



## OPEN ACCESS

## EDITED BY

Harry Hochheiser,  
University of Pittsburgh, United States

## REVIEWED BY

Moom Rahman Roosan,  
Chapman University, United States  
Watjana Lilaonitkul,  
University College London, United Kingdom

## \*CORRESPONDENCE

Robert C. Free  
✉ rob.free@leicester.ac.uk

RECEIVED 08 June 2023

ACCEPTED 18 September 2023

PUBLISHED 09 October 2023

## CITATION

Free RC, Lozano Rojas D, Richardson M, Skeemer J, Small L, Haldar P and Woltmann G (2023) A data-driven framework for clinical decision support applied to pneumonia management. *Front. Digit. Health* 5:1237146. doi: 10.3389/fdgth.2023.1237146

## COPYRIGHT

© 2023 Free, Lozano Rojas, Richardson, Skeemer, Small, Haldar and Woltmann. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A data-driven framework for clinical decision support applied to pneumonia management

Robert C. Free<sup>1\*</sup>, Daniel Lozano Rojas<sup>2</sup>, Matthew Richardson<sup>2</sup>, Julie Skeemer<sup>3</sup>, Leanne Small<sup>3</sup>, Pranabashis Haldar<sup>2</sup> and Gerrit Woltmann<sup>2,3</sup>

<sup>1</sup>School of Computing and Mathematical Sciences, University of Leicester, Leicester, United Kingdom,

<sup>2</sup>Department of Respiratory Sciences, University of Leicester, Leicester, United Kingdom, <sup>3</sup>Respiratory Medicine and Allergy Department, Glenfield Hospital, Leicester, United Kingdom

Despite their long history, it can still be difficult to embed clinical decision support into existing health information systems, particularly if they utilise machine learning and artificial intelligence models. Moreover, when such tools are made available to healthcare workers, it is important that the users can understand and visualise the reasons for the decision support predictions. Plausibility can be hard to achieve for complex pathways and models and perceived “black-box” functionality often leads to a lack of trust. Here, we describe and evaluate a data-driven framework which moderates some of these issues and demonstrate its applicability to the in-hospital management of community acquired pneumonia, an acute respiratory disease which is a leading cause of in-hospital mortality world-wide. We use the framework to develop and test a clinical decision support tool based on local guideline aligned management of the disease and show how it could be used to effectively prioritise patients using retrospective analysis. Furthermore, we show how this tool can be embedded into a prototype clinical system for disease management by integrating metrics and visualisations. This will assist decision makers to examine complex patient journeys, risk scores and predictions from embedded machine learning and artificial intelligence models. Our results show the potential of this approach for developing, testing and evaluating workflow based clinical decision support tools which include complex models and embedding them into clinical systems.

## KEYWORDS

clinical decision support, pneumonia, artificial intelligence, machine learning, data-driven

## 1. Introduction

The concept of clinical decision support (CDS) goes back to the 1960s (1) but it was only when digitisation of patient data became a reality in the 1980s that CDS systems became possible. The purpose of CDS is to facilitate optimised and standardised healthcare and as a result modern CDS can vary in complexity from clinical risk score calculators, to medication and patient pathway management and prognostic and outcome modelling. However, options for what is possible within the hospital infrastructure depend on the degree of integration of the electronic patient record (EPR). Where users routinely access disparate or linked systems, CDS provision may become limited or unviable. From a technical and cost standpoint, it can also be difficult to setup and maintain CDS tools. If not embedded at the EPR design stage, CDS tools are often disease/discipline specific, standalone and/or proprietary and not fully interoperable with other information systems (2). Incomplete system integration necessitates duplicate data entry, which not only adds to the healthcare workers' workload but can introduce transcription errors. The resultant

cognitive overload and trigger fatigue effectively prohibits preemptive identification and stratification of patients according to clinical risk. This occurs without considering the complexity added by any data processing and lack of system interoperability.

Initially CDS were knowledge-based expert systems driven by a defined set of logical rules to drive the decision making—essentially a series of if/then steps. It is only more recently that non-knowledge based CDS which utilise machine learning (ML) and artificial intelligence (AI) models have been implemented (3). These latter approaches use existing data as a basis for models which identify patterns corresponding to specific predictions—patterns which they can subsequently recognise in any new data presented to them (4). For example, a CDS containing an ML model trained from data containing markers of 30-day mortality for a particular disease, could use this to identify which patients should be prioritised when supplied with new data. However, the growing complexity and scale of AI and ML models used for CDS means they need access to multi-modal data sources such as images, omics and longitudinal test results from separate systems. This requires prior integration into EPRs to become relevant to clinical management. Additionally, the increased use of non-knowledge based CDS presents ethical concerns. The complexity of these (often proprietary) models makes them appear to be a “black-box.” Understanding these models is required to engender trust in predictions, particularly how such models perform on local data, so that users are encouraged to use the full potential of the system. Perceived lack of transparency, plausibility and trust in accuracy limits generalisability for clinical practitioners who then often prefer to rely on years of knowledge and experience (5).

Here we describe an approach for developing CDS which considers some of the concerns above. This is underpinned by a generic software framework for creating and handling real-time CDS called Embeddable AI and State-based Understandable Logic (EASUL), built based on our experience with management of community acquired pneumonia (CAP) admissions locally. CAP is an acute lung infection, which is the leading cause of death in UK hospitals, with mortality associated with admissions varying from between 2% and 30% depending on disease severity, comorbidities and age. In addition, it is estimated that annual healthcare costs to the UK National Health Service (NHS) associated with CAP exceed £1 billion. In our local hospital, CAP patients are managed and supported by a specialist pneumonia intervention nurse (SPIN) team and in our previous collaboration we showed that prompt interventions are associated with significantly improved outcomes (6). Here we demonstrate how enhancing CAP management with our informatics approach could enable better and more timely use of data and potentially further improve prioritisation and care of patients with more severe disease.

