thus, these plots answer the question, *"How does the model output change for various values of a medical indicator?"*. For a static indicator l, the shape function represents the function described by f^l_{MLP} and the corresponding parameters in the connection module, that is, it shows values $\mathbf{X}_{static}[j]$ for the l-th static indicator on the x-axis and

$$f^l_{MLP}(\mathbf{X}_{static}[l]), \tag{12}$$

weighted by the parameters in the connection module, on the y-axis. For sequential data j, the iLSTM layer can be illustrated similarly, with the x-axis showing the values $\mathsf{X}^{(i)}_{seq}[j,t]$ of a sequential indicator j for an individual pathway and a single time step t, and the y-axis, showing the corresponding effects on the model output via

$$f^j_{iLSTM}(\mathsf{X}^{(i)}_{seq}[j,:t]), \tag{13}$$

weighted by the parameters in the connection module. This plot can also be extended to interactions of two sequential indicators with a three-dimensional plot, in which the color denotes the interaction effect on the model output, as exemplarily shown in Appendix A. We call this plot the sequential medical indicator interaction plot. Note that the f_{iLSTM} layer preserves the history up to time step t. In our case, we are showing the effect of sequential data, depending on the history of a patient's pathway.

An indicator importance can be derived by computing the area under the shape functions, that is, under the plots that are described in Eqs. 12 and 13. Thereby, PatWay-Net allows computing the overall importance of static and sequential indicators, answering the question, *"Which medical indicators are the most important?"*.

A medical indicator transition illustrates the change of effect on the model output for sequential data from time step $t-1$ to t. This answers the question, *"How does the model output change from the previous to the current time step?"*. To answer this question, for a given sequential indicator j with the sequence $\mathsf{X}^{(i)}_{seq}[j]$, we calculate the difference in the effect in the sequential module between the time steps t and $t-1$, that is, we calculate

$$f^j_{iLSTM}(\mathsf{X}^{(i)}_{seq}[j,t]) - f^j_{iLSTM}(\mathsf{X}^{(i)}_{seq}[j,t-1]), \tag{14}$$

weighted by the parameters in the connection module. To illustrate all combinations of changes in the value from the last to the current time step, along with the change of effect on

Table 4 Summary statistics of the numerical medical indicators in our real-life data set

Medical indicator	Obs.	Mean	SD	Percentile				
				5%	25%	50%	75%	95%
Age	724	72.12	15.48	40.0	65.0	75.0	85.0	90.0
CRP	2,388	111.66	83.53	12.0	44.0	94.0	156.0	276.0
LacticAcid	992	1.98	1.49	0.7	1.1	1.6	2.3	4.7
Leukocytes	2,525	13.24	16.87	2.8	7.6	11.0	15.1	24.9

Note: Obs. = Number of observations, SD = Standard deviation

the model output, we use a three-dimensional plot, in which the z-axis (the color) describes an increase or decrease in the change of the effect.

Lastly, a medical indicator development describes the trajectory of an indicator over time. Due to the design of PatWay-Net's DNN model, we can illustrate the sequential effect over time. That is, the model output can be tracked for each time step of the patient pathway's sequential information and plotted afterward. This plot is specifically useful to answer the question, *"What effect did a sequential medical indicator of a given patient pathway have on the model output over time?"*. As such, we adopt the general idea of Weinzierl et al. [36] to provide transparency at the local instance level. However, instead of using a post-hoc explanation method, we can directly plot the interpretable results from our sequential and connection module. Mathematically, this can be derived for a sequence $\mathsf{X}_{seq}^{(i)}[:\ t],\ t = 1, \ldots, T$ of indicator j through a plot, showing the time steps $1, \ldots, T$ on the x-axis, and, on the y-axis,

$$f_{iLSTM}(\mathsf{X}_{seq}^{(i)}[j, :\ t]), \qquad \text{with } t = 1, \ldots, T, \tag{15}$$

weighted by a scalar value from the connection module.

5 Evaluation and application of PatWay-Net

We evaluate PatWay-Net and demonstrate its applicability using a real-world use case from a Dutch hospital. After introducing the use case (Section 5.1) and describing the baseline models (Section 5.2), we perform a three-step evaluation procedure. First, we evaluate the predictive performance of PatWay-Net's intrinsically interpretable DNN model through a benchmark study (Section 5.3). Second, we evaluate the meaningfulness of PatWay-Net's interpretation plots through a demonstration as part of a comprehensive dashboard for clinical decision-makers and a discussion including clinical evidence of the visualized interpretation aspects (Section 5.4). Finally, we validate the utility of PatWay-Net for decision-makers through structured interviews with clinicians from different hospitals and different domains. The results of those interviews are also presented in Section 5.4, while additional information can be found in Appendix C.[3]

5.1 Use case description

Our real-life publicly-available data set comprises pathways of patients with sepsis symptoms from a Dutch hospital with approximately 50,000 patients per year [11].[4] The hospital uses an enterprise resource planning (ERP) system to track all performed patient events. The process consists of logistical activities, including the patient's stations through the hospital, and medical activities, such as blood value measurements and medical treatments. Although the aforementioned process can be described in a fairly structured manner based on the information provided by the use case provider [11], this structure is only reflected to a limited extent in the underlying event log. In addition, patients can run through different activities in highly individual pathways, making it difficult to detect patterns to estimate an individual pathway's outcome manually.

Based on these patient pathways, we predict whether a patient will be admitted to the ICU. This prediction is highly relevant for both healthcare providers and insurance companies. First, capacity and staff planning in ICUs are crucial and influence the patient's probability of recovery [59]. Second, admissions to the ICU for septic patients are among the highest costs compared to other diseases [24].

The patient events of the event log can be differentiated into 16 activities with different purposes, for example, release type, type of measurement, or stating whether the patient was admitted to normal care. They all represent sequential medical indicators. In addition to the control-flow informa-

[3] We further verify and demonstrate the validity of PatWay-Net's created interpretation plots by comparing these with post-hoc-generated explanations for black-box models, and verify the end-to-end training capability of PatWay-Net's DNN model (Appendix A). We also perform a simulation study (Appendix B) and verify the prediction capability of PatWay-Net's DNN model with two additional data sets from other domains (Appendix D).

[4] https://data.4tu.nl/articles/dataset/Sepsis_Cases_-_Event_Log/12707639

Table 5 Comparison between baseline models and PatWay-Net

ML approach	F1-score (weighted)		AUC_{ROC}	
	Validation	Test	Validation	Test
OUR APPROACH				
PatWay-Net (with interaction)	0.886 (±.016)	**0.896** (±.016)	0.820 (±.028)	**0.734** (±.058)
PatWay-Net (without interaction)	0.883 (±.015)	0.893 (±.016)	**0.821** (±.027)	0.724 (±.049)
INTERPRETABLE SHALLOW MACHINE LEARNING				
Decision tree	0.879 (±.019)	0.890 (±.016)	0.753 (±.060)	0.665 (±.069)
K-nearest neighbor	**0.892** (±.019)	0.859 (±.025)	0.673 (±.047)	0.600 (±.049)
Naïve Bayes	0.363 (±.242)	0.416 (±.228)	0.723 (±.043)	0.689 (±.056)
Logistic regression	0.881 (±.016)	0.890 (±.015)	0.769 (±.044)	0.684 (±.063)
NON- INTERPRETABLE MACHINE LEARNING				
LSTM network (with static module)	0.890 (±.018)	0.898 (±.014)	0.840 (±.028)	0.757 (±.049)
XGBoost	0.883 (±.018)	0.896 (±.016)	0.817 (±.014)	0.703 (±.018)
Random forest	0.881 (±.017)	0.885 (±.011)	0.804 (±.013)	0.725 (±.016)

Note: Highlighted are the best performances among interpretable models

tion, the event log contains another 27 indicators. Three are sequential and numerical and represent the measured values of *C-reactive protein (CRP)*, *Leukocytes*, and *LacticAcid*. Furthermore, patient *Age* is a numerical and static indicator. The summary statistics of the numerical indicators are presented in Table 4.

Besides *Age*, there are 22 categorical static indicators (e.g., type of medical staff executing the activity), or binary values (e.g., stating whether or not the patient received an infusion) [11]. To avoid data leakage, we remove the medical indicator *diagnosis* as the large majority of the patient pathways with certain diagnoses describe patients who are later admitted to the ICU. That is, it can be assumed that the hospital guidelines require all patients with a certain diagnosis to be admitted to the ICU. A detailed description of the further data preprocessing steps, as well as evidence that the size of the used event log is appropriate for PatWay-Net's DNN model to achieve accurate and timely predictions, can be found in Appendix A.

5.2 Baseline models

We benchmark PatWay-Net[5] against three groups of ML approaches. The first group includes decision tree, K-nearest neighbor, naïve Bayes, and logistic regression. This group allows us to assess how well PatWay-Net performs compared to traditional shallow ML models that are intrinsically interpretable. These models are limited to processing static patient information and cannot handle sequential data. The second group includes random forest and XGBoost. This group allows us to assess how well PatWay-Net performs compared to commonly used black-box ML approaches. Like the first group, these models are limited to processing static patient information and cannot handle sequential data. Third, we include a state-of-the-art LSTM model that uses the static module of PatWay-Net and combines it with an unrestricted LSTM cell [44] to process the sequential patient information. This model represents the model with the highest flexibility and modeling capacity. Yet, it does not allow for a transparent interpretation of how the predictions are derived. Thus, its applicability in high-stakes decisions is generally limited.

These baseline models are then compared to our proposed PatWay-Net. Here, we compare two versions. First, PatWay-Net without any interactions between the sequential medical indicators. Second, PatWay-Net with pairwise interaction between a set of sequential medical indicators.[6] In doing so, we tune models by applying a grid search, evaluate models by performing a five-fold stratified cross-validation, and measure the predictive performance of models by calculating AUC_{ROC} and F1-score. More details about our model tuning, model evaluation, and model selection can be found in Appendix A.

5.3 Results on predictive performance

The predictive performance of PatWay-Net and the baselines for the use case are summarized in Table 5. Among the interpretable shallow ML models, logistic regression and naïve

[5] For simplicity, we will sometimes refer to PatWay-Net's DNN model as PatWay-Net in the following.

[6] Information on further experiments with multiple pairwise interactions can be found in Appendix A.

Bayes models outperform the decision tree and K-nearest neighbor models with an improvement of 1.9 to 8.9 percentage points in terms of AUC_{ROC} performance on the test sets.

PatWay-Net outperforms all shallow ML baseline models across all metrics. We observe that PatWay-Net, without any interactions, pushes the predictive performance by 5.1% in comparison to shallow ML models. By incorporating an interaction term in our sequential module, we achieve an AUC_{ROC} performance on the test sets of 0.734, which is an improvement of 7.3% compared to the logistic regression, and an improvement of 10.4% compared to the decision tree. PatWay-Net with the interaction in the sequential module even outperforms the non-interpretable models XGBoost and random forest by 4.4% and 1.2%, respectively. We conduct a Friedman test and a Wilcoxon signed-rank test with Holm p-value adjustment [60], which shows that the difference is statistically significant with $\alpha = 1\%$ for the decision tree, logistic regression, and K-nearest neighbor models, and with $\alpha = 10\%$ for the naïve Bayes model. Further information on the statistical tests can be found in Appendix A.

As an upper bound, the state-of-the-art LSTM network leads to an AUC_{ROC} performance on the test sets of 0.757, which is only slightly higher than our proposed PatWay-Net. However, it does not allow any intrinsic model interpretation.

5.4 Results on interpretation

PatWay-Net's interpretation plots are presented to medical decision-makers via a comprehensive dashboard (see Fig. 3), which is structured into four parts, a) to d).

