A Study into Improving Prediction for the Early Diagnosis of Septic Patients

Sam Barnes-Thornton
Department of Computer Science
University of Bristol
Bristol, UK
tk19310@bristol.ac.uk

Joel Grimmer
Department of Computer Science
University of Bristol
Bristol, UK
ki19061@bristol.ac.uk

Theo Phillips
Department of Engineering Mathematics
University of Bristol
Bristol, UK
em18029@bristol.ac.uk

Finn McFall
Department of Engineering Mathematics
University of Bristol
Bristol, UK
vw19407@bristol.ac.uk

Callum Paton
Department of Engineering Mathematics
University of Bristol
Bristol, UK
eg19901@bristol.ac.uk

***Abstract*—This report focuses on aiding the early hour-by-hour prediction of sepsis, returning evidence-based estimations that can act as a guide to medical professionals. Using clinical health data, three models will be trained and tested. Accompanying the models, a dashboard is designed and produced where medical professionals can select data for specific patients. It identifies the closest patients to them and whether they developed sepsis, as well as displaying the final model prediction.**

***Index Terms*—sepsis, lstm, ensemble, dashboard, prediction**

I. Introduction

A. Motivation

Sepsis is a life-threatening condition that occurs when the body's extreme response to an infection leads to tissue damage or organ failure [1]. In 2017, there were an estimated 49 million cases of sepsis worldwide, resulting in 11 deaths [2]. In that year, almost half of the cases occurred in children, with 2.9 million deaths in children under five years old [2]. There also exists a major regional disparity in the incidence of sepsis. According to the World Heath Organisation (WHO), 85% of cases and deaths occur in low and middle income countries [2]. Sepsis is also extremely prevalent in healthcare facilities, in the US more than 1 in 3 people who die in hospital have sepsis [3]. The cost of managing sepsis is huge, and exceeds that of any other condition, taking up 13% of healthcare expenses in the US and costing $24 billion annually [4]. It is a major global heath issue responsible for considerable healthcare expenses, morbidity and mortality.

B. Sepsis Diagnosis

A key element of sepsis prevention is early diagnosis. Once the condition progresses to septic shock, the most severe form of sepsis, there is an increase in mortality of between 3.9-9.9% every hour that treatment is delayed [5]. However, diagnosing sepsis is challenging. The condition is syndromic, meaning it does not have a single cause but rather a collection of symptoms and indicators that are correlated with it. These symptoms can also vary widely depending on the site of infection, the patients immune response, and other factors. Healthcare professionals use multiple scoring criteria, such as SOFA, qSOFA, and APACHEII, to identify sepsis, which combine physiological factors such as blood pressure, heart rate, respiratory rate and temperature as well as laboratory values such as pH or potassium levels to indicate organ failure. Nevertheless, these measures often are only useful in the latter stages of sepsis. A report by the Centres for Disease Control and Prevention suggests that sepsis, or the infection causing it, starts prior to a patient being admitted to hospital in around 87% of cases [3]. As such there is a desire for early and reliable detection of sepsis.

C. Aims and objectives

Machine learning is a powerful tool for pattern recognition, making it well-suited for detecting sepsis from large volumes of clinical data. In 2019, a worldwide competition hosted by PhysioNet challenged participants to develop an automated system for early sepsis detection in ICU patients [6]. The competition provided data from two US hospital systems that contained vitals reading, patient descriptors, and lab values for 40,336 patients. The data is binned into hours with each hour given a binary label indicating if the patient was septic.

In this report, we use the open source data to develop both a deep learning Long Short-Term Memory (LSTM) neural network and ensemble models for early prediction of sepsis. Rather than simply addressing the problem posed by the PhysioNet/Computing in Cardiology Challenge, we will take a comprehensive data science approach, considering factors such as data privacy and ethics, data processing, model development, and analysis. A focus is given to front end visualisation of results, including the development of a dashboard for medical professionals to track the likelihood of sepsis in their patients.