## 2. Method

### 2.1. The EASUL framework

The proof-of-concept EASUL framework is based on a Python-based library which enables data-driven CDS plans to be setup and

subsequently executed. **Figure 1** shows how the EASUL framework can be configured in different ways to develop CDS tools. Key to this are plans which contain the data sources, steps, algorithms, states and visualisations representing potential patient journey.

Key to these plans are a series of determinative and connected steps as shown in **Figure 2**. These support algorithms of different modalities—varying from simple clinical risk scores and logical (if/then) comparisons to advanced ML and AI models—to determine a specific patient journey. To simplify the integration of scores and models we also included data-related concepts in EASUL (schemas, algorithms and data sets) which allow the quick implementation and inclusion of new and existing models within plans. EASUL also contains tools to serialize algorithms and calculate appropriate metadata (including model performance metrics, model explainers and other values) and specific result contexts which are used to generate appropriate visualisations. To improve performance, the library can pre-calculate and cache metadata and result context to ensure visualisations are provided to the user in a timely fashion. The visualisations are produced using HTML, MermaidJS (7), and matplotlib (8). Information related to models such as area under the curve, sensitivity, specificity and accuracy; and measures such as explainers are generated using scikitlearn (9). For outputting explainable elements, the LIME library is used to calculate patient-specific explainers (10) and the results of the explanation are output using plain HTML. The framework is designed to be data source agnostic although data is converted internally into Python data structures. The framework currently supports text-delimited files, SQLite databases, and Redis-based messages (the demo admissions data feed), but could be extended to handle other data sources such as other database systems and biomedical data. Sources can also include on-the-fly processing functions to transform data as required. Furthermore, EASUL also has the capability to combine multiple data sources into one collated data source. There is also rudimentary support for handling missing data by including default values if variables are missing.

A key advantage of EASUL is that through its use of adaptable and flexible plans CDS tools can be initially defined, evaluated and tested using static data sources, before they are adapted into information systems driven through real-time data sources. This latter ability is implemented through an optional web API (built using the Django framework), which can serve outputs and results (visualisations and data related to the plan), and a flexible engine supporting use of different broker and client setups to message other systems such as EPRs or databases as required. This approach is generalisable and adaptable to different scenarios and conditions, potentially enabling EASUL to support decision making at any stage of the patient journey.

The current version of EASUL is available for download, along with documentation related to its use and configuration at <https://github.com/rcfgroup/easul>.

### 2.2. Data sources

To develop and evaluate the CAP management plan, individual hospital episodes from adults ( $\geq 16$  years old) admitted between