Part a) provides general (static) information on a patient, such as age, height, or weight as well as their history during their current stay. For example, it shows when a patient has been admitted or when certain measurements have been taken. Part b) provides a short textual description of the

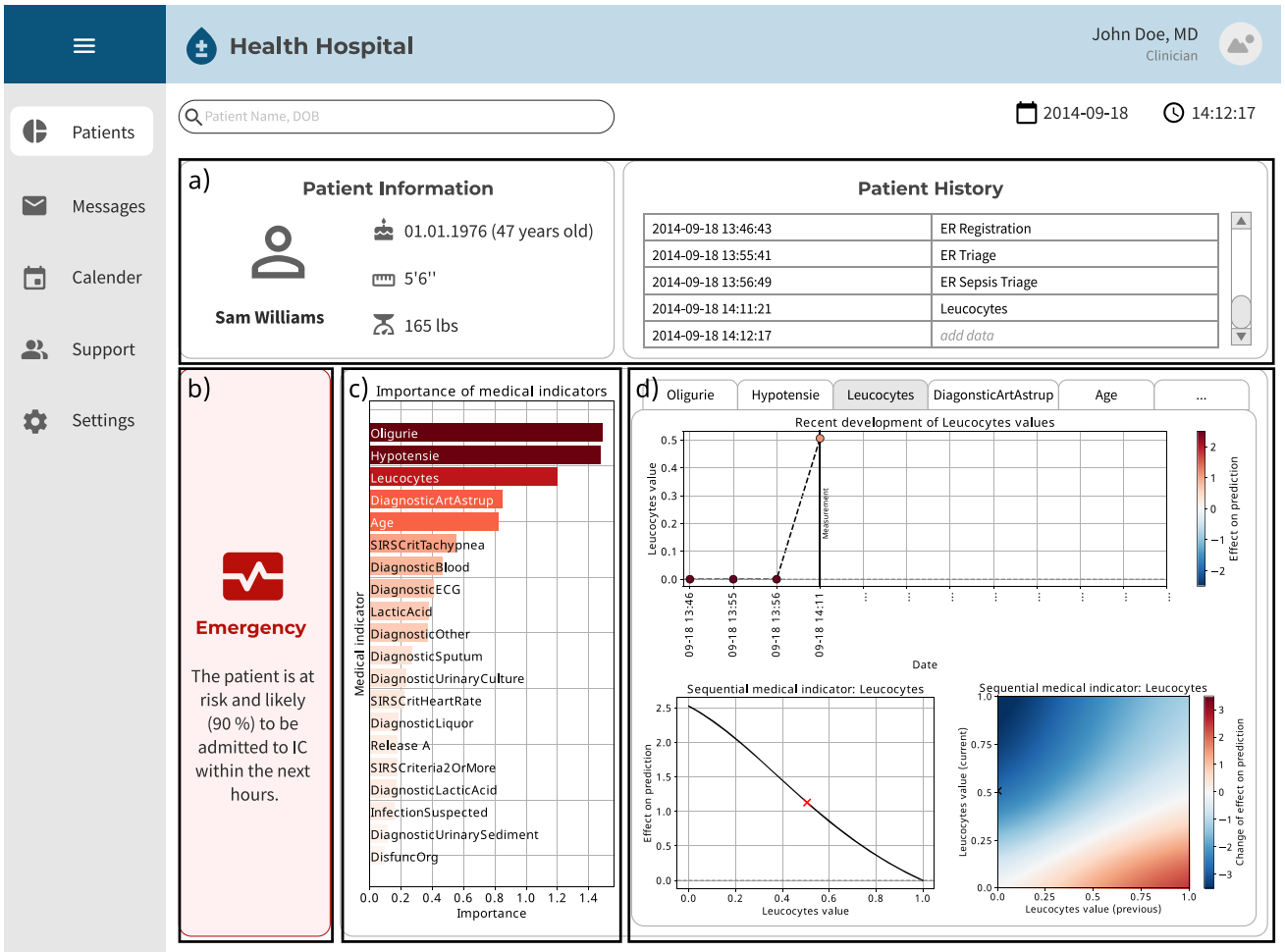


Fig. 3 Medical dashboard with PatWay-Net's interpretation plots

urgency of the ICU admission depending on the model's prediction. Parts c) and d) comprise PatWay-Net's interpretation plots. In particular, part c) shows an overview of the most impactful medical indicators on the model prediction, and d) provides further interpretation details on selected static or sequential medical indicators. In what follows, we focus on parts c) and d) of the dashboard and demonstrate PatWay-Net's interpretation plots for the medical indicator importance as well as static and sequential medical indicators.

The interviews conducted with medical experts show that the dashboard is helpful as a support for decision-making. Moreover, all medical experts confirm the usefulness of the interpretation plots to understand at a glance what caused the prediction. The interviewees also positively assessed the visual plots and thought that such plots are the language that is spoken medically. All interviewees stated that they prefer simple plots because they usually have to act relatively quickly. Another outcome of the interviews is that interpretations in the form of PatWay-Net's dashboard would positively influence their trust in the predictions. Therefore, they think that it increases the acceptance of such predictions. Additional information on the interviews can be found in Appendix C.

5.4.1 Importance of medical indicators

Figure 4 shows the medical indicator importance plot, highlighting the 20 most impactful indicators in our model. These indicators have the greatest effects on the model output in forecasting the potential need for ICU admission.

The medical indicators *Oligurie*, *Hypotensie* (hypotension), and *Leukocytes* emerge as the top contributors with the most substantial impact on the model prediction. The static indicator *Oligurie*, representing decreased urine output, is an essential medical indicator in our model for predicting ICU admission as it is often associated with severe sepsis due to its connection with reduced kidney perfusion [e.g., 61]. Likewise, *Hypotensie*, or low blood pressure, is a critical static medical indicator in our model as it can represent a possible consequence of significant blood vessel dilation caused by systemic inflammation [e.g., 62]. The sequential indicator *Leukocytes*, that is white blood cell count, also holds significant importance in our model for the prediction of ICU admission. Variations in leukocyte counts often signal the body's immune response to infections such as sepsis [e.g., 63]. As our proposed ML framework's unique capability is to include this sequential medical indicator in the analysis, it enables us to compare the importance of sequential indicators with the importance of the static indicators directly.

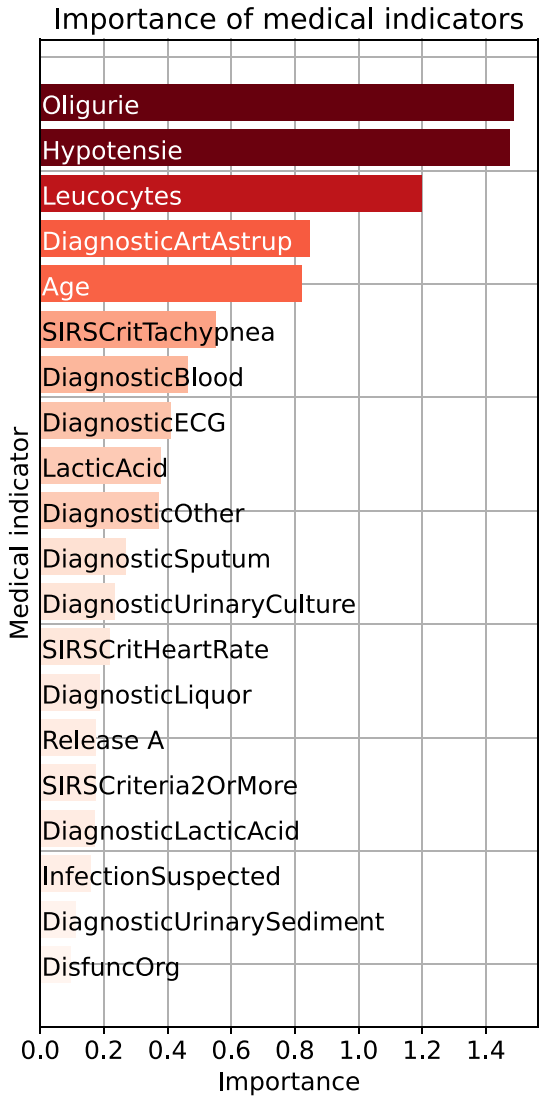


Fig. 4 Importance for static and sequential medical indicators

5.4.2 Static medical indicators

Figure 5 shows the interpretation plot for the static medical indicator *Age* in the dashboard. Specifically, it shows PatWay-Net's shape function for this specific indicator, revealing a non-linear effect of *Age* on ICU admission prediction. This demonstrates the flexibility of PatWay-Net's DNN model in capturing arbitrary relationships between individual medical indicators and the prediction target, which, inspired by traditional GAMs [e.g., 51–55], is more suitable for detecting and learning complex patterns in data than simple linear models.

Interestingly, the effect for the medical indicator *Age* decreases as the value (i.e., the patient's age) increases. At

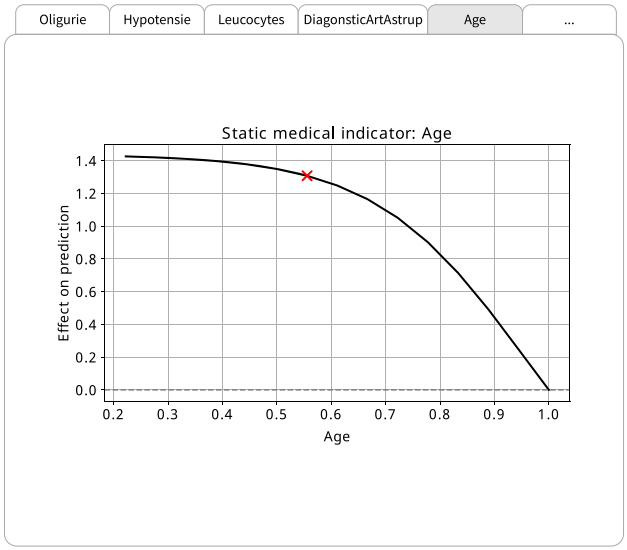


Fig. 5 Interpretation plot for static medical indicator *Age* in the dashboard

first glance, this trend may appear counter-intuitive, considering that higher age is typically associated with more severe sepsis cases and a higher risk of adverse outcomes [e.g., 64]. However, the patient population in our dataset is comparatively old, which might be a reason for this observed medical indicator shape. Additionally, there could be specific hospital protocols or clinical guidelines that apply to patients above a certain age, which could influence the patient's pathway and the eventual outcome.

5.4.3 Sequential medical indicators

Figure 6 shows the interpretation plots for the sequential medical indicator *Leukocytes* in the dashboard, depending on a given patient's previous pathway. The plots reveal the model's transparent decision logic regarding the development, shape, and transition of the medical indicator.

The medical indicator shape plot (lower left in Fig. 6) shows the shape function of the sequential medical indicator *Leukocytes*. Again, we can see the flexibility of PatWay-Net's DNN model in capturing non-linear relationships between the medical indicator and the prediction target. This time, however, the indicator represents a sequential feature captured in its natural form, which constitutes an innovative advancement over traditional interpretable models, such as GAMs and decision trees.

Leukocytes play a pivotal role in the body's immune response, and a considerable alteration in the leukocyte count is a common physiological response to sepsis [e.g., 63]. In Fig. 6, while the *Leukocytes* value decreases, the effect on the

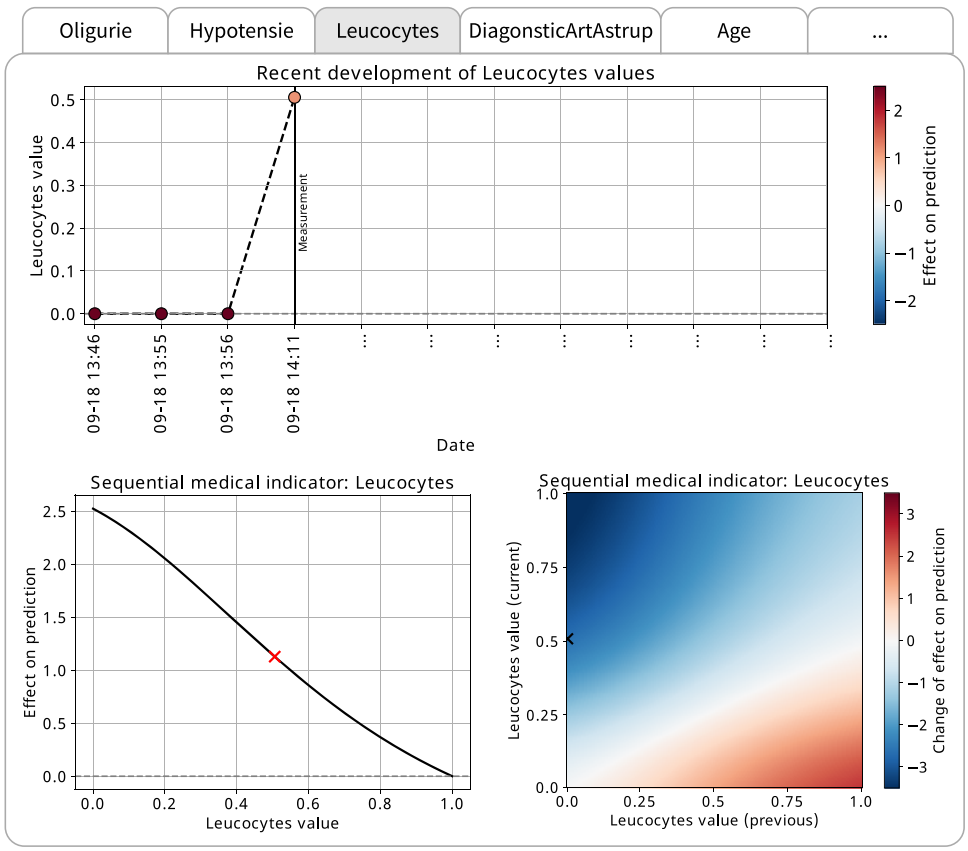


Fig. 6 Interpretation plots for sequential medical indicator *Leukocytes* in the dashboard

prediction for ICU admission increases considerably. Thus, we can see a substantial alteration in the leukocyte count. Moreover, an elevated leukocyte count can be a typical indicator of an ongoing systemic inflammatory response to an infection, like sepsis [e.g., 65]. However, a decrease in *Leukocytes* can also occur in severe cases where the immune system is overwhelmed, indicating a worsening of the patient's condition [e.g., 66]. In such an acute case, there exists a potential necessity for the patient to receive intensive care.

The medical indicator transition plot (lower right in Fig. 6) shows how the prediction changes from the previous to the current *Leukocytes* value measurement. The figure illustrates that a decrease in the *Leukocytes* value (from a previous value of 0.0 to a current value of 1.0) corresponds to an increased probability of the patient requiring ICU admission. This is consistent with clinical understanding, as a decrease in leukocytes often denotes a heightened vulnerability to developing an infection like sepsis, suggesting a more severe disease course that may require intensive care. Conversely, if there was a low *Leukocytes* value at the previous time step that subsequently increases by the current time step to a normal value, prediction indicates a lower likelihood of the patient being transferred to the ICU. This could suggest that the patient's immune response is stabilizing, or the infection is being effectively controlled, thus reducing the necessity for intensive care.

The medical indicator development plot (upper plot in Fig. 6) shows what effect the sequential medical indicator *Leukocytes* has on the model prediction over time. Up to time step three (2014-09-18 13:46 - 2014-09-18 13:56), the effect of *Leukocytes* is high since no measurement has been taken yet. From time step three to four (2014-09-18 13:56 - 2014-09-18 14:11), the effect on the prediction decreases, as a medium-high *Leukocytes* value of 0.51 has been measured in this time period.