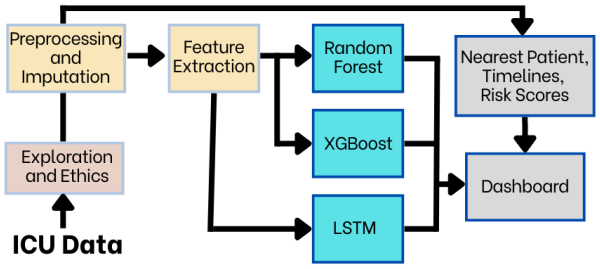

Fig. 1: Project flow chart

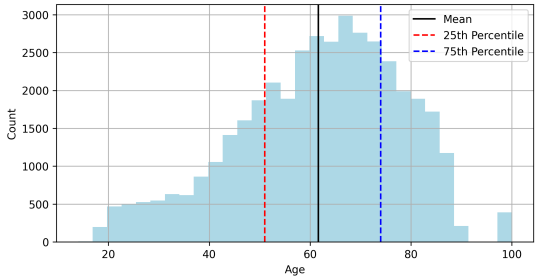

Fig. 2: The distribution of ages of patients in the dataset.

II. Data

A. Ethics and Privacy

The ethical and social implications of using patient data must be carefully considered. Many health care datasets are private due to holding sensitive personal information about patients. For this reason direct identifiers, such as patient name, and confidential information have already been removed. However, two quasi-identifying features, age and gender, remain. In this case, there is not enough information to reach unambiguous identification. Therefore, in spite of the fact that the dataset contains confidential attributes, there is sufficient privacy to protect the identity of entrants.

The use of new technology such as predictors and artificial intelligence in healthcare raises concerns about the possibility of a new source of inaccuracy and data breach [7]. Healthcare is a high-risk area; mistakes can have severe consequences for patient outcomes and there are concerns of unethical behaviour exhibited by these techniques. As such, the techniques used in this report are aimed to be used in conjunction with medical professionals, rather than as a replacement. Instead, our aim is to offer evidence-based estimations that can act as a guide to clinicians.

B. Data Structure

The data from each of the two hospital systems is stored in two folders, with one file per patient. It is tabular, with every row showing the binned readings for an hour. In the case where multiple readings are taken in an hour, the median value is used. The dataset consists of 40 features, including eight vitals, 26 laboratory variables and six demographic variables. The six demographic variables are age, gender, hours between hospital admission and ICU admission, hours since ICU admission and 2 variables for the type of ICU.

Regarding the patient population, 55.9% are male and 44.1% female, with an average age of 61.64 years. A distribution of ages is seen in Fig. 2. We decided to combine the patient data from each hospital system by introducing a patient ID column. Across the two datasets, there is a total of 1,552,210 hours of data, and a large variation in the amount of data for each patient, due to the differing length of time that an individual stays in ICU.

Diagnosis of sepsis t_{sepsis} was determined by medical professionals according to the SOFA score. For septic patients, `SepsisLabel` is 1 if $t \geq t_{sepsis} - 6$ or 0 if $t < t_{sepsis} - 6$, for non-septic patients `SepsisLabel` is 0. This offsetting enables models to be built specifically for early hour by hour predictions.

Throughout all the data, just 7.3% of patients are labelled as having sepsis. 15% of these enter ICU with it and the remaining 85% develop it during their stay. Despite this, the number of rows labelled with 1 is just 1.7%, meaning there is a major class imbalance in the dataset.

C. Pre-processing

A major consideration within the data is dealing with missing data. This is common in healthcare data due to various reasons including a lack of staff or equipment to record certain vitals or lab values [8]. Figure 3 illustrates the extent of missing data in one of the two hospital systems through a Data-Density matrix, where black cells indicate the presence of a reading. The lab values are particularly sparse, with 24 of the 26 missing over 90% of values.

Fig. 3: Data-Density matrix showing the sparsity of the data from the first hospital system. White cells are NaN values.

As a criterion for removing a feature from the dataset, we investigated how regularly readings were never recorded for a patient. A feature may be sparse, but if there is at least one value for most patients, it is still insightful. Figure 4 expresses the percentage of all patients who never receive a reading for each of the features. At a threshold of 70%, four readings are removed from the dataset: `TroponinI`, `Fibrinogen`, `Bilburin_direct` and `EtCO2`. A further three features: `Unit1`, `Unit2` and `Unammed: 0`, are removed on the basis that they are not relevant for Sepsis prediction.