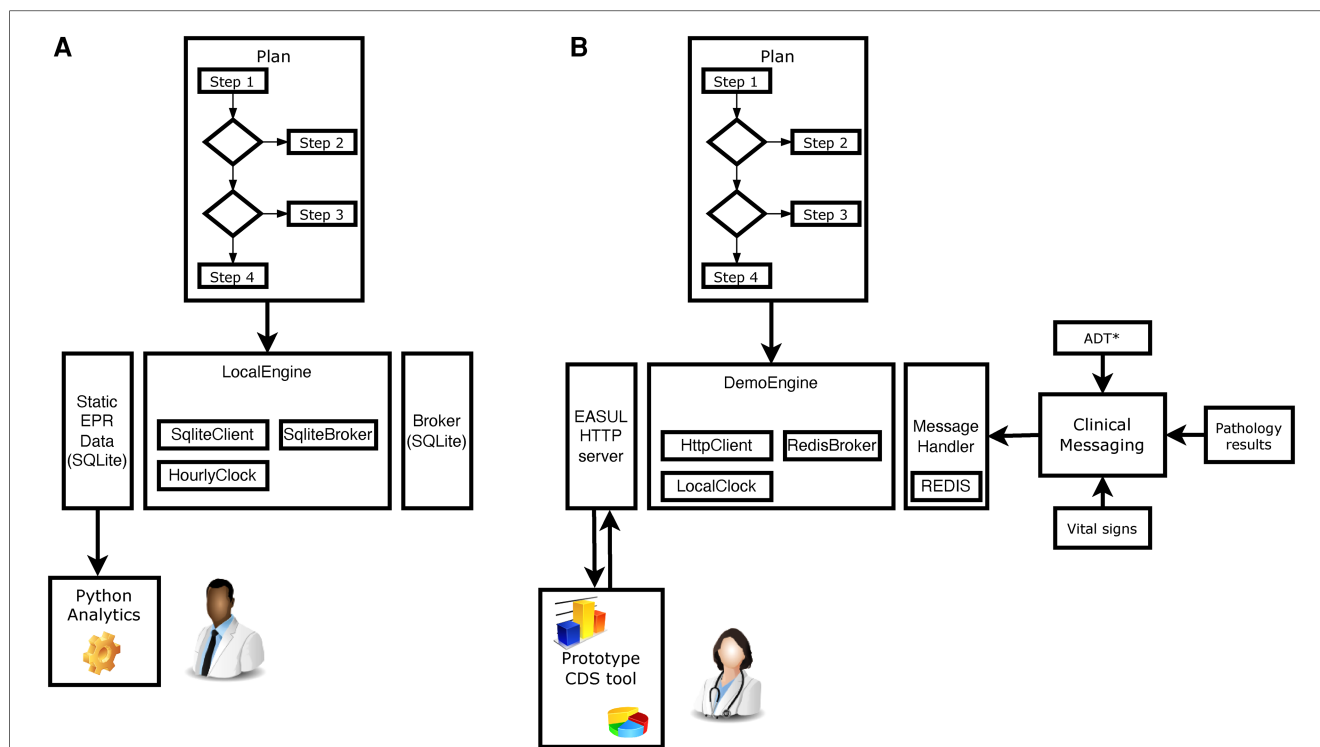


FIGURE 1

Schematic showing how EASUL-based tools were configured and utilised in two different ways. (A) For research, quality and service improvement using static data sets and Python scripting and analytics. (B) Creation of a prototype CDS tool through integration of outputs/results into a clinical information system. \*ADT = hospital admissions, discharges and transfers. In all cases, a Plan is initially defined using Python classes. These plans act as containers for available re-usable components including DataSource, Algorithm, State and Visual classes. The main logic is encompassed within determinative Steps, which support algorithms of different modalities—varying from simple clinical risk scores and logical (if/then) comparisons to advanced ML and AI models. Data received by steps at particular points determine the specific patient journey undertaken. Once a Plan has been defined, it is executed using an Engine - which encompasses a Client, Broker and Clock. The client handles the local storage of states and results, the broker provides/receives data to drive the plan and the clock handles the temporal aspects of flows. For example, in (A) the client was a SQLite database which stored information for later analysis, the broker was a static SQLite database which provided input data and the clock was setup to increment forward hourly within each CAP admission to simulate progression. Clipart is from draw.io.

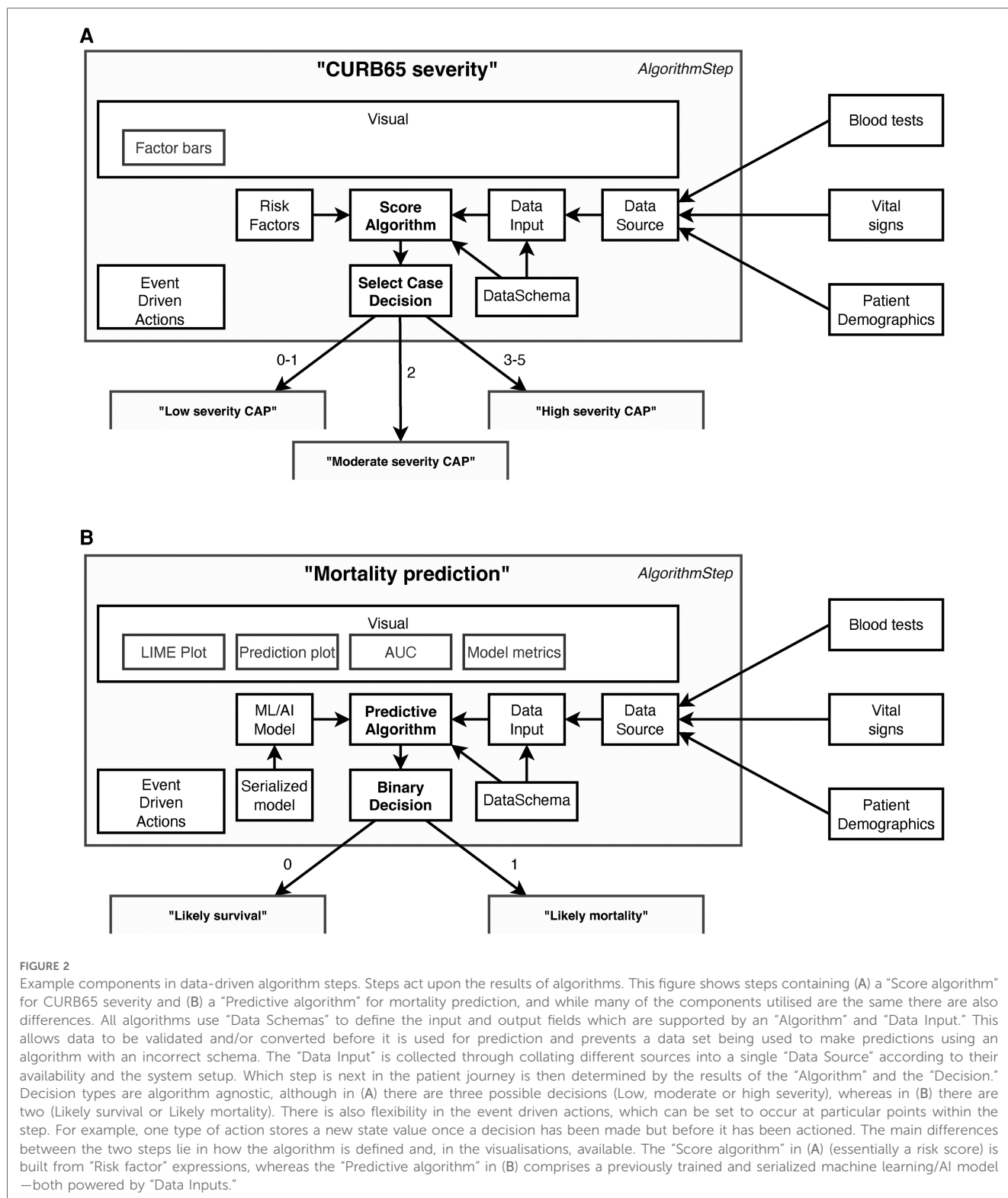
April 2022 and June 2022 were extracted from the hospital data warehouse as delimited text files. Alongside this, records of patients reviewed by the SPIN team were extracted from a locally run service database. The hospital data included demographics (patient ID, admission ID, dates and times of admission/discharge and the nature of the discharge) as well as International Statistical Classification of Diseases and Related Health Problems version 10 coded primary and secondary diagnoses, blood test results, oxygen records and drug records. The SPIN data included patient ID, dates/times of admission and CURB65 severity calculated by the SPIN team. Data files were de-identified prior to loading into a SQLite database for subsequent evaluation.

### 2.3. Decision workflow plan development and evaluation

An EASUL-based CAP management plan was defined in Python containing the appropriate steps and associated properties. The plan also included data transformations/processes so that EASUL would convert and process data in non-standard formats on-the-fly. This included date/time parsing and CAP

diagnosis based on an algorithm utilised previously (6) but refactored into an EASUL compatible process. For this development and evaluation, the SQLite database described above was established as the main data source in this plan.