6 Discussion and future work

6.1 Implications for healthcare management and practice

Our research has multiple implications for healthcare management and practice. First, PatWay-Net supports a straightforward analysis of patient pathways using patients' historical event data. In this way, subjectivity is avoided, and manual effort can be reduced to a minimum in decision-making. Likewise, our model provides high predictive performance in the context of patients with symptoms of sepsis without relying on explicit process knowledge. This allows flexibility for decision support applications in highly complex and dynamic healthcare environments. In our case, experiments have shown that the predictive performance is superior to traditional approaches by combining patients' static features with sequential features in a DNN architecture that remains fully interpretable. This is a great advantage because prediction tasks in the healthcare sector are usually dominated by linear and logistic regression models with underlying static features to ensure a high degree of transparency [e.g., 12–15].

At the same time, PatWay-Net can improve decision-making in both patient-specific and administrative decision contexts. For example, in a patient-specific decision context, a model interpretation for admission to ICU prediction may indicate an increase in a patient's probability of being transferred to the ICU after being treated with a certain medication. Based on this insight, medical experts have the chance to intervene and apply corrective treatments to prevent worse consequences. In an administrative context, model interpretation could reveal shortcomings in the hospital's IT system. For instance, conflicting predictions between PatWay-Net and clinicians can be traced down to potentially missing patient information within an ERP system, allowing for optimization of hospital operations.

Finally, PatWay-Net provides timely decision support. From a technical point of view, PatWay-Net's inference time is similar to one of the shallow interpretable models as the underlying model of PatWay-Net represents a function mapping the data input to the prediction output. Compared to the inference time, the training time of PatWay-Net is considerably higher than the training time of the shallow interpretable models as PatWay-Net is a DNN with a recurrent iLSTM cell. Further, the training time increases with each sequential feature as each sequential feature is passed through a single corridor in the iLSTM cell. However, for our purpose, the inference time is far more important than the training time as the models are created and trained before they are applied in an online mode where the models are used for providing effective decision support. On the other hand, PatWay-Net's interpretations can be immediately retrieved from the model itself. In doing so, it is considerably faster than applying a post-hoc explanation method such as SHAP for reconstructing explanations for non-interpretable DNN models.

6.2 Implications for research

PatWay-Net combines two crucial streams of research. The first stream follows the idea that more complex models, such as DNNs, can naturally model specific structures of the underlying data and, thereby, increase predictive performance. Thus, PatWay-Net employs an iLSTM in its sequential module to model temporal structures of sequential data, and several MLPs in its static module to model non-linear structures of static data. The second stream follows the idea that explanations of complex models can never provide the same understanding as that of intrinsically interpretable models [33, 49]. Consequently, approximated explanations

of complex models should be avoided or used carefully. As a remedy, PatWay-Net remains fully interpretable and prevents uncontrolled interactions of static and sequential features by incorporating the main principle of GAMs into its entire DNN architecture. As such, the model also provides an extension to traditional GAMs, which are unable to capture sequential data structures in their natural form [e.g., 51–55].

Within the realm of medical research, this work is aligned with emerging trends advocating for a shift from static, tabular data to multimodal data representation [67]. Traditional approaches often simplify complex health data such as images and vital signs into aggregated statistics or explicit features, thereby losing important information and only capturing a snapshot of the patient's health. Our framework addresses this gap by accurately modeling health trajectories through both, sequential and static data. The architecture is not limited to merely processing patient pathway data but it can also be adapted to other temporal sequences commonly encountered in healthcare, such as data from wearable and ambient biosensors [68]. By facilitating a more rigorous representation of human data, we improve not only the predictive performance but also the clinical utility of ML models in healthcare settings.

6.3 Limitations and outlook

As with any research, our work is not free of limitations. First, we focused in this paper on a use case of patients with symptoms of sepsis to demonstrate the benefits of PatWay-Net in cases where trust in the ML system is crucial to allow for practical applications. However, the application of PatWay-Net is not limited to this use case but can also be used to predict process-related outcomes in other tasks or domains involving static and sequential features, as shown by the results of the additional use cases in Appendix D. Here, we find mixed results, highlighting that full generalizability in other contexts requires further work.

Second, the event log sample from the real-life data application was relatively small, and using this sample for training PatWay-Net showed a performance decrease from validation to test scores. This difference could be an indicator of model selection criterion overfitting [69], which might affect PatWay-Net's generalizability to unseen data. However, to mitigate the effect of this overfitting type, we followed the suggestion from Cawley and Talbot [69] and adopted solutions for the problem of overfitting to the training criterion. In particular, we tested model regularization, hyperparameter minimization, and early stopping [69–71]. Among these solutions, performing early stopping achieved the best predictive performance for our use case of patients with symptoms of sepsis. In addition, results that we obtained from a further use case on loan applications (see Appendix D) confirm that this type of overfitting is likely to be less present when the event log size is larger. However, despite these overfitting concerns, PatWay-Net achieved relatively high predictive performance, and being a neural network, it can be expected that its predictive performance will further improve when trained with more data [22].

Third, PatWay-Net's sequential medical indicator plots provide interpretations that are tied to a patient's individual pathway. This limitation is necessary because the predictive effect of a sequential medical indicator within our iLSTM cell is determined by the patient-specific trajectory over previous time steps. As a result, varying historical trajectories can lead to different outcomes, which may also affect the results of the interpretation plots. However, at this point, it is not practical for clinical decision support to include global interpretation plots for all conceivable trajectory variants across all patients in a single dashboard. Therefore, we decided to focus on developing a patient-specific dashboard with all relevant information to support clinicians in an easily accessible way. Nonetheless, future research should address this limitation to identify new ways of how feature effects of multiple sequences over several time steps can be visualized in a comprehensive, yet fully understandable manner. This may require new visualization techniques (e.g., interactive filter mechanisms) or additional abstraction layers (e.g., clustering of patient trajectories leading to similar outcomes and interpretation plots), which offer promising directions for future work.

Fourth, PatWay-Net's mechanism to automatically detect and integrate interactions covers pairwise interactions among sequential features. For the use case addressed in this paper, we can show that the predictive performance of PatWay-Net with this mechanism is close to the predictive performance of an unrestricted LSTM cell (see Appendix A). Nevertheless, we assume that other types of interactions (e.g., more complex interactions between sequential features or interactions between sequential and static features) are more present in other use cases. The results obtained for a further use case on hospital billings (see Appendix D) give the first indication for this assumption and therefore provide an entry point for future research.

Fifth, the proposed version of PatWay-Net does not currently consider a mechanism for selecting relevant features. This may become relevant when dealing with a large collection of features in other real-world applications, where the full set of features may lead to impractically large computational costs and a higher risk of overfitting. Future research could follow up on this point to investigate which feature selection methods are appropriate for combining static and sequential features. Nevertheless, PatWay-Net already provides some guidance for selecting the most important features

(or medical indicators) through its medical indicator importance plot, thus facilitating clinicians or hospital management when dealing with a large collection of medical indicators.

Sixth, as with any ML model, PatWay-Net's results are only as good as the data it consumes. That is, PatWay-Net is not only a reflection of possibly biased decisions made in the past but also of any data quality issues embedded in the data set. For example, in our use case of patients with symptoms of sepsis, some interpretation plots showed counter-intuitive relationships between medical indicators and the prediction target that may not be reflected in the medical literature. These findings underscore the need for rigorous data management in hospital operations to enable analytics tools like PatWay-Net to enhance decision-making substantially. Similarly, we want to emphasize that the learned feature effects should not be interpreted causally, as they are still based on correlations. Thus, it is not possible to say with certainty why some of the effects shown in the interpretation plots are present. This could be due to correlations with other (unmeasured) features, or other underlying phenomena. However, despite these limitations, PatWay-Net still offers a fully transparent model that can be used to allow clinicians to compare the model results with their domain knowledge to iteratively debug and improve the model, identify underlying data quality issues, or initiate further investigations for the identification of causal relationships.

Finally, our current approach pertains to the creation of a clinical dashboard that relies solely on the provided interpretation plots generated from the shape functions of PatWay-Net. While these plots offer exact insights into the model's decision logic, they may still lack the level of context and intuitiveness required for effective clinical application. In the next steps, we intend to address this limitation by harnessing the capabilities of large language models [72, 73]. By incorporating a large language model in an adaptive dialogue system, we aim to provide more intuitive and contextually relevant explanations for clinical professionals when presenting the interpretation plots. This enhancement will not only make the model's outputs more accessible but also foster improved communication between the model and the healthcare practitioners, thereby enhancing the model's utility in real-world clinical settings.

Appendix A Further details and experiments on the main use case

In this appendix, we provide further details on the use case we address to evaluate the clinical utility of PatWay-Net, our proposed ML framework for interpretable predictions in patient pathways.

In what follows, we provide details on the preprocessing of the data set (Appendix A.1), before we present a heuristic for automatic interaction detection (Appendix A.2). After that, we provide details on model tuning, model evaluation, and model selection (Appendix A.3), and present further results on statistical tests (Appendix A.4), predictive performance (Appendix A.5, A.6, and A.7), interpretation quality (Appendix A.8), and runtime performance (Appendix A.9).

A.1 Preprocessing of the data set

We remove outliers in our data set by only considering completed patient pathways that are longer than two but shorter or equal to 50 patient events. We also remove patient pathways that do not start with activity *ER registration* because we assume this activity to be the central entry point into the patient pathway. As a result, the event log contains 724 patient pathways with 675 different variants over a period of 1.5 years.

Our real-life data set comprises binary, categorical, or continuous medical indicators. The values of a binary medical indicator are mapped to 0 or 1, categorical values are onehot-encoded, and continuous values are scaled into the range [0, 1]. Patient activities, such as *Measure Leukocyte count*, are either encoded by standard onehot encoding or by a custom encoding. In standard one-hot encoding, the patient activity a is encoded as a vector containing only zeros, except for a single position that corresponds to a, which is set to 1. In our custom encoding, we explicitly model the relationship between activities and their existing continuous medical indicators in the data. In detail, if an activity can be described by a continuous value, we set the corresponding position in the vector to this continuous value. Further, to provide more meaningful interpretations for sequential medical indicators, we also keep this continuous value for subsequent patient activities, as long the value does not change.

We extract all prefixes from $\mathbf{X}_{static} \times \mathsf{X}_{seq}$; that is, we extract all subsequences of the sequential data, denote those as X_{seq}^{sub}, and retain the static data as is, to predict at each time step. This step increases the number of training samples, which enables us to evaluate the ML models on how early they can already make accurate patient pathway predictions in the future.

For each patient pathway prefix, the target labels $\mathbf{y}$ are created as follows: Given a prefix $(\mathbf{X}_{static}^{(i)}, \mathsf{X}_{seq}^{(i)}[:t^*])$ and the patient activity of interest (e.g., Admission to ICU), we check the activities of patient pathway i. Then, if the activity of interest appears in the patient pathway's activities, we set the target label to 1 and else to 0. We discard all prefixes and corresponding labels where the activity of interest is part of the sequential data. This is important to avoid data leakage problems in patient pathway predictions.

A.2 Automatic search for interactions

Given all subsequences of the sequential data X_{seq}^{sub}, PatWay-Net's interaction detection iterates 100 times to identify the most relevant pairwise feature interactions in these data (see Algorithm 1). Per iteration, a feature pair (j, k) is randomly determined from sequential features $\mathcal{D}_{seq}$, and the sequential data for the features j and k are retrieved and reshaped. Then, the sequential data and label data are split into an 80% training set and 20% test set, an XGBoost [43] model f_{XGB} with standard parameters is trained based on this data, and the trained model is applied to the test set to calculate an AUC_{ROC} value. Subsequently, the current interaction is added together with the respective AUC_{ROC} value to $\mathbf{r}$, from which the best interactions are selected. In addition, the current interaction is added to $\mathcal{K}$ so that the same interaction cannot be used again in future iterations of this procedure. After performing all iterations, the interactions with the highest AUC_{ROC} values are first selected based on $\mathbf{r}$ and k^* (number of best interactions) and then transferred to the iLSTM layer, in which they are considered as additional sequential features.

Algorithm 1 PatWay-Net's interaction detection.

Given: X_{seq}^{sub}, $\mathbf{y}$, k^*, $\mathcal{D}_{seq}$, $f_{XGBoost}$.
for $i \leftarrow 1$ **to** *100* **do**
  $(j, k) \leftarrow$ feature pair($\mathcal{D}_{seq}$).
  if $j \neq k \wedge (j, k) \notin \mathcal{K}$ **then**
    $\mathsf{X}_{seq}^{sub} \leftarrow (\mathsf{X}_{seq}^{sub}[j], \mathsf{X}_{seq}^{sub}[k])$.
    $\mathbf{X}_{seq}^{sub} \leftarrow$ reshape(X_{seq}^{sub}).
    $\mathbf{X}_{seq}^{train}, \mathbf{y}^{train} \leftarrow$ split($\mathbf{X}_{seq}^{sub}, \mathbf{y}$).
    $\mathbf{X}_{seq}^{test}, \mathbf{y}^{test} \leftarrow$ split($\mathbf{X}_{seq}^{sub}, \mathbf{y}$).
    $f_{XGBoost}^{*} \leftarrow$ train($f_{XGBoost}, \mathbf{X}_{seq}^{train}, \mathbf{y}^{train}$).
    $perf_i \leftarrow$ test($f_{XGBoost}^{*}, \mathbf{X}_{seq}^{test}, \mathbf{y}^{test}$).
    $\mathbf{r} \leftarrow \mathbf{r} + (perf_i, (j, k))$.
    $\mathcal{K} \leftarrow \mathcal{K} \cup \{(j, k)\}$.
  end
end
return *top-k-inter*($\mathbf{r}, k^*$).