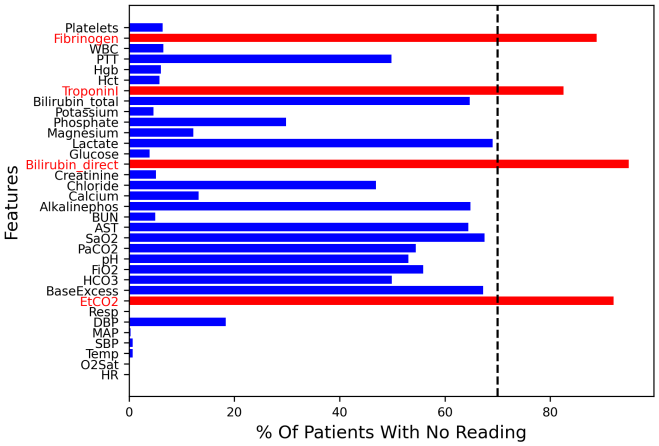

Fig. 4: Plot showing the % of patients missing each of the readings. Those missing in more thab 70% of patients are removed.

Having removed the most sparse features from the dataset, the number of NaN values still remains over 62%. To deal with this, multiple data imputation methods were employed.

D. Imputation

Finding the most efficient way of imputing the missing data was critical to producing the most accurate prediction models possible. The data has to be grouped patient by patient, as it would not be accurate to copy values between patients. This meant for many patients there were features that contained no data at all. From this, two different requirements for imputation emerged. The first was where patients had at least one actual reading in the data, and the second where features contained no data for a patient.

The first requirement had many more possible options for imputation methods compared to the second. Our initial work used forwards and backwards filling. This involved setting all of the missing values before the first reading in a patient's feature to the same as the value of that first reading. The rest of the missing data points were filled with the value of the closest data point before them. In features containing some actual data, all missing values are filled and nothing is synthesised as the values equal those actually measured. The primary issue with this method was that it caused unusual spikes in the data where forward filled values would reach the next actual data point. In order to mitigate this, we trialled linear interpolation instead. This will fill in the missing data points between actual values with a smooth line that connects the two actual values. It eliminates the unusual spikes, producing a more realistic estimation of the data.

To test the filling and interpolation options, we purposefully removed some existing data and kept track of the values and location. We then imputed this new data and compared the imputed values with the actual values that should be in their location by computing the normalised root mean squared error (NRMSE). Several samples of removed data were used to ensure a fair experiment. The results, as shown in Fig. 5, suggest interpolation performs better than forward and backward filling on average over all the features. Any features it performs worst in have a very small variation in the NRMSE. As mentioned before, any patients with features that contain no actual values will still have missing values. Interpolation will also not fill in any data points before the first reading.

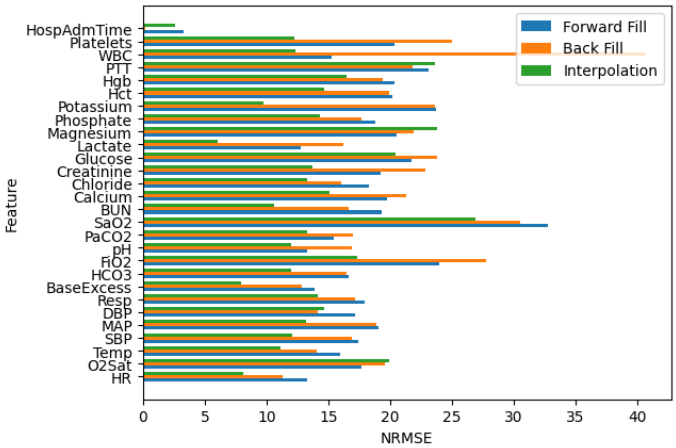

Fig. 5: Results of testing imputation methods for patients with partially filled features

To impute the leftover missing values, we devised two more potential choices. The first was to fill them all with a constant value. This would then not affect the outcome of model predictions but would avoid having to design models that explicitly deal with missing values. This completely avoids synthesising data, but means that we lose the potential for these data points to increase the accuracy of the model. The alternative was to use Multivariate Imputation by Chained Equations (MICE). This is where for each feature with missing values, a conditional distribution is defined for the missing data, given all the other features in the data set [9]. It assumes that data is Missing at Random (MAR), which means that whether a value is missing is determined only by observed values and not by unobserved values [10]. This property is satisfied by the electronic health records in our data. It works by running a series of regression models on each variable in the data. This means that models best fitted to the distributions of individual variables can be chosen, although in our case linear regression will be used for every feature. The basic steps of the algorithm are as follows, taken from a paper aiming to clarify the methods used in MICE [11].