To evaluate the utility of the CAP management plan two data sets were created. The first data set comprised data derived from executing the EASUL plan across all the admissions in the SQLite database to determine diagnoses, patient journeys, derived states and algorithm results. The second data set was created by extracting the SPIN review data from the SQLite database and linking it to the coded admission records through an anonymised patient ID and date/time matching algorithm, with admissions not reviewed by the service identified and filtered out. This resulted in two sets of delimited text files representing the CAP admissions: one containing the EASUL-related data for all admissions, including those determined to have CAP based on the above CAP diagnosis algorithm and the other admissions reviewed by the SPIN service. The two data sets were merged and then appropriate variables compared using Python and the pandas library. The primary comparison undertaken in our evaluation was between the level of severity recorded by the SPIN team and that determined by EASUL. As defined previously, in the UK CAP disease severity is



classified using CURB65 a composite validated clinical risk score which uses presence/absence of mental confusion; high blood urea; high respiratory rate; low blood pressure; and age (at or over 65) (11). Disease severity is then classified according to NHS guidelines as low (CURB 0-1), moderate (CURB 2) or high (CURB 3-5) (12). While EASUL was configured to automatically calculate CURB65 and the derived severity level for each hour of each CAP admission,

the SPIN data only contained a single record of a manually calculated CURB65 score. Therefore, we used this score to categorise severity and then compared it to the highest severity level determined by EASUL during an admission. The frequency of each severity pair (e.g. low-medium, high-high) was tallied. The plan did not include an ML model at this stage—this was only added in the prototype system.

## 2.4. Proof-of-concept CDS prototype system

Our CDS implementation utilised the Headfake library (13) as the basis of a bespoke in-silico “simulated admissions” tool. This output near real-time data matching the record/field formats obtained from the hospital data warehouse and pertaining to admissions/discharges and chest X-ray reports, pathology tests and vital signs—with the records encoded as simple JSON structures and being sent as messages at times during each “admission.” The EASUL plan was also altered to accept near real-time data from the broker and to embed an ML classifier from an ongoing project which predicts whether patients had died in-hospital or within 30 days of discharge from CAP (14). This classifier, based on tree gradient boosting and built using XGBoost (15) was wrapped as an EASUL algorithm along with a data schema and encoder/decoder designed to process routine hospital data into the correct format of data for input into the model. Additionally, visualisations were also defined representing this algorithm for both the overall model (e.g. performance metrics) and the row (e.g. breakdown of probabilities and other measures of interpretability). Due to the headless nature of EASUL, to provide an appropriate user interface for the CAP management system, we used a local bespoke clinical data collection platform and built a CAP data collection/management module and custom extensions to handle near real-time data from the EASUL broker.

## 3. Results

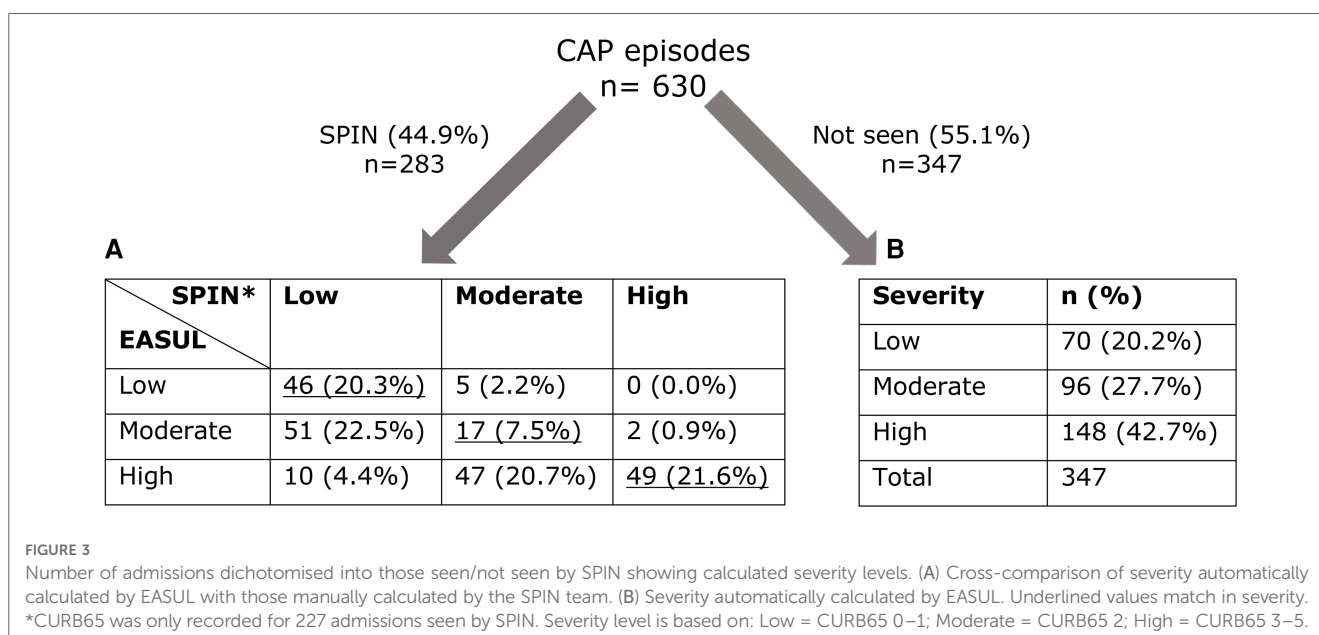
### 3.1. Creating and evaluating EASUL for CAP management

To demonstrate the potential utility of EASUL for CAP management CDS we worked with local physicians and members

of the SPIN team to identify the key steps and decision points in the current person-driven service process (6) and used EASUL to set this up as a data-driven workflow plan for CAP management.