A.3 Model tuning, model evaluation, and model selection

Table 6 reports the tuning parameters used in our grid search. We performed initial experiments for all models to find

Table 6 Hyperparameters used in grid search

ML approach	Hyperparameter	Hyperparameter range
PatWay-Net	Hidden size per sequential feature	4, **8**
(with interaction)	Hidden size per static feature	4, **8**
	Learning rate	0.001, **0.01**
	Batch size	**32**, 128
PatWay-Net	Hidden size per sequential feature	**4**, 8
(without interaction)	Hidden size per static feature	**4**, 8
	Learning rate	**0.001**, 0.01
	Batch size	**32**, 128
LSTM network	Hidden size sequential features	4, 32, **128**
(with static module)	Hidden size per static feature	**4**, 8
	Learning rate	**0.001**, 0.01
	Batch size	**32**, 128
Decision tree	Max. depth	2, 3, **4**
Logistic regression	Regularization strength	$10^{-3}, 10^{-2}, \ldots, \mathbf{10^{0}} \ldots, 10^{+3}$
K-nearest neighbor	Number of neighbors	3, 5, **10**
Naïve Bayes	Variance smoothing	$10^{-9}, 10^{-8}, \ldots, \mathbf{10^{-4}}, \ldots, 1$
XGBoost	Max. depth	**2**, 6, 12
	Learning rate	0.3, 0.1, **0.03**
Random forest	Max. depth	2, 6, **12**
	Number of estimators	100, 200, **400**
	Max. leaf nodes	2, 6, **12**

Note: Best values over the five folds are marked in bold

appropriate value ranges for the hyperparameters. For the PatWay-Net models, the hyperparameters *hidden size per sequential feature* and *hidden size per static feature* define the used vector space for each sequential and static feature, respectively. For example, if the hidden size per static feature is set to 4, each model's MLP has a hidden layer with 4 neurons. In contrast to the PatWay-Net models, the baseline LSTM model uses a traditional unrestricted LSTM cell instead of the proposed, restricted iLSTM cell, and therefore, the hidden size per sequential feature defines the used vector space for all sequential features. Further, we set the maximum number of neurons for the LSTM model with the unrestricted LSTM cell to 8 and for the PatWay-Net models with the restricted iLSTM cell to 128 as experiments in the use case showed that more vector space is required for the unrestricted LSTM to identify arbitrary dependencies between all sequential features and less vector space is sufficient for the restricted LSTM to compute each sequential feature effectively and efficiently. For the shallow ML models, we set the value ranges of the hyperparameters such that the resulting models are still interpretable. For instance, we bounded the depth of the decision tree or the number of neighbors in the K-nearest neighbor model.

The procedure for model evaluation is summarized in Algorithm 2. Given the static data $\mathbf{X}_{static}$, sequential data X_{seq}, and target outcome $\mathbf{y}$, we start our evaluation by performing a five-fold stratified cross-validation with random shuffling on patient pathway level. That is, training and validation of each run are performed based on an entire pathway, without randomly shuffling sequentially ordered events, to avoid any temporal data leakage that could result from erroneously using future events as part of the evaluation of past events. After retrieving the training set and test set in each fold, we split the training set into a sub-training set (train*) and a validation set. Subsequently, we select the best model through a grid search using the sub-training and validation set and apply the best model to the test set to compute the test performance. We perform this procedure five times ($k = 5$). In total, we repeat the entire model evaluation procedure five times, each with a different seed, and calculate the average performance and standard deviation over the performance values of these executions.

To measure the performance of the ML models on the validation and the test set, we calculate the AUC_{ROC} [74] (primary measure) and the weighted F1-score (secondary measure). AUC_{ROC} determines how well a classifier can distinguish between classes [75], and remains unbiased when dealing with highly imbalanced class distributions [76]. F1-score is the harmonic mean of precision and recall [74]. Moreover, we select the model with the highest test AUC_{ROC} from the model evaluation procedure to retrieve the interpretation.

For model selection, we perform a grid search, as formalized in Algorithm 3. For each hyperparameter constellation

Algorithm 2 Model evaluation.

Given: $\mathbf{X}_{static}, \mathsf{X}_{seq}, \mathbf{y}, k, f$
$(tr_1, te_1), \ldots, (tr_k, te_k) \leftarrow$ split-k-stratified($\mathbf{X}_{static}, \mathbf{y}$).
for $i \leftarrow 1$ **to** k **do**
  $\mathbf{X}^{train}_{static}, \mathsf{X}^{train}_{seq}, \mathbf{y}^{train} \leftarrow$ retrieve($\mathbf{X}_{static}, \mathsf{X}_{seq}, \mathbf{y}, tr_i$).
  $\mathbf{X}^{test}_{static}, \mathsf{X}^{test}_{seq}, \mathbf{y}^{test} \leftarrow$ retrieve($\mathbf{X}_{static}, \mathsf{X}_{seq}, \mathbf{y}, te_i$).
  $\mathbf{X}^{train*}_{static}, \mathsf{X}^{train*}_{seq}, \mathbf{y}^{train*} \leftarrow$ split($\mathbf{X}^{train}_{static}, \mathsf{X}^{train}_{seq}, \mathbf{y}^{train}$).
  $\mathbf{X}^{val}_{static}, \mathsf{X}^{val}_{seq}, \mathbf{y}^{val} \leftarrow$ split($\mathbf{X}^{train}_{static}, \mathsf{X}^{train}_{seq}, \mathbf{y}^{train}$).
  $f_{best} \leftarrow$ grid-search($f, \mathbf{X}^{train*}_{static}, \mathsf{X}^{train*}_{seq}, \mathbf{y}^{train*}, \mathbf{X}^{val}_{static}, \mathsf{X}^{val}_{seq}, \mathbf{y}^{val}$).
  $perf_i \leftarrow$ test($f_{best}, \mathbf{X}^{test}_{static}, \mathsf{X}^{test}_{seq}, \mathbf{y}^{test}$).
end
return $\frac{1}{k}\sum_{i=1}^{k} perf_i$.

$(p_1, \ldots, p_l) \in \mathcal{P}$, the model f is trained and validated. During training, we perform early stopping based on the validation set after 10 epochs to avoid overfitting. After training, we apply the model to the validation set and calculate the AUC_{ROC}. The AUC_{ROC} is used as our selection criterion for the grid search, and the model with the highest AUC_{ROC} value on the validation set is selected for model testing.

Algorithm 3 Model selection (grid-search).

Given: $\mathbf{X}^{train}_{static}, \mathsf{X}^{train}_{seq}, \mathbf{y}^{train}, \mathbf{X}^{val}_{static}, \mathsf{X}^{val}_{seq}, \mathbf{y}^{val}, f, \mathcal{P}$.
for $(p_1, \ldots, p_l) \in \mathcal{P}_1 \times, \ldots, \times \mathcal{P}_l$ **do**
  $f_{(p_1,\ldots,p_l)} \leftarrow$ train($f, (p_1, \ldots, p_l), \mathbf{X}^{train}_{static}, \mathsf{X}^{train}_{seq}, \mathbf{y}^{train}$).
  $AUC_{ROC,(p_1,\ldots,p_l)} \leftarrow$ validate($f_{(p_1,\ldots,p_l)}, \mathbf{X}^{val}_{static}, \mathsf{X}^{val}_{seq}, \mathbf{y}^{val}$).
  if $AUC_{ROC,(p_1,\ldots,p_l)} > AUC_{ROC,best}$ **then**
    $f_{best} \leftarrow f_{(p_1,\ldots,p_l)}$.
    $AUC_{ROC,best} \leftarrow AUC_{ROC,(p_1,\ldots,p_l)}$.
  end
end
return f_{best}

A.4 Statistical tests

We perform statistical tests for the interpretable ML approaches used in our real-life data application. In particular, for our main metric (test AUC_{ROC}), we conduct a Friedman test showing significant differences among the results (statistic = 56.45143, p = 6.56037e-11). As a post-hoc test, we conduct a Wilcoxon signed-rank test with Holm p-value adjustment [60]. Table 7 shows the pairwise p-values. PatWay-Net with one interaction outperforms the decision tree ($p = .001503$), logistic regression ($p = .000593$), and K-nearest neighbor ($p = .000001$) models significantly with $\alpha = 1\%$ and the setting with one interaction significantly outperforms the naïve Bayes ($p = .044226$) models with $\alpha = 10\%$. PatWay-Net's setting without interaction also outperforms the decision tree ($p = .000637$), logistic regression ($p =$

Table 7 Overview of pairwise p-values for test AUC_{ROC} for Wilcoxon signed-rank test

p-values for Test AUC_{ROC}	PatWay-Net (with int.)	PatWay-Net (without int.)	Decision tree	K-nearest neighbor	Naïve Bayes	Logistic regression
PatWay-Net (with int.)		.379944	.001503	.000001	.044226	.000593
PatWay-Net (without int.)	.379944		.000637	.000001	.118250	.002009
Decision tree	.001503	.000637		.001461	.340556	.915693
K-nearest neighbor	.000001	.000001	.001461		.000160	.000383
Naïve Bayes	.044226	.118250	.340556	.000160		.915693
Logistic regression	.000593	.002009	.915693	.000383	.915693	

.002009), and K-nearest neighbor ($p = .000001$) models significantly with $\alpha = 1\%$.

A.5 Predictive performance for controlled vs. uncontrolled interactions

Table 8 describes the results for PatWay-Net, in which the number of interactions varies. To provide a fair comparison, the experiments for PatWay-Net with two and three interactions are performed in the same way as described in Appendix A.3. Overall, we observe robust results with only marginal differences. We find that in our real-life use case, a single sequential interaction leads to the highest AUC_{ROC} and F1-score on the test set.

Further, Fig. 7 illustrates the most relevant pairwise sequential medical indicator interaction in terms of the predictive performance of PatWay-Net (with one interaction). The figure demonstrates the interaction between the indicators *CRP* (x-axis) and *LacticAcid* (y-axis). If both indicator values are low, the interaction effect is low (black-colored region in the lower left corner). However, the interaction effect becomes stronger with increasing values of both indicators. So, if the values for both indicators are close to 1, the interaction effect is high (yellow-colored region in the upper right corner).

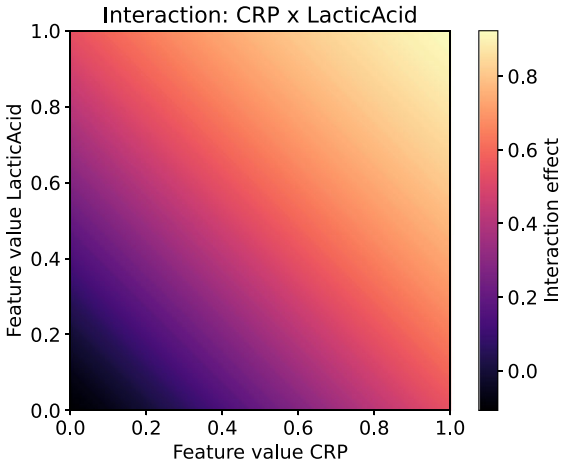


Fig. 7 Feature interaction between the indicators *CRP* and *LacticAcid*

A.6 Predictive performance for different training sample sizes

Figure 8 shows the test AUC_{ROC} scores of different training sample sizes for PatWay-Net with one interaction, the best-performing variant. More specifically, we create training samples with 10%, 20%, 40%, 60%, 80%, and 100% of the instances of the complete training set. To avoid data leakage issues, we create the training set samples based on

Table 8 Comparison of PatWay-Net with different interactions and the LSTM network (with static module)

ML approach	F1-score (weighted)		AUC_{ROC}	
	Validation	Test	Validation	Test
Controlled Interactions in Our Approach				
PatWay-Net (one interaction)	0.886 (±.016)	0.896 (±.016)	0.820 (±.028)	0.734 (±.058)
PatWay-Net (two interactions)	0.886 (±.018)	0.894 (±.013)	0.821 (±.035)	0.720 (±.045)
PatWay-Net (three interactions)	0.884 (±.017)	0.892 (±.015)	0.817 (±.035)	0.718 (±.042)
Uncontrolled Interactions in Non- Interpretable Machine Learning				
LSTM network (with static module)	0.890 (±.018)	0.898 (±.014)	0.840 (±.028)	0.757 (±.049)

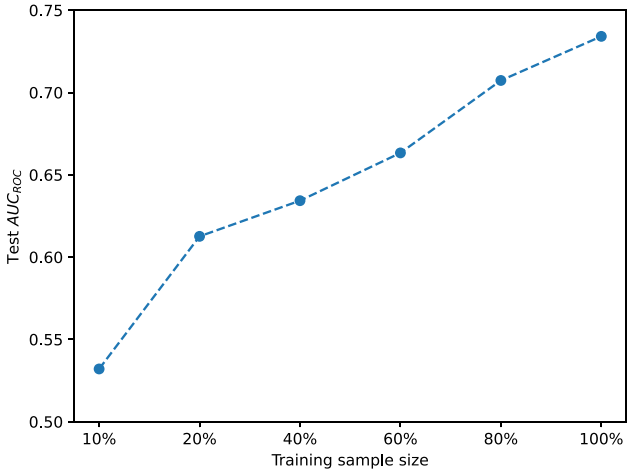


Fig. 8 Predictive performance for different training sample sizes

complete patient pathways and split each of them into a train set and validation set before the prefixes are created from the complete patient pathways. For each sample, we perform a five-fold cross-validation, as described in Appendix A.3, but with default hyperparameters.