1) A simple imputation, often using the mean value of the feature, is applied to each missing value in the dataset.
2) One feature is chosen, and all the newly imputed values are set back to missing.
3) The observed values in the chosen feature are regressed on all other features in the data. In our case this will mean performing linear regression with the appropriate assumptions.

4) Missing values are then replaced with predictions from the regression model. These imputed values can then be used when producing regression models for other features, along with the observed values.
5) Steps 2-4 are repeated for each feature with missing data until all values are replaced by predictions from regressions, rather than simple imputations. Cycling through all features constitutes one cycle.
6) Multiple cycles are repeated, with the imputations being updated each time.

The number of cycles performed can be changed to suit the use case. In general, the distribution of the parameters that determine the regression models should have converged by the end of the cycles [11].

The reason MICE was chosen for our project is due to the use of all features to impute missing values. Some of the features are very sparse for all patients, even after pre-processing, so using a simple imputation method like the mean of all values would not be reliable. It would also not be accurate as means should not be taken across patients. One negative of the MICE approach is that it doesn't take into account the position of hourly medical records in time. An alternative method, designed specifically for electronic health data, was suggested that finds how similar each of the patient records is to the missing value [12]. It uses this to weight those records in the imputation phase. As a part of the similarity measurements, the records before and after the one in question are also compared which ensures patients are also on the same trajectory of health. We initially wanted to implement this method, as taking into account trajectories of patient health is clearly more accurate. However, the data it was used for was different to the data we were provided with, so we would have had to spend time adapting the method which was not possible in the time-frame. Also, the method was shown to only marginally outperform MICE in the paper [12].

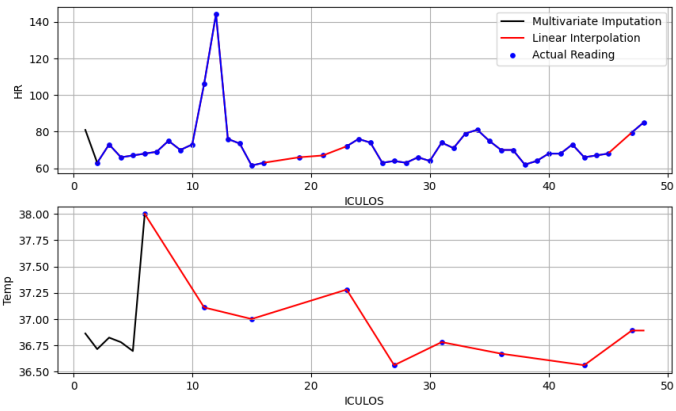

Fig. 6: Demonstration of the imputation methods applied to the heart rate 'HR' and 'Temp' data for a patient.

An example of two features from a patient, produced using linear interpolation followed by MICE, is shown in figure 6. The 'HR' feature contains many observed values, with interpolation used for joining up the slightly more sparse measurements around hour 20. The 'Temp' feature still has many observed values, but they have been taken infrequently. Linear interpolation does a good job here of producing a smooth transition between them. Both features have used multivariate imputation before the first reading, and no erroneous data has been produced from this.

Comparing MICE with constant filling at this stage was not possible as they were not trying to achieve the same end goal. Instead we decided to produce two datasets, one for each method. Any models we produced could then be trained on both sets of data, choosing the one that produced the most accurate predictions.

E. Train, Validation, and Test Sets

Prior to the development of models for early sepsis detection, the data was divided into training, validation and test sets. To construct the training set, data from the first hospital system was used, along with the data from all patients in the second hospital system who are labelled as septic for the entire duration of their time in ICU. This was done to improve the class imbalance during model training. The remaining data from the second hospital system was split into two equal parts to create the validation and test sets. A summary of the data in each of the sets is shown in Table I.

TABLE I: Summary of training, validation and test sets.

	Non-Septic	**Developed Sepsis**	**Admitted With Sepsis**
Train	18,546 (90%)	1,589 (8%)	426 (2%)
Val	9,443 (95%)	445 (5%)	0
Test	9,415 (95%)	474 (5%)	0

III. Model Development

Having pre-processed and imputed the data, models for early prediction of sepsis were developed. Three approaches were considered: two ensemble methods and a Long Short-Term Memory Model (LSTM) neural network.