The utility of this plan was assessed through a local quality improvement (QI) project. This involved comparing the results obtained from feeding retrospective admissions data into EASUL, with the actual events and information recorded by the service team. Essentially augmenting a manual process (performed by the SPIN team) with data-driven informatics (driven by EASUL) to prioritise patients for review. The admissions data set consisted of 52,471 adult admissions, with 630 diagnosed retrospectively with CAP at admission.

Figure 3 shows that 283 (44.9%) of the CAP admissions were reviewed by the SPIN team. Of these it was possible to compare the CURB65 derived severity determined by EASUL and the SPIN team for 227 (80.2%). Only 112 (49.4%) had the same severity determined by both EASUL and the SPIN team, which meant that over half did not match, with 57 (25.1%) of reviewed admissions identified by EASUL as high severity but recorded only as low or moderate severity by the SPIN team. Conversely, 5 (2.2%) of the admissions were classified as moderate severity by the SPIN team, but deemed only low severity by EASUL. Crucially for the approach, no patients were classified as high severity by the SPIN team and as low severity by EASUL. To confirm the comparison was correct, the raw input data from a proportion of the patients who were classified differently by EASUL and the SPIN team was examined manually by the main author to ensure the algorithm was working correctly and potential reasons for the observed differences were discussed with clinical staff in the team. Based on this it appeared that some of the patients had indeed been incorrectly scored by the SPIN team. There are several reasons why this could be the case including data missing/missed during the admission; results arriving or being recorded later due to clinical pressures; short lower/higher severity bursts calculated by EASUL but missed / ignored for clinical reasons by clinical staff; or patients being





discharged prior to review. Additionally, exactly how and when CURB65 is applied is not always systematic and can be based on clinical judgement. For example, it may be calculated and recorded before a patient deteriorates and severity increases. Finally, it is also important to note that 347 (55.1%) of the CAP admissions were not reviewed by the SPIN team, with 244 (70.3%) of these deemed to have had moderate or severe disease.

### 3.2. Embedding decision support into a clinical tool

In addition to showing how it could improve prioritisation of CAP admissions, we also wanted to demonstrate how our framework could be used to develop embeddable CDS tools. **Figure 4** shows a proof-of-concept architecture for achieving this. A key component of this is a “simulated admissions” tool which imitates patient flow by generating near real-time synthetic data. This pushes appropriate simulated data to the Broker at corresponding time points which drives the Handler service, utilising the HTTP Client to progress the patient journey accordingly. When no data is available at a particular step in the plan, or the journey reaches completion (e.g. the patient is discharged or dies) the Handler pauses and either awaits further data or finalises the journey. During the process, EASUL outcome data is sent back to the broker to feed the Receiver service in the Clinical System. This leads to the creation of new or updated clinical records according to where a patient is in their journey. The Clinical Frontend can also access and display data and visualisations from the Client related to specific journeys. The results of this are shown in the screenshots in **Figure 5** which demonstrates a prototype Clinical Frontend based around a prioritisation dashboard with embedded visual elements and results from the EASUL plan for CAP management.

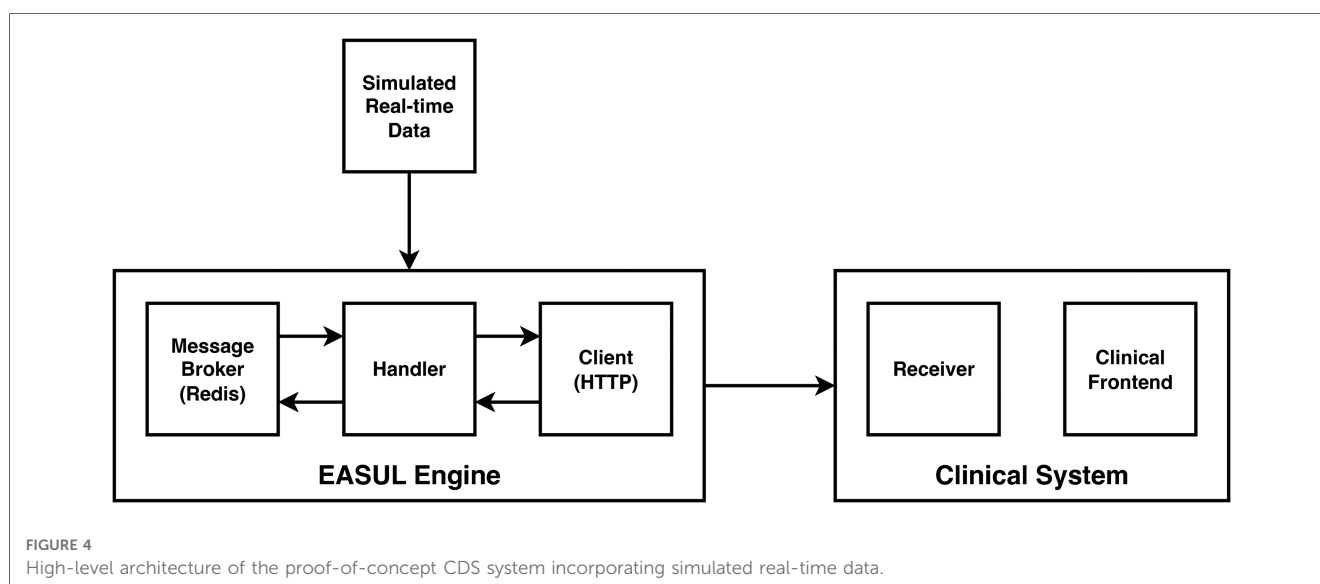
As part of our prototype, we also embedded an ML model which predicted pneumonia outcome (14). This gave us the opportunity to implement visualisations to improve on