The figure shows that the test AUC_{ROC} increases steadily with more training instances. Given that, we conclude that the event log size has an impact on the predictive performance and more training instances lead to a higher predictive performance. Further, as PatWay-Net outperforms all interpretable shallow ML baseline models in our comparison, we conclude that the use of the complete training set is appropriate for PatWay-Net to create predictions that outperform the predictions of interpretable ML baselines regarding AUC_{ROC}. Finally, as PatWay-Net belongs to the family of DNNs [22], we assume that it achieves an even higher predictive performance and maybe a greater difference in predictive performance compared to the interpretable ML baselines when more data are used for model training.

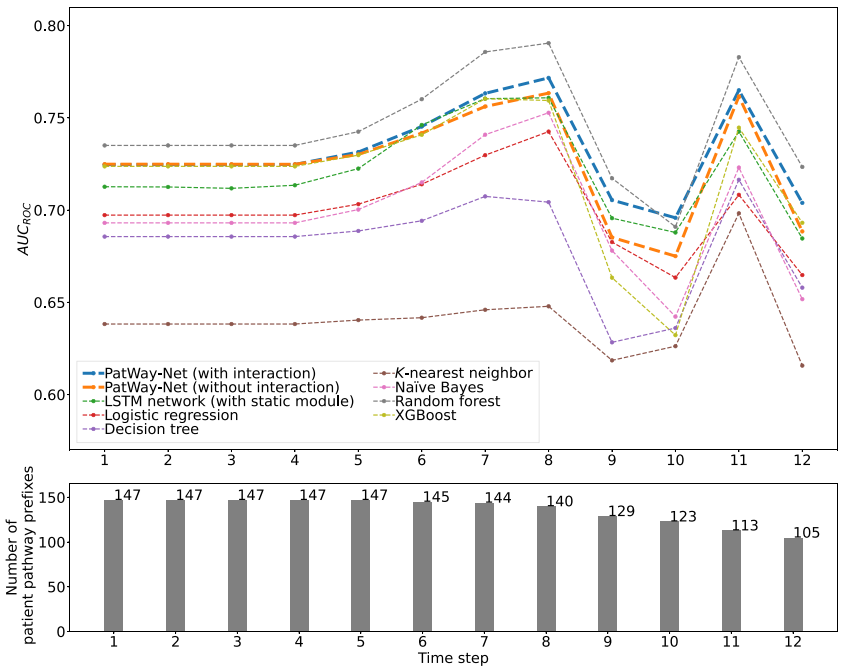
A.7 Predictive performance over time

Figure 9 shows the test AUC_{ROC} scores of the first 12 time steps of patient pathways from the use case for PatWay-Net (without and with interaction) and the baselines. At each time step, prefixes of patient pathways of the corresponding size are considered. For calculating the AUC_{ROC} scores per time step, we tune, evaluate, and select the models as described in Section A.3.

The figure shows that PatWay-Net (with and without interaction) is already able to create predictions in the first steps of patient pathways, which are considerably more accurate in terms of AUC_{ROC} than the predictions of the interpretable ML baselines. Only the not-interpretable ML approach random forest outperforms PatWay-Net (with and without interaction) in nearly all of the considered time steps. By considering only longer sequences (longer than 8), a significant number of patient paths are filtered out from this experiment, and thus the performance varies for all ML models.

Overall, this experiment empirically shows that the size of the given event log is appropriate for PatWay-Net to create timely predictions.

Fig. 9 Predictive performance over time



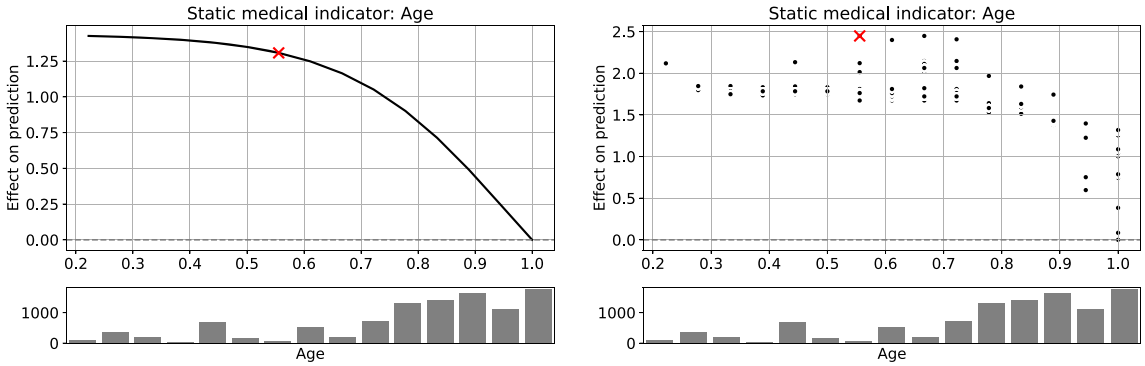


Fig. 10 Shape plot (left) and SHAP plot (right) for medical indicator *Age*

A.8 Intrinsic interpretability vs. post-hoc explainability

Figure 10 shows two interpretation plots for the same static medical indicator *Age* of our use case. While the medical indicator shape function retrieved from a PatWay-Net model with one interaction is illustrated on the left side, post-hoc generated SHAP values for a black-box XGBoost model are shown on the right side. While the trend of both plots is similar, the SHAP plot shows a high variance for the effect on the model prediction for a single value of the static medical indicator *Age*. In contrast, the shape plot for the same indicator shows a continuous function that is presumably easier to comprehend for non-technical users.

A.9 Runtime performance

Table 9 compares the training and inference time of the static model of PatWay-Net's DNN model, which is end-to-end trained, with a two-step version of the static module, which is not end-to-end trained. For the latter, we first trained an MLP model per static feature and then used the output of all MLP models as input for training and applying a subsequent logistic regression model. The experiments for this comparison were conducted on a workstation with 12 CPU cores and 128 GB RAM.

Table 9 Runtime performance

	Training time (sec)	Inference time (sec)
PatWay-Net (static module)	51.569 (±30.790)	0.005 (±.000)
MLPs + logistic regression	1646.935 (±125.027)	0.024 (±.001)

The results show that the training time of PatWay-Net's static module with end-to-end training is on average 51.569 seconds, whereas the training time for the variant without end-to-end training takes on average 1.646,935 seconds. In other words, PatWay-Net's static module is about 33 times faster than training the individual models. Thus, the training of all MLP models in one single architecture is considerably more efficient. Concerning inference, the average time for both variants is very low.

Appendix B Simulation study

In this appendix, we verify and demonstrate PatWay-Net's validity concerning the generated interpretation plots. More specifically, we follow the idea of other interpretable model proposals [e.g., 53, 54], in which the authors perform simulation studies based on synthetic data with controllable feature effects. In doing so, we prefer like in our real-life data application the designation *(medical) indicator* over *feature* because it is more comprehensible for decision-makers in the medical domain.

In the following, we provide details on the simulation data creation (Appendix B.1), the used experimental setting (Appendix B.2), the obtained results (Appendix B.3), and additional results (Appendix B.4).

B.1 Simulation data creation

For our simulation study, we create a synthetic event log containing static and sequential medical indicators.[7] The event log consists of 50,000 patient pathways with 12 events each. Every patient pathway begins with a start activity, called *ER*

[7] For the purpose of reproducibility, additional material can be found in the repository.

Registration, followed by three measurements of the *Heart Rate* and *Blood Pressure* each, and then the administration of medication (four times *A* and one time *B* in random order). Further, the data set contains four static and two sequential medical indicators. Two of the static medical indicators are numerical, namely *Age* and *BMI* (Body Mass Index), whereas the remaining two are binary, namely *Gender* and *Foreigner*. They are all set randomly. The sequential medical indicator *Heart Rate* occurs whenever the respective activity *Heart Rate* appears. Thus, it has different values that either solely increase or decrease over time. In our simulation study, the increase or decrease is exemplarily always set to 30%. The sequential medical indicator *Blood Pressure* occurs whenever the respective activity *Blood Pressure* appears. The values can increase and decrease randomly over time for one instance.

We created a continuous label for solving a regression-like prediction task. The label contains five different additive parts to introduce five different medical indicator effects concerning the static and sequential attributes in our event log: y_{gender}, y_{age}, $y_{pattern}$, y_{hr-nl}, and y_{hr} (see Eq. B1). Each part can take values between 0 and 0.2, thus, the label y can take values between 0 and 1. The remaining medical indicators (e.g., *Foreigner* and *BMI*) are only included as noise terms and do not have any effect on the target variable.

$$y = y_{gender} + y_{age} + y_{pattern} + y_{hr-nl} + y_{hr}. \tag{B1}$$

The first part y_{gender} shows the influence of the static medical indicator *gender*, where b^{gender}_{static} is 1 if the patient is female and 0 otherwise:

$$y_{gender} = 0.2 * b^{gender}_{static}, \quad \text{with } b^{gender}_{static} = \begin{cases} 1, & \text{if } x^{gender}_{static} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{B2}$$

The second part, y_{age}, demonstrates the effect of the static medical indicator *age* on the label, which we model as a downward open parabola:

$$y_{age} = -0.8 * (x^{age}_{static} - 0.5)^2 + 0.2. \tag{B3}$$

Besides the influence of static medical indicators on the label y, we also include an influence of sequential medical indicators. First, $b^{pattern}_{seq}$ is 1 if an instance contains a certain pattern in its sequence of activities regarding the administration of the medication, namely *"Medication A, Medication A, Medication A, Medication B"*:

$$y_{pattern} = 0.2 * b^{pattern}_{seq}, \quad \text{with } b^{pattern}_{seq} = \begin{cases} 1, & \text{if } x^{meda}_{t,seq} = 1, \forall t \in \{8, \ldots, 11\} \\ & \wedge\, x^{medb}_{12,seq} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{B4}$$

Further, y_{hr-nl} shows the effect of the medical indicator *Heart Rate* at time step t_2 as a downward open parabola:

$$y_{hr-nl} = -0.8 * (x^{hr}_{2,seq} - 0.5)^2 + 0.2. \tag{B5}$$

Lastly, y_{hr} describes the effect of the behavior of the medical indicator *Heart Rate* over time. The values can increase or decrease over time for a single patient pathway. If the values increase, b^{hr}_{seq} will take the value 1, otherwise it will be 0:

$$y_{hr} = 0.2 * b^{hr}_{seq}, \quad \text{with } b^{hr}_{seq} = \begin{cases} 1, & \text{if } x^{hr}_{t,seq} - x^{hr}_{t-1,seq} > 0, \text{ for } t \in \{2, 3, 4\}, \\ 0, & \text{otherwise.} \end{cases} \tag{B6}$$

B.2 Experimental setting

We optimize PatWay-Net based on the complete simulation data set comprising 50,000 patient pathways and then generate interpretation plots based on 1,000 patient pathways of the same data set. Furthermore, we do not consider any sequential medical indicator interaction as we are interested in investigating how well the model can learn the five additive parts described in the previous section. Finally, we set the hidden size per sequential and static medical indicator to 16, and the batch size, number of epochs, and learning rate to 32, 1,000, and 0.001, respectively, as the loss converged well with these values.

B.3 Results

We present PatWay-Net's interpretation plots for the created synthetic event log and, based on the interpretation plots, we validate how well PatWay-Net captures the effects we modeled in the synthetic event log data.

B.3.1 Importance of medical indicators

Figure 11 shows the medical indicator importance plot at time step t_{12}. As expected, the medical indicators *Gender*, *Heart Rate*, *Age*, *Medication A*, and *Medication B* show an effect on the model output. The remaining indicators correctly show no effect, as they were only included in the simulation to serve as irrelevant noise terms.

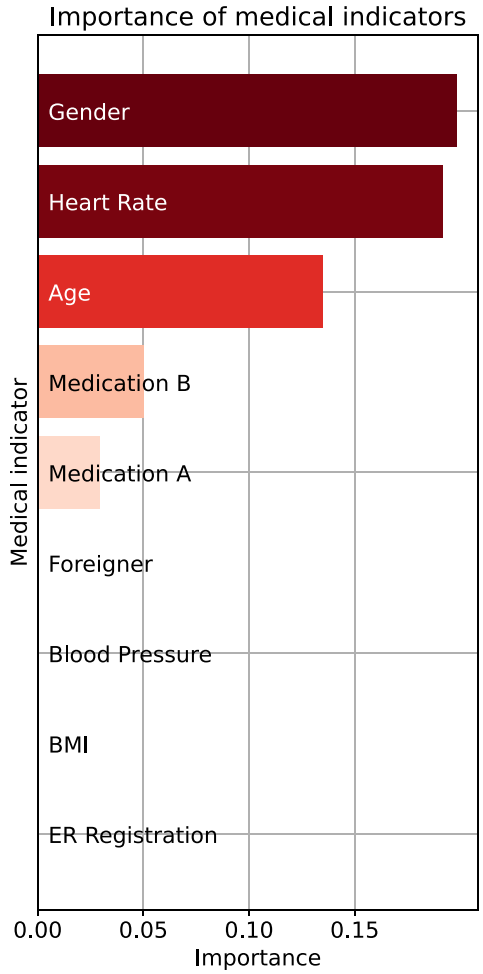


Fig. 11 Importance for static and sequential medical indicators

B.3.2 Static medical indicator shape

Figure 12 compares the global effect of the model output for the static categorical medical indicators *Gender* (left) and *Foreigner* (right). PatWay-Net correctly detects the effect of the medical indicator *Gender* with a constant value of 0.2, which equals the magnitude of the simulated coefficient whenever the indicator value is 1 (= female). By contrast, the indicator *Foreigner* has no effect.

Similarly, Fig. 13 shows the static medical indicator shape plot for indicator *Age* (left) compared to the indicator *BMI* (right).[8] PatWay-Net can correctly detect the parabolic effect on the label of the medical indicator *Age*, with the strongest effect of 0.2, being achieved at a value of 0.5, as modeled in Eq. B3. By contrast, no influence was correctly detected for the indicator *BMI*.

[8] For better interpretability, we applied post-processing to the effect values by adjusting the y-axis to 0.

B.3.3 Sequential medical indicator shape

For the sequential medical indicators *Heart Rate* and *Blood Pressure*, the indicator effect on the prediction can be evaluated for each time step. Figure 14 shows the sequential medical indicator shape for *Heart Rate* (left) and *Blood Pressure* (right) exemplarily for the time step t_4 and t_7, respectively.[9] PatWay-Net can correctly detect the quadratic effect on the label for the indicator *Heart Rate*. Indicator *Blood Pressure*, on the other hand, has no effect, as correctly detected by PatWay-Net.

B.3.4 Sequential medical indicator transition

As sequential medical indicators may change over time, it is crucial to investigate their effect not only at a certain time step but also the transition between two consecutive time steps. Figure 15 shows exemplarily the sequential medical indicator transition of the indicator value as well as the change of effect on the prediction for the indicators *Heart Rate* and *Blood Pressure* from time step t_3 (previous) to time step t_4 (current). For the medical indicator *Heart Rate*, the indicator value can either linearly increase or linearly decrease by 30% from time step t_3 (x-axis) to t_4 (y-axis). For increasing cases, we can observe the effect changes stronger negatively, that is, decreases from t_3 to t_4, from any indicator value to a higher indicator value (blue region). If the increase concerns two indicator values with a lower value, the change in the indicator effect is lower. This is due to the parabolic shape of the medical indicator *Heart Rate* (see Fig. 14) as defined in Eq. B6. For the decreasing cases, we can observe the effect changes stronger positively, that is, increases from t_3 to t_4, from a higher to a lower indicator value (red region). As for increasing cases, if the decrease concerns two indicator values with a lower value, the change in the indicator effect is lower. Again, this is due to the parabolic shape of the medical indicator *Heart Rate* (see Fig. 14).

For the medical indicator *Blood Pressure*, the indicator values can either randomly increase or decrease from time step t_6 (x-axis) to t_7 (y-axis), as demonstrated by the plot (see Fig. 14). Further, the effect on the prediction of the medical indicator does not change over time, as the complete region is colored white, indicating a change in the indicator effect of 0. Thus, in summary, we can see that PatWay-Net correctly detects the effect of the transition of the *Heart Rate* as modeled in Eq. B6, whereas *Blood Pressure* does not affect the model output.

[9] We select these time steps because they are the last time steps at which a *Heart Rate* and *Blood Pressure* activity occur. However, the remaining figures can be found in the repository.

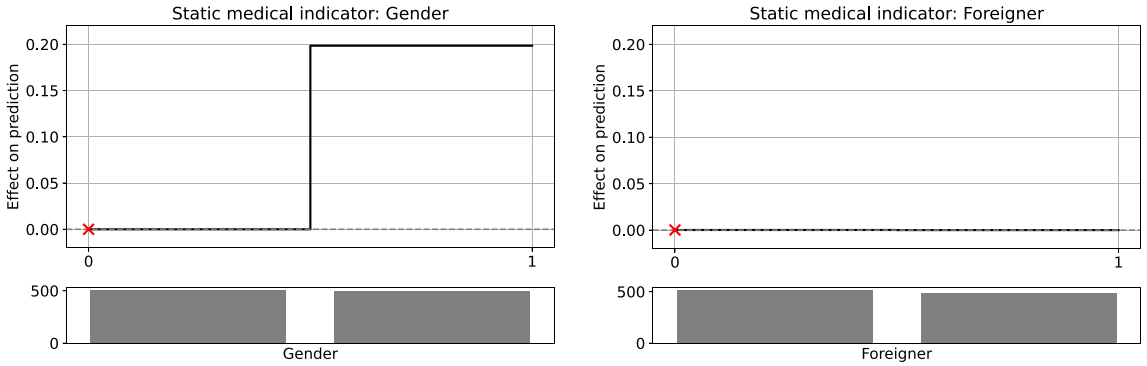


Fig. 12 Static medical indicator shape for indicators *Gender* (left) and *Foreigner* (right)

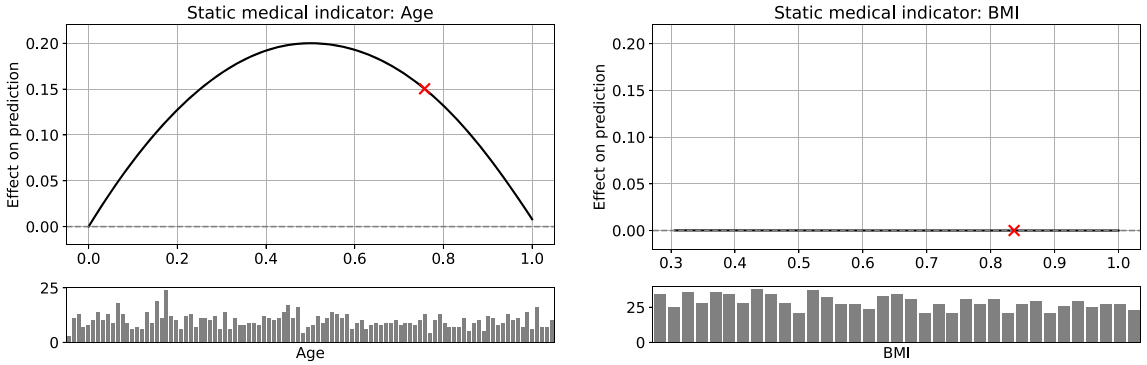


Fig. 13 Static medical indicator shape for indicators *Age* (left) and *BMI* (right)

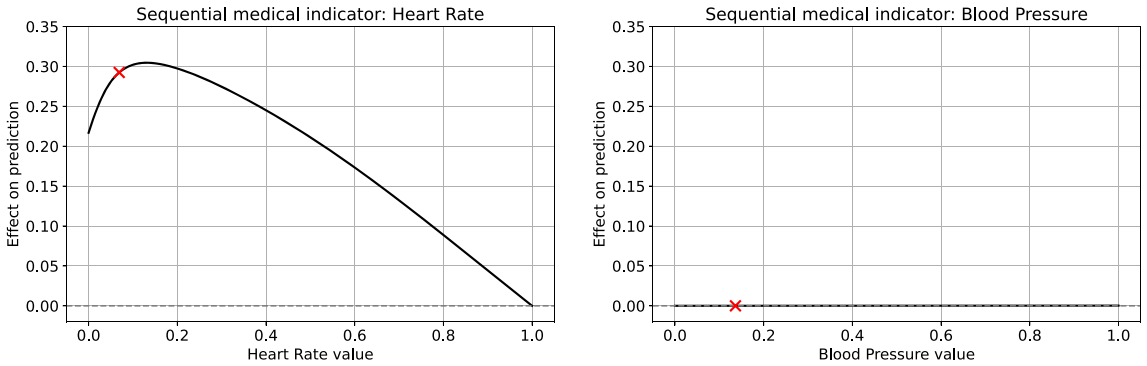


Fig. 14 Sequential medical indicator shape for indicators *Heart Rate* (left) and *Blood Pressure* (right)

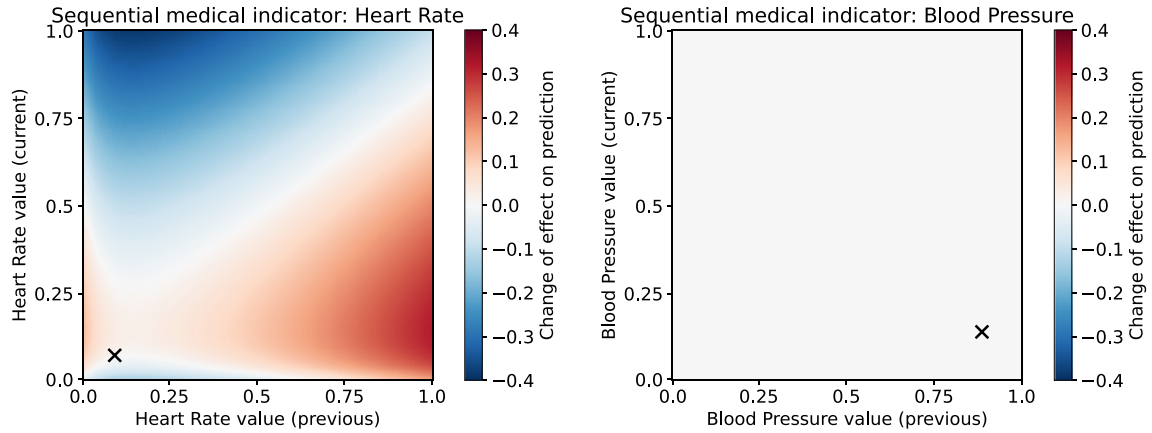


Fig. 15 Sequential medical indicator transition for indicators *Heart Rate* (left) and *Blood Pressure* (right)

B.3.5 Sequential medical indicator development

Figure 16 demonstrates the sequential medical indicator effect on the prediction for the indicator *Heart Rate* for a given patient pathway over time. The strongest increase in the indicator effect can be observed from time step t_1 to t_2, as *Heart Rate* is first measured at time step t_2. From t_2 to t_3, and from t_3 to t_4, the indicator effect increases more weakly, while the *Heart Rate* value steadily decreases. After time step t_4, the effect remains the same, as the activity *Heart Rate* does not occur after that.

B.4 Additional results

We conduct additional experiments with the created synthetic event log and interpretable baselines regarding training performance and interpretability. Table 10 summarizes the average train performance of PatWay-Net models without an interaction and baseline models over five seeds. The train performance indicates how well ML models capture the effects we modeled in the synthetic data. For measuring the training performance, we calculate the mean squared error (MSE), mean absolute error (MAE), and R^2. As baselines, we use the interpretable shallow ML approaches ridge regression, lasso regression, and decision tree regression. For training the lasso regression models and decision tree regression models, we set the regularization strength to 0.01 and the maximum depth to 3, respectively. We use default values for the remaining hyperparameters. We observe that the PatWay-Net models considerably outperform the baseline models in terms of all regression measures. This indicates that the PatWay-Net models can better capture the modeled effects in the synthetic data.

Figure 17 shows the loss curve for the training of PatWay-Net. The loss curve ranges from epoch 10 to 100 for better readability. Loss indicates the MSE. MSE values of the baseline models are added for a direct comparison. Based on this figure, the MSE values for the PatWay-Net model are increasingly lower from epoch 20 to 100. Therefore, the longer we train the PatWay-Net model, the better it can capture the modeled effects in the synthetic data.

Tables 11, 12, and Fig. 18 represent coefficients of the ridge regression model, coefficients of the lasso regression model, and the structure of the decision tree regression

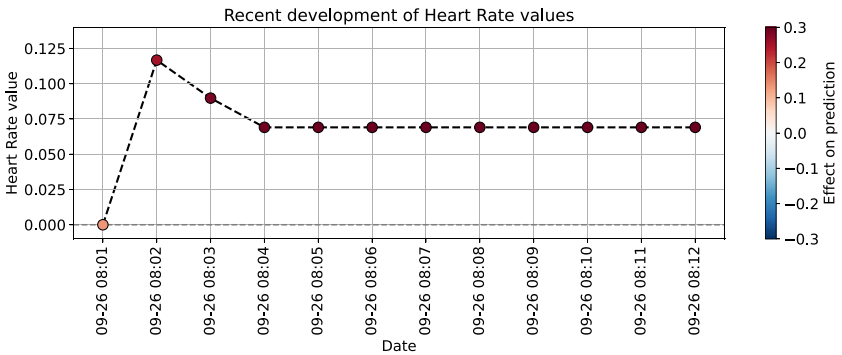


Fig. 16 Sequential medical indicator development for indicator *Heart Rate*

Table 10 Train performances for the simulation study

	Train MSE	Train MAE	Train R^2
PatWay-Net (without interaction)	0.000 (±.000)	0.002 (±.000)	0.997 (±.000)
Ridge regression	0.024 (±.000)	0.124 (±.000)	0.290 (±.000)
Lasso regression	0.024 (±.000)	0.126 (±.000)	0.278 (±.000)
Decision tree	0.021 (±.000)	0.120 (±.000)	0.379 (±.000)

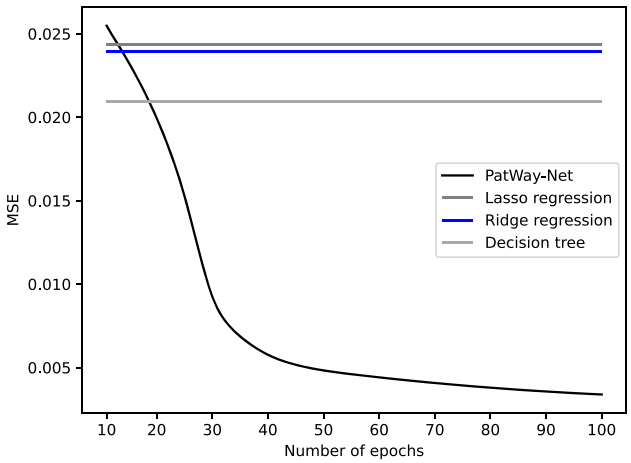


Fig. 17 Loss of PatWay-Net (from epoch 10 to 100) and baselines

Table 11 Model coefficients for static medical indicators of ridge regression

Static indicators	Gender	Foreigner	BMI	Age
Model coefficients	0.197653	0.000666	-0.001051	0.007274

Table 12 Model coefficients for static medical indicators of lasso regression

Static indicators	Gender	Foreigner	BMI	Age
Model coefficients	0.157634	0.00	-0.00	0.00

model, respectively. By comparing the different interpretation plots of PatWay-Net presented in Appendix B with the outputs of the baseline models (i.e., model coefficients and tree structure), it becomes clear that PatWay-Net with its interpretation plots provides much more comprehensible model outputs than the interpretable shallow ML models.