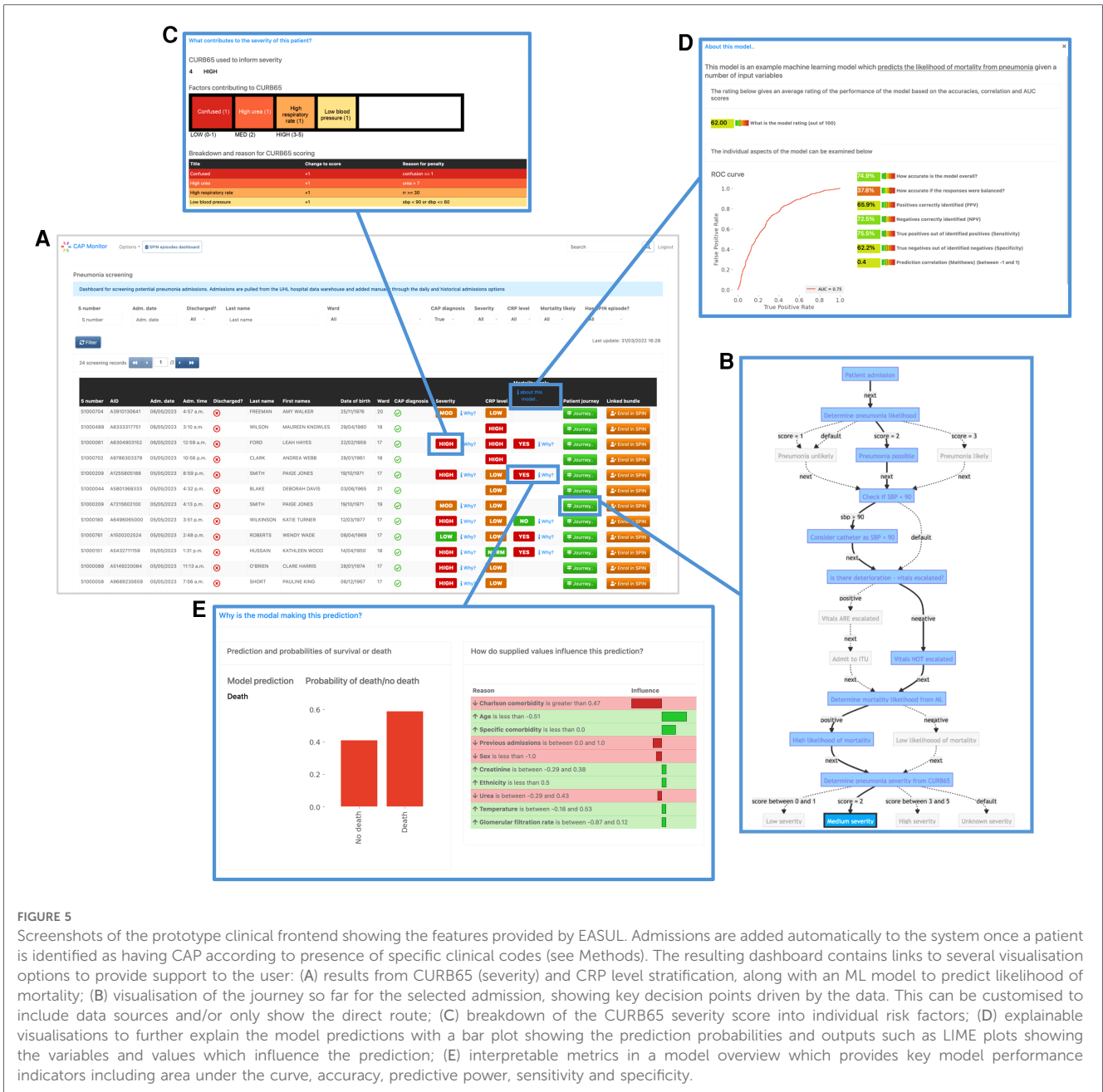
interpretation of specific results output by the model (e.g. LIME-based plots) and present explainable reasons for predictions, along with metrics showing model performance, such as ROC (including area under the curve), accuracy, recall and precision.

## 4. Discussion

We demonstrate here how our approach can effectively be used in operations, research and decision support tool development. Through our QI project we were able to show its utility for prioritising and triaging patients who may not otherwise receive the care required in a timely fashion. As we previously reported (6), patients not reviewed by the SPIN service have notably poorer outcomes, most likely because they are prioritised less effectively. Early identification of the >200 high priority CAP admissions not reviewed by the service suggests that better use of data through EASUL-type approaches would improve this situation, particularly if used as a triage to screen and exclude lower priority patients. Furthermore, in the patients reviewed by the SPIN team there were clear inconsistencies between the severity levels calculated by the SPIN team and those derived through the EASUL-based tool—further strengthening the case for using tools such as EASUL to support the team’s decision making. We believe that tools like the prototype CDS we demonstrate would have significant utility in clinical practice and envisage many situations where they could benefit patient outcome through automation and simplification of straightforward data-driven processes in hospital patient management to identify earlier treatment opportunities, reduce the burden on clinical teams and enable more consistent clinical practice.

Our interest in pneumonia management was driven by a need for equitable access to evidence based interventions across a large multi-site healthcare organisation. Local QI priorities facilitated access to appropriately granular severity data. The very experienced and database enabled SPIN team played a key role in recording and promoting the quality of this dataset as relevant





**FIGURE 5** Screenshots of the prototype clinical frontend showing the features provided by EASUL. Admissions are added automatically to the system once a patient is identified as having CAP according to presence of specific clinical codes (see Methods). The resulting dashboard contains links to several visualisation options to provide support to the user: (A) results from CURB65 (severity) and CRP level stratification, along with an ML model to predict likelihood of mortality; (B) visualisation of the journey so far for the selected admission, showing key decision points driven by the data. This can be customised to include data sources and/or only show the direct route; (C) breakdown of the CURB65 severity score into individual risk factors; (D) explainable visualisations to further explain the model predictions with a bar plot showing the prediction probabilities and outputs such as LIME plots showing the variables and values which influence the prediction; (E) interpretable metrics in a model overview which provides key model performance indicators including area under the curve, accuracy, predictive power, sensitivity and specificity.

to outcome prediction. Additionally, the team’s collection of secondary qualitative data facilitated this comparative quality improvement project. Cognisant of more advanced pneumonia-focussed decision support tools in active use (16) our main objective here was to develop and evaluate a generic tool which could be utilised in similar acute hospital trusts, and we are not comparing our CAP solution with more fully featured single-purpose CDS tools. While we considered using an existing freely available dataset to demonstrate our concept, we were keen to apply the tool to a local real-life acute respiratory emergency care scenario and data set rather than a hypothetical one, thereby accepting some additional complexities and limitations.