Appendix C Expert interviews

To assess PatWay-Net's interpretation quality, we conducted structured interviews with four independent medical experts. The medical experts are clinicians in different hospitals in different fields (surgery, trauma surgery, internal medicine, and anesthesiology), but are not directly associated with our use case. They have between five and ten years of professional experience.

The interviews were structured into five parts. First, all relevant information on the use case of patients with symptoms of sepsis was described to the interviewees. Second, a rough overview of different patient pathways was illustrated using a visualized process model. Third, general questions on predictive applications and their practical relevance were discussed. Fourth, we provided PatWay-Net's visual output and some additional descriptions of the interpretation in multiple steps. Based on this, we asked questions regarding usefulness, applicability, and trust in the predic-

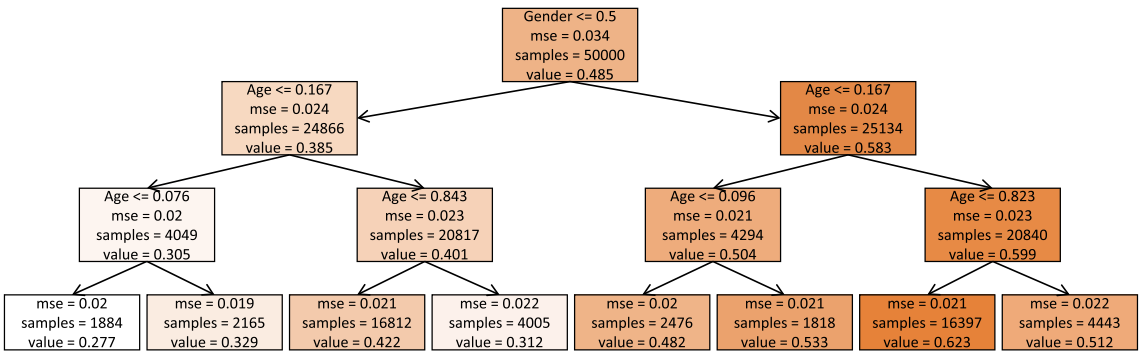


Fig. 18 Decision tree model with depth 3

tions. Lastly, overarching questions on the dashboard were asked.[10]

In general, the benefit of applying ML models in the medical domain was confirmed by the interviewees, as "*the idea of supporting medical differentiation through ML makes a lot of sense*" **(I4)**. Specifically, using predictive approaches is helpful **(I2)** and "*important for quick decision-making and provision or estimation of the required resources*" **(I1)**. It enables "*better assessment of patient risk, prioritization, [...], and transfer to another hospital if necessary*" **(I4)**. Additionally, it fosters to "*properly assess the patients*" **(I2)** and to "*order closer monitoring of the blood values*" **(I3)**. Further, it enables to "*better assess capacity*" **(I2)**, so that practitioners "*can also adjust ICU beds*" **(I3)**. In particular, the admission to ICU is helpful for medical experts with less experience: "*For younger colleagues [...] who don't have that much experience with [determination of ICU admission] yet, it's certainly helpful, also just to make sure that nothing is overlooked*" **(I2)**.

However, all interviewees confirmed that they "*would not trust such predictions without further explanation*" **(I1)** as they "*think it's important to know what the decision is based on*" **(I2)**. Clinicians "*have to justify the process to the patients and relatives and to do that [they] want to understand the process to be able to justify it*" **(I4)**.

Our dashboard was evaluated to be "*helpful as a support for decision-making*" **(I1)**, because "*it is similar in processes to [their] own decision-making*" **(I1)** and it is "*prepared very well*" **(I3)**. Additionally, the interviewees "*believe these tools will not miss findings and will guide you to look at everything again when it automatically shows up in an overview*" **(I1)**.

Moreover, all medical experts confirm the usefulness of the interpretation plots as, for example, **I2** "*find[s] it helpful in any case [to] have an overview of which factors have been included [and to] [...] understand at a glance what caused the system to do this. [...] Because then [they] can also take a closer look at what the blood pressure is doing or where the leukocytes are*". They positively assess the visual plots and think that "*this is the language that is spoken medically in actually every continent*" **(I3)**. All interviewees stated that they prefer simple plots, because "*there are always so many new colleagues [...] in the clinic that I think keeping it as simple as possible makes the most sense*" **(I2)** and that they usually have to act "*relatively quickly*" **(I2, I3)**.

They prefer the suggested plots over common SHAP plots because they look "*clearer and [are] easier to follow*" **(I2, I4)**. They feel that SHAP plots are "*confusing and not really self-explaining*" because there are "*too many dots in the plot*" **(I3)** (cf. also Appendix A). Finally, one interviewee suggested providing interpretations in the form of "*written text*" **(I4)**.

All interviewees also confirm that interpretations in the form of PatWay-Net's dashboard "*would positively influence [their] [...] trust in the predictions*" **(I1, I4)**, because the dashboard provides "*exactly what [they] work out [themselves] during a [...] visit*" **(I3)**. Therefore, they think that it "*increases [the] acceptance of such predictions*" **(I3)**.

Appendix D Additional use cases

To prove PatWay-Net's robustness and generalizability, we use two additional data sets from different domains (i.e., hospital billing and loan application). The event logs are publicly available, and we use the version by Teinemaa et al. [77] as they include labels for process outcome prediction. The data sets are preprocessed in the same way as for the event log (Appendix A.1). To keep the computational effort manageable, we only used the first 5,000 traces. Interactions of the models are detected and incorporated as for the use case (Appendix A.2). The hyperparameters used for the experiments can be found in the repository.

First, we use the *hospital billing*[11] event log. It captures billing data from an ERP system of a hospital about conducted services. Its preprocessed version contains two static medical indicators. These two indicators, namely *speciality* and *caseType*, are categorical and contain 22 and eight different values, respectively. Besides the activity with 18 different values, the preprocessed event log contains another sequential medical indicator, namely *state*, which is also categorical with ten different values. Finally, the label indicates whether a case is reopened [77]. The second data set is the *bpi2012*[12] event log. The data was taken from a Dutch financial institute and captures the process of loan applications. After preprocessing, it contains one static numerical medical indicator, namely *amount_req*, and the activity includes 36 different values. The label indicates if the application was accepted.

The results for PatWay-Net and the baselines for both data sets are summarized in Tables 13 and 14. For both event logs, we can show that PatWay-Net outperforms the baselines regarding the AUC_{ROC}. For the bpi2012 event log, we observe that PatWay-Net also outperforms the baselines regarding the F1-score.

[10] The interview guideline can be found in the online repository.

[11] https://research.tue.nl/en/datasets/hospital-billing-event-log

[12] https://data.4tu.nl/articles/dataset/BPI_Challenge_2012/12689204

Table 13 Comparison between baseline models and PatWay-Net for the hospital billing event log

ML approach	F1-score (weighted)		AUC_{ROC}	
	Validation	Test	Validation	Test
OUR APPROACH				
PatWay-Net (with interaction)	0.944 (±.006)	0.950 (±.004)	0.691 (±.049)	0.627 (±.040)
PatWay-Net (without interaction)	0.944 (±.006)	0.950 (±.004)	0.694 (±.053)	0.614 (±.056)
INTERPRETABLE SHALLOW MACHINE LEARNING				
Decision tree	0.944 (±.006)	0.950 (±.005)	0.589 (±.071)	0.546 (±.043)
K-nearest neighbor	0.932 (±.010)	0.936 (±.012)	0.512 (±.020)	0.534 (±.023)
Naïve Bayes	0.942 (±.004)	0.947 (±.010)	0.670 (±.051)	0.605 (±.048)
Logistic regression	0.944 (±.006)	0.950 (±.004)	0.674 (±.055)	0.599 (±.049)
NON- INTERPRETABLE MACHINE LEARNING				
LSTM network (with static module)	0.966 (±.014)	0.964 (±.007)	0.814 (±.057)	0.758 (±.103)
XGBoost	0.944 (±.006)	0.951 (±.005)	0.696 (±.054)	0.623 (±.038)
Random forest	0.944 (±.006)	0.951 (±.005)	0.692 (±.060)	0.626 (±.047)

Table 14 Comparison between baseline models and PatWay-Net for the bpi2012 event log

ML approach	F1-score (weighted)		AUC_{ROC}	
	Validation	Test	Validation	Test
OUR APPROACH				
PatWay-Net (with interaction)	0.638 (±.008)	0.662 (±.003)	0.740 (±.006)	0.750 (±.003)
PatWay-Net (without interaction)	0.647 (±.006)	0.665 (±.009)	0.741 (±.007)	0.750 (±.005)
INTERPRETABLE SHALLOW MACHINE LEARNING				
Decision tree	0.319 (±.012)	0.369 (±.002)	0.519 (±.005)	0.529 (±.007)
K-nearest neighbor	0.512 (±.026)	0.527 (±.019)	0.528 (±.024)	0.534 (±.021)
Naïve Bayes	0.412 (±.075)	0.458 (±.072)	0.515 (±.014)	0.542 (±.029)
Logistic regression	0.320 (±.012)	0.369 (±.002)	0.473 (±.013)	0.490 (±.016)
NON- INTERPRETABLE MACHINE LEARNING				
LSTM network (with static module)	0.648 (±.014)	0.672 (±.009)	0.744 (±.008)	0.760 (±.012)
XGBoost	0.448 (±.027)	0.509 (±.032)	0.520 (±.003)	0.554 (±.013)
Random forest	0.352 (±.040)	0.414 (±.065)	0.521 (±.010)	0.537 (±.012)

Funding Open Access funding enabled and organized by Projekt DEAL. Mathias Kraus and Patrick Zschech acknowledge funding from the Federal Ministry of Education and Research (BMBF) on "White-Box-AI" (Grant 01IS22080). Patrick Zschech acknowledges funding from the Federal Ministry of Education and Research (BMBF) on "AddIChron" (Grant 16SV8995). Martin Matzner acknowledges funding from the German Reseach Foundation (DFG) on "CoPPA" (Grant 456415646).

Data Availability Only a public data set was used, as cited in the text.

Code availability The code is available under the link provided in the text.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Rojas E, Munoz-Gama J, Sepúlveda M, Capurro D (2016) Process mining in healthcare: A literature review. J Biomed Inform 61:224–236. https://doi.org/10.1016/j.jbi.2016.04.007
2. Morton A, Bish E, Megiddo I, Zhuang W, Aringhieri R, Brailsford S, Deo S, Geng N, Higle J, Hutton D et al (2021) Introduction to the special issue: Management science in the fight against Covid-19. Health Care Manage Sci 24(2):251–252. https://doi.org/10.1007/s10729-021-09569-x
3. Bertsimas D, Boussioux L, Cory-Wright R, Delarue A, Digalakis V, Jacquillat A, Kitane DL, Lukin G, Li M, Mingardi L et al (2021) From predictions to prescriptions: A data-driven response to COVID-19. Health Care Manage Sci 24:253–272. https://doi.org/10.1007/s10729-020-09542-0
4. Janiesch C, Zschech P, Heinrich K (2021) Machine learning and deep learning. Electron Mark 31(3):685–695. https://doi.org/10.1007/s12525-021-00475-2
5. Kraus M, Feuerriegel S, Oztekin A (2020) Deep learning in business analytics and operations research: Models, applications and managerial implications. Eur J Oper Res 281(3):628–641. https://doi.org/10.1016/j.ejor.2019.09.018
6. Barrera Ferro D, Brailsford S, Bravo C, Smith H (2020) Improving healthcare access management by predicting patient no-show behaviour. Decis Support Syst 138:113398. https://doi.org/10.1016/j.dss.2020.113398
7. Yang C-S, Wei C-P, Yuan C-C, Schoung J-Y (2010) Predicting the length of hospital stay of burn patients: Comparisons of prediction accuracy among different clinical stages. Decis Support Syst 50(1):325–335. https://doi.org/10.1016/j.dss.2010.09.001
8. Elitzur R, Krass D, Zimlichman E (2023) Machine learning for optimal test admission in the presence of resource constraints. Health Care Manage Sci 1–22. https://doi.org/10.1007/s10729-022-09624-1
9. Krämer J, Schreyögg J, Busse R (2019) Classification of hospital admissions into emergency and elective care: A machine learning approach. Health Care Manag Sci 22:85–105. https://doi.org/10.1007/s10729-017-9423-5
10. Komorowski M, Celi LA, Badawi O, Gordon AC, Faisal AA (2018) The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. Nat Med 24(11):716–1720. https://doi.org/10.1038/s41591-018-0213-5
11. Mannhardt F, Blinde D (2017) Analyzing the trajectories of patients with sepsis using process mining. In: Proceedings of the 18th international working conference on business process modeling, pp 72–80
12. Lee S-Y, Chinnam RB, Dalkiran E, Krupp S, Nauss M (2020) Prediction of emergency department patient disposition decision for proactive resource allocation for admission. Health Care Manage Sci 23(3):339–359. https://doi.org/10.1007/s10729-019-09496-y
13. Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N (2015) Intelligible models for health care: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1721–1730
14. Shipe ME, Deppen SA, Farjah F, Grogan EL (2019) Developing prediction models for clinical use using logistic regression: An overview. J Thorac Dis 11(S4):574–584. https://doi.org/10.21037/jtd.2019.01.25
15. Bertoncelli CM, Altamura P, Vieira ER, Iyengar SS, Solla F, Bertoncelli D (2020) Predictmed: A logistic regression–based model to predict health conditions in cerebral palsy. Health Inform J 26(3):2105–2118. https://doi.org/10.1177/14604582198985
16. Magunia H, Lederer S, Verbuecheln R, Gilot BJ, Koeppen M, Haeberle HA, Mirakaj V, Hofmann P, Marx G, Bickenbach J et al (2021) Machine learning identifies ICU outcome predictors in a multicenter COVID-19 cohort. Critical Care 25(1):295. https://doi.org/10.1186/s13054-021-03720-4
17. Bertsimas D, Dunn J, Gibson E, Orfanoudaki A (2022) Optimal survival trees. Mach Learn 111(8):2951–3023. https://doi.org/10.1007/s10994-021-06117-0
18. Liu M, Guo C, Guo S (2023) An explainable knowledge distillation method with XGBoost for ICU mortality prediction. Comput Biol Med 152:106466. https://doi.org/10.1016/j.compbiomed.2022.106466
19. Reddy BK, Delen D (2018) Predicting hospital readmission for lupus patients: An RNN-LSTM-based deep-learning methodology. Comput Biol Med 101:199–209. https://doi.org/10.1016/j.compbiomed.2018.08.029
20. Ye X, Zeng QT, Facelli JC, Brixner DI, Conway M, Bray BE (2020) Predicting optimal hypertension treatment pathways using recurrent neural networks. Int J Med Inform 139:104122. https://doi.org/10.1016/j.ijmedinf.2020.104122
21. Zilker S, Weinzierl S, Zschech P, Kraus M, Matzner M (2023) Best of both worlds: Combining predictive power with interpretable and explainable results for patient pathway prediction. In: Proceedings of the European Conference on Information Systems
22. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444. https://doi.org/10.1038/nature14539
23. Barredo Arrieta A, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, Garcia S, Gil-Lopez S, Molina D, Benjamins R, Chatila R, Herrera F (2020) Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fus 58:82–115. https://doi.org/10.1016/j.inffus.2019.12.012
24. Moerer O, Plock E, Mgbor U, Schmid A, Schneider H, Wischnewsky MB, Burchardi H (2007) A german national prevalence study on the cost of intensive care: An evaluation from 51 intensive care units. Critical Care 11(3):1–10. https://doi.org/10.1186/cc5952
25. Mans RS, van der Aalst WMP, Vanwersch RJB (2015) Process mining in healthcare: Evaluating and exploiting operational healthcare processes. Springer
26. Rebuge Á, Ferreira DR (2012) Business process analysis in healthcare environments: A methodology based on process mining. Inf Syst 37(2):99–116. https://doi.org/10.1016/j.is.2011.01.003
27. Wiens J, Saria S, Sendak M, Ghassemi M, Liu VX, Doshi-Velez F, Jung K, Heller K, Kale D, Saeed M et al (2019) Do no harm: A roadmap for responsible machine learning for health care. Nat Med 25(9):1337–1340. https://doi.org/10.1038/s41591-019-0548-6
28. Rhodes A, Evans LE, Alhazzani W, Levy MM, Antonelli M, Ferrer R, Kumar A, Sevransky JE, Sprung CL, Nunnally ME et al (2017) Surviving sepsis campaign: International guidelines for management of sepsis and septic shock: 2016. Intensive Care Med 43:304–377. https://doi.org/10.1007/s00134-017-4683-6