Existing generic approaches for implementing web-based interfaces for statistical models and processes (e.g. RShiny) are

not focussed on data and system integration and generally entail entering input data into a user interface. While our project does have some similarities with other projects, there are key differences. Leiner et. al describe a “vendor-neutral” AI infrastructure although their solution is specifically targeted towards imaging (17); while the creation and deployment of AI-based models as CDS is a key focus of the KETOS platform (18). Another similar project identified was DEPLOYR (19) which deals more with deployment and continuous monitoring of researcher-created ML models—with a key focus on a specific EPR system; while EASUL focuses more on combining ML with workflows, appears more data source agnostic and also includes UI visualisations. There have also been other projects which focus on knowledge-based CDS run as microservices (18) and

semi-automation of multi-modal AI model development (20). However, unlike EASUL these do not appear to include workflow-driven approaches and customisable visualisations or support different modalities of source data.

Our proof-of-concept clinical system allowed us to demonstrate how algorithms of varying complexities could be embedded into an information system in a re-usable way. Using EASUL we were able to include both simple risk scores and a pre-existing ML model in a real-time data-driven workflow and present it to the end-user. This is significant since ML models can be effective, but are not always simple to integrate into such workflows. Additionally, we included support for both model-based and patient-based visual elements to help with explainability, interpretability and accountability. By supporting these types of visualisations we hope to increase levels of trust in black-box ML models, by providing users with more transparency—how and why models are making predictions—and help to simplify the deployment of CDS tools containing ML models with these features. There is a growing interest in this field and we agree with others (2, 21) that transparency and accountability around both the strengths and weaknesses of ML and AI is crucial for gaining the trust of clinicians when using CDS based on these models, particularly given the vast increase in the number of published predictive models (22) and continuing advances in the field. However, we acknowledge that this must be done carefully to ensure that increased transparency does not compromise the security of models/algorithms and lead to potential increases in bias (23).

The flexible nature of our approach means it can be extended to support different data modalities (e.g. omics data, imaging); adaptive workflows including multi-tiered/decision models and potentially mobile applications. Additionally, it could also be used to support patient directed healthcare actions, such as remote monitoring. From a technical perspective our use of generic plans makes it straightforward to modify the approach and allows parts of plans to be re-used, while other aspects can be customised as needed in the same project (e.g. for separate evaluation and implementation). The framework also includes support for storage and re-use of previously trained models, with input data validation based on a data schema and different extraction-transform processes, including the ability to collate different data sources together and to have custom extraction-transform processes applied at the point records are received. While this has some similarities to approaches such as FIDDLE (24), our method is data source and format agnostic. However, we recognise its current lack of support for standards such as Health Level Seven and Fast Healthcare Interoperability Resources (25) is a limitation and are keen to add this due to a growing interest in the UK NHS and elsewhere (26, 27).

The work presented here is preliminary research to establish the potential of our approach and we fully acknowledge that further development would be required to improve security, resilience, and scalability. This also includes an important but often overlooked aspect of CDS - handling missing data—in terms of both determining optimal imputation methods and also in terms of real-time practicalities. For instance, if all but one

variable is available to run a predictive model, we would need to determine how to handle the situation by waiting a certain amount of time before imputing data; using previous data; and/or triggering a notification that a test has been missed. Despite these limitations, we believe that EASUL and similar approaches are important steps in making better use of health data from multiple sources and would help to strengthen trust and accountability in complex ML and AI model enabled CDS. Further research and development in this area is required to evaluate and utilise the available data and models being developed in clinical practice and produce digital health tools which are fit for use in clinical settings.

## Data availability statement

The original contributions presented in the study are included in the article/Supplementary Materials, further inquiries can be directed to the corresponding author/s.

## Ethics statement

The study involving use of patient data was approved by University of Leicester Medicine and Biological Sciences Research Ethics Committee. The studies were conducted in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required from the participants or the participants' legal guardians/next of kin in accordance with the national legislation and institutional requirements.

## Author contributions

RCF conceived the project, developed the framework, undertook the analysis and wrote the first draft of the manuscript. GW and PH contributed to conception and design of the evaluation. DL developed the machine learning model used in the prototype CDS. JS and LS led and undertook data acquisition as part of the SPIN team. DL and MS provided intellectual contributions. All authors contributed to manuscript revision, read, and approved the submitted version.

## Funding

This research was supported by the NIHR Leicester Biomedical Research Centre.