29. Rello J, Valenzuela-Sánchez F, Ruiz-Rodriguez M, Moyano S (2017) Sepsis: A review of advances in management. Adv Therapy 34:2393–2411. https://doi.org/10.1007/s12325-017-0622-8
30. Schuurman AR, Sloot PM, Wiersinga WJ, van der Poll T (2023) Embracing complexity in sepsis. Critical Care 27(1):102. https://doi.org/10.1186/s13054-023-04374-0
31. Thorsen-Meyer H-C, Nielsen AB, Nielsen AP, Kaas-Hansen BS, Toft P, Schierbeck J, Strøm T, Chmura PJ, Heimann M, Dybdahl L et al (2020) Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: A retrospective study of high-frequency data in electronic patient records. Lancet Digit Health 2(4):179–191. https://doi.org/10.1016/S2589-7500(20)30018-2
32. Loh HW, Ooi CP, Seoni S, Barua PD, Molinari F, Acharya UR (2022) Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022). Comput Methods Prog Biomed 226:107161. https://doi.org/10.1016/j.cmpb.2022.107161
33. Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell 1(5):206–215. https://doi.org/10.1038/s42256-019-0048-x
34. Moreno RP, Metnitz PGH, Almeida E, Jordan B, Bauer P, Campos RA, Iapichino G, Edbrooke D, Capuzzo M, Le Gall J-R, SAPS 3 Investigators (2005) SAPS 3–From evaluation of the patient to evaluation of the intensive care unit. Part 2: Development of a prognostic model for hospital mortality at ICU admission. Intensive Care Med 31(10): 1345–1355. https://doi.org/10.1007/s00134-005-2763-5
35. Galanti R, Coma-Puig B, d Leoni M, Carmona J, Navarin N (2020) Explainable predictive process monitoring. In: Proceedings of the 2nd international conference on process mining, pp 1–8
36. Weinzierl S, Zilker S, Brunk J, Revoredo K, Matzner M, Becker J (2020) XNAP: Making LSTM-based next activity predictions explainable by using LRP. In: Proceedings of the BPM 2020 international workshop, pp 129–141
37. Breuker D, Matzner M, Delfmann P, Becker J (2016) Comprehensible predictive models for business processes. MIS Quarterly 40(4):1009–1034
38. Lakshmanan GT, Shamsi D, Doganata YN, Unuvar M, Khalaf R (2015) A Markov prediction model for data-driven semi-structured business processes. Knowl Inf Syst 42(1):97–126. https://doi.org/10.1007/s10115-013-0697-8
39. Kaji DA, Zech JR, Kim JS, Cho SK, Dangayach NS, Costa AB, Oermann EK (2019) An attention based deep learning model of clinical events in the intensive care unit. PloS one 14(2):0211057. https://doi.org/10.1371/journal.pone.0211057
40. Zhang D, Yin C, Hunold KM, Jiang X, Caterino JM, Zhang P (2021) An interpretable deep-learning model for early prediction of sepsis in the emergency department. Patterns 2(2):100196. https://doi.org/10.1016/j.patter.2020.100196
41. Esteban C, Staeck O, Baier S, Yang Y, Tresp V (2016) Predicting clinical events by combining static and dynamic information using recurrent neural networks. In: Proceedings of the 2016 IEEE international conference on healthcare informatics, pp 93–101
42. Breiman L (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324
43. Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd international conference on knowledge discovery and data mining, pp 785–794
44. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
45. Rai A (2020) Explainable AI: From black box to glass box. J Acad Market Sci 48(1):137–141. https://doi.org/10.1007/s11747-019-00710-5
46. Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Proceedings of the 30th conference on advances in neural information processing systems, pp 4765–4774
47. Senoner J, Netland T, Feuerriegel S (2022) Using explainable artificial intelligence to improve process quality: Evidence from semiconductor manufacturing. Manage Sci 68(8):5704–5723. https://doi.org/10.1287/mnsc.2021.4190
48. Babic B, Gerke S, Evgeniou T, Cohen IG (2021) Beware explanations from AI in health care. Science 373(6552):284–286. https://doi.org/10.1126/science.abg1834
49. Zschech P, Weinzierl S, Hambauer N, Zilker S, Kraus M (2022) GAM(e) changer or not? An evaluation of interpretable machine learning models based on additive model constraints. In: Proceedings of the 30th European Conference on Information Systems, pp 1–18
50. Hastie T, Tibshirani R (1986) Generalized additive models. Stat Sci 1(3):297–310. https://doi.org/10.1214/ss/1177013604
51. Lou Y, Caruana R, Gehrke J, Hooker G (2013) Accurate intelligible models with pairwise interactions. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 623–631
52. Lou Y, Caruana R, Gehrke J (2012) Intelligible models for classification and regression. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 150–158
53. Yang Z, Zhang A, Sudjianto A (2021) Gami-net: An explainable neural network based on generalized additive models with structured interactions. Pattern Recogn 120:108192. https://doi.org/10.1016/j.patcog.2021.108192
54. Agarwal R, Melnick L, Frosst N, Zhang X, Lengerich B, Caruana R, Hinton GE (2021) Neural additive models: Interpretable machine learning with neural nets. In: Proceedings of the 34th Conference on Advances in Neural Information Processing Systems, pp 4699–4711
55. Kraus M, Tschernutter D, Weinzierl S, Zschech P (2023) Interpretable generalized additive neural networks. Eur J Oper Res. https://doi.org/10.1016/j.ejor.2023.06.032
56. Marquez-Chamorro AE, Resinas M, Ruiz-Cortes A (2018) Predictive monitoring of business processes: A survey. IEEE Trans Serv Comput 11(6):962–977. https://doi.org/10.1109/TSC.2017.2772256
57. Huang Z, Lu X, Duan H, Fan W (2013) Summarizing clinical pathways from event logs. J Biomed Inform 46(1):111–127. https://doi.org/10.1016/j.jbi.2012.10.001
58. Kingma D, Ba J (2015) Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations
59. Moreno R, Miranda DR (1998) Nursing staff in intensive care in europe: The mismatch between planning and practice. Chest 113(3):752–758. https://doi.org/10.1378/chest.113.3.752
60. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30. https://doi.org/10.5555/1248547.1248548
61. Klein SJ, Lehner GF, Forni LG, Joannidis M (2018) Oliguria in critically ill patients: A narrative review. J Nephrol 31:855–862. https://doi.org/10.1007/s40620-018-0539-6
62. Hotchkiss RS, Moldawer LL, Opal SM, Reinhart K, Turnbull IR, Vincent J-L (2016) Sepsis and septic shock. Nat Rev Dis Prim 2(1):1–21. https://doi.org/10.1038/nrdp.2016.45
63. Urrechaga E, Bóveda O, Aguirre U (2018) Role of leucocytes cell population data in the early detection of sepsis. J Clin Pathol 71(3):259–266. https://doi.org/10.1136/jclinpath-2017-204524
64. Pera A, Campos C, López N, Hassouneh F, Alonso C, Tarazona R, Solana R (2015) Immunosenescence: Implications for response to infection and vaccination in older people. Maturitas 82(1):50–55. https://doi.org/10.1016/j.maturitas.2015.05.004

65. Riley LK, Rupert J (2015) Evaluation of patients with leukocytosis. Am Fam Phys 92(11):1004–1011
66. Belok SH, Bosch NA, Klings ES, Walkey AJ (2021) Evaluation of leukopenia during sepsis as a marker of sepsis-defining organ dysfunction. PLoS One 16(6):0252206. https://doi.org/10.1371/journal.pone.0252206
67. Acosta JN, Falcone GJ, Rajpurkar P, Topol EJ (2022) Multimodal biomedical AI. Nat Med 28(9):1773–1784. https://doi.org/10.1038/s41591-022-01981-2
68. van Weenen E, Banholzer N, Föll S, Zueger T, Fontana FY, Skroce K, Hayes C, Kraus M, Feuerriegel S, Lehmann V et al (2023) Glycaemic patterns of male professional athletes with type 1 diabetes during exercise, recovery and sleep: Retrospective, observational study over an entire competitive season. Diabetes, Obesity and Metabolism. https://doi.org/10.1111/dom.15147
69. Cawley GC, Talbot NL (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. J Mach Learn Res 11:2079–2107. https://doi.org/10.5555/1756006.1859921
70. Cawley GC, Talbot NL (2007) Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters. J Mach Learn Res 8(4). https://doi.org/10.5555/1248659.1248690
71. Qi Y, Minka TP, Picard RW, Ghahramani Z (2004) Predictive automatic relevance determination by expectation propagation. In: Proceedings of the 21st International Conference on Machine Learning, p 85
72. Slack D, Krishna S, Lakkaraju H, Singh S (2023) Explaining machine learning models with interactive natural language conversations using TalkToModel. Nat Mach Intell 5(8):873–883. https://doi.org/10.1038/s42256-023-00692-8
73. Feuerriegel S, Hartmann J, Janiesch C, Zschech P (2023) Generative AI. Business & Information Systems Engineering. https://doi.org/10.1007/s12599-023-00834-7
74. Davis J, Goadrich M (2006) The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning, pp 233–240
75. McClish DK (1989) Analyzing a portion of the ROC curve. Med Dec Making 9(3):190–195. https://doi.org/10.1177/0272989X89009003
76. Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2
77. Teinemaa I, Dumas M, Rosa ML, Maggi FM (2019) Outcome-oriented predictive process monitoring: Review and benchmark. ACM Trans Knowl Dis Data 13(2):1–57. https://doi.org/10.1145/3301300

Authors and Affiliations

Sandra Zilker[1,2] · Sven Weinzierl[2] · Mathias Kraus[3] · Patrick Zschech[4] · Martin Matzner[2]

Sven Weinzierl
sven.weinzierl@fau.de

Mathias Kraus
mathias.kraus@informatik.uni-regensburg.de

Patrick Zschech
patrick.zschech@uni-leipzig.de

Martin Matzner
martin.matzner@fau.de

[1] Technische Hochschule Nürnberg Georg Simon Ohm, Professorship for Business Analytics, Hohfederstraße 40, 90489 Nuremberg, Germany

[2] Friedrich-Alexander-Universität Erlangen-Nürnberg, Chair of Digital Industrial Service Systems, Fürther Straße 248, 90429 Nuremberg, Germany

[3] University of Regensburg, Chair for Explainable AI in Business Value Creation, Bajuwarenstraße 4, 93053 Regensburg, Germany

[4] Leipzig University, Professorship for Intelligent Information Systems and Processes, Grimmaische Straße 12, 04109 Leipzig, Germany