## Acknowledgments

The authors would like to thank the SPIN team, past and present for their assistance during the project and Mei-Mei Cheung and other members of the Research and Innovation



team at the University Hospitals of Leicester NHS Trust for providing data exports for the evaluation.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

- Warner HR, Toronto AF, Veasey LG, Stephenson R. A mathematical approach to medical diagnosis: application to congenital heart disease. *JAMA*. (1961) 177:177–83. doi: 10.1001/jama.1961.03040290005002
- Sutton RT, Pincock D, Baumgart DC, Sadowski DC, Fedorak RN, Kroeker KI. An overview of clinical decision support systems: benefits, risks, strategies for success. *npj Digit Med*. (2020) 3:1–10. doi: 10.1038/s41746-020-0221-y
- Berner ES, La Lande TJ. Overview of clinical decision support systems. In: Berner ES, editor. *Clinical decision support systems: theory, practice (health informatics)*. New York, NY: Springer (2007). p. 3–22. Available from: [https://doi.org/10.1007/978-0-387-38319-4\\_1](https://doi.org/10.1007/978-0-387-38319-4_1)
- Bishop CM. *Pattern recognition, machine learning*. Information Science and Statistics. New York: Springer (2006).
- Rasheed K, Qayyum A, Ghaly M, Al-Fuqaha A, Razi A, Qadir J. Explainable, trustworthy, and ethical machine learning for healthcare: a survey. *Comput Biol Med*. (2022) 149:106043. doi: 10.1016/j.compbiomed.2022.106043
- Free RC, Richardson M, Pillay C, Hawkes K, Skeemer J, Broughton R, et al. Specialist pneumonia intervention nurse service improves pneumonia care and outcome. *BMJ Open Respir Res*. (2021) 8:e000863. doi: 10.1136/bmjresp-2020-000863
- [Dataset] mermaid - Markdownish syntax for generating flowcharts, sequence diagrams, class diagrams, Gantt charts and Git graphs (2023). Available from: <https://mermaid.js.org/> (Accessed May 17, 2023).
- Hunter JD. Matplotlib: a 2D graphics environment. *Comput Sci Eng*. (2007) 9:90–5. doi: 10.1109/MCSE.2007.55
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. (2011) 12:2825–30. doi: 10.5555/1953048.2078195
- Ribeiro MT, Singh S, Guestrin C. “Why should I trust you?”: explaining the predictions of any classifier (2016). Available from: <https://doi.org/10.48550/arXiv.1602.04938>
- Lim W, van der Eerden MM, Laing R, Boersma W, Karalus N, Town G, et al. Defining community acquired pneumonia severity on presentation to hospital: an international derivation, validation study. *Thorax*. (2003) 58:377–82. doi: 10.1136/thorax.58.5.377
- [Dataset] BTS guidelines for the management of community acquired pneumonia in adults: update 2009 | British Thoracic Society | better lung health for all (2018). Available from: <https://www.britthoracic.org.uk/quality-improvement/guidelines/pneumonia-adults/> (Accessed May 17, 2023).
- [Dataset] Free RC, Ali A (2021). . Headfake: a Python-based tool for generating fake data. Available from: <https://doi.org/10.5281/zenodo.4890383>. Language: English.
- Lozano-Rojas D, Mcewan AA, Woltmann G, Free RC. Machine learning for mortality prediction in community-acquired pneumonia emergency admissions. *Eur Respir J*. (2022) 60(Suppl 66). doi: 10.1183/13993003.congress-2022.1419

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery, Data Mining*. Association for Computing Machinery (2016). p. 785–94. Available from: <https://doi.org/10.1145/2939672.2939785>. ArXiv:1603.02754 [cs]
- Dean NC, Jones BE, Jones JP, Ferraro JP, Post HB, Aronsky D, et al. Impact of an electronic clinical decision support tool for emergency department patients with pneumonia. *Ann Emerg Med*. (2015) 66:511–20. doi: 10.1016/j.annemergmed.2015.02.003
- Leiner T, Bennink E, Mol CP, Kuijff HJ, Veldhuis WB. Bringing AI to the clinic: blueprint for a vendor-neutral AI deployment infrastructure. *Insights Imaging*. (2021) 12:11. doi: 10.1186/s13244-020-00931-1
- Gruendner J, Schwachhofer T, Sippl P, Wolf N, Erpenbeck M, Gulden C, et al. KETOS: clinical decision support, machine learning as a service—a training and deployment platform based on Docker, OMOP-CDM, and FHIR Web Services. *PLoS ONE*. (2019) 14:e0223010. doi: 10.1371/journal.pone.0223010
- Corbin CK, Maclay R, Acharya A, Mony S, Punnathanam S, Thapa R, et al. DEPLOYR: a technical framework for deploying custom real-time machine learning models into the electronic medical record. *J Am Med Inform Assoc*. (2023) 30(9):1532–42. doi: 10.1093/jamia/ocad114
- Semenov I, Osenev R, Gerasimov S, Kopanitsa G, Denisov D, Andreychuk Y. Experience in developing an FHIR medical data management platform to provide clinical decision support. *Int J Environ Res Public Health*. (2019) 17:E73. doi: 10.3390/ijerph17010073
- Watson DS, Krutzinna J, Bruce IN, Griffiths CE, McInnes IB, Barnes MR, et al. Clinical applications of machine learning algorithms: beyond the black box. *BMJ*. (2019) 364:l886. doi: 10.1136/bmj.l886
- Meskó B, Görög M. A short guide for medical professionals in the era of artificial intelligence. *npj Digit Med*. (2020) 3:1–8. doi: 10.1038/s41746-020-00333-z
- Zhou Y, Kantarcioglu M. On transparency of machine learning models: a position paper. In: *AI for Social Good Workshop*. Harvard University, Center for Research on Computation and Society (2020).
- Tang S, Davarmanesh P, Song Y, Koutra D, Sjoding MW, Wiens J. Democratizing EHR analyses with FIDDLE: a flexible data-driven preprocessing pipeline for structured clinical data. *J Am Med Inform Assoc*. (2020) 27:1921–34. doi: 10.1093/jamia/ocaa139
- [Dataset] HL7 FHIR foundation: enabling health interoperability through FHIR (2022). Available from: <https://www.fhir.org/> (Accessed May 17, 2023).
- [Dataset] FHIR UK core (2022). Available from: <https://digital.nhs.uk/services/fhir-uk-core> (Accessed May 17, 2023).
- [Dataset] New CMS interoperability rule would streamline prior authorization processes (2020). Available from: <https://www.healthcareitnews.com/news/new-cms-interoperability-rule-would-streamline-prior-authorization-processes> (Accessed May 17, 2023).