A. Feature Extraction

A key step in developing accurate models for sepsis prediction is to use feature extraction to draw meaningful information from the existing data. In our study, we extracted three categories of additional features. The first category focuses on the higher prevalence of missing values among non-septic patients. To capture this correlation, we devised a 'time since last reading' feature, which counts the time interval between measurements. For the periods before the first measurement, a constant (-1000) is assigned. This feature was applied to all vital and lab features, adding 30 more columns to the dataset.

Secondly, to capture the time dependency of the data we extract two additional categories of features.

- **Differential Features:** These represent a change in a reading from consecutive time points. As first readings have no previous value to compare against, they are assigned a constant value. This is applied to all vital and lab features, adding an additional 30 columns.
- **Rolling Window Statistics:** For each of the vital and lab readings features, we introduce seven rolling window statistics, these include: $\min$, $\max$, standard deviation, mean and, 0.25, 0.5 and 0.75 percentiles. Windows are 6 hours long and include the current measurement along with the five previous. Imputed constants are treated as NaN values. If all of the 6 previous hours are NaN, the features are assigned a constant value (-1000).

A noticeable characteristic of the dataset is the increased prevalence of missing data amongst non-septic patients. Accordingly, we captured this relationship using two missingness features. The frequency and interval between readings was counted for each feature, producing a further two columns for each existing feature.

The final category of additional features are the empirical risk scores. Abnormalities in measurements are quantified using well-known risk scoring systems. Eight features are scored based on the NEWS [13], qSOFA [14], and SOFA scores [14].

Having extracted the additional features, there are now a total of 316 columns in the data set. However, not all features are required for all models. The LSTM takes the multidimensional time-series data for each patient as an input and as such will not require the differential and rolling window features. It captures the essence of these features inside its architecture, as through the use of feedback loops the recurrent neural network captures time dependencies between features [15].

B. Ensemble Methods

After initial testing of multiple ensemble methods, we discovered that random forest and XGBoost returned the highest F1 scores on the validation set. These models were then developed and optimised for early sepsis prediction.

1) Random Forest: Tree-based methods have been a popular choice for classification tasks in the healthcare industry for several decades due to their reliability and effectiveness [16]. Decision trees, in particular, are well-suited for disease prediction and medical diagnosis as they provide a simple and interpretable representation of decision processes [17]. However, they are susceptible to overfitting and not robust to new data. As such, random forests were designed to overcome these limitations [18]. By leveraging the power of multiple decision trees, they provide a more generalised prediction model [19]. In recent years, random forests have been successfully used for early prediction of changes to heamoglobin in type 2 diabetics [20], forecasting COVID-19 patients outcome [21], and for sepsis detection itself [22].

2) XGBoost: XGBoost is a development of the gradient boosting algorithm, which combines multiple weak predictors to create a strong model, and regularization techniques to handle overfitting-underfitting issues. Within the health science field, XGBoost models have been successfully used for prediction of COVID-19 [23], heart disease [24], and sepsis [25].

A first stage of development for the random forest and XGBoost models was to deal with the class imbalance in the training set. Despite increasing the number of septic patients, the training set was still heavily weighted towards the majority class, meaning the models were strongly biased towards non-septic prediction. Therefore we implemented a unique class balancing method. Specifically, we extracted all data rows that followed the first instance of `SepsisLabel` = 1, along with the 10 hours of data leading up to that point to capture the development of sepsis. Next, for the model to learn the discrepancies between a septic and non-septic patient, we downsampled the data from patients with `SepsisLabel` = 0. This gave a final class balance of 29% septic, 50% 'true' non-septic, and 21% labelled non-septic but in the 10 hours prior to sepsis onset.

For training of the models, all additional features were utilised. Unlike the LSTM model, which takes a patients multidimensional time-series as an input, these models take single hour entries and therefore require the differential and rolling window statistics to capture time dependency. To compare the effectiveness of the imputation methods, models were trained using both multivariate imputation and imputing constants. It followed that for both the XGBoost and random forest models, the multivariate imputation significantly outperformed imputing with constants, achieving a macro F1 score of 0.34 compared to 0.15 with random forest and 0.58 compared to 0.12 with XGBoost. As such, only the multivariate dataset was used in further model development.

To fine tune the hyperparameters of the models we used Bayesian optimisation. This works by converting an objective function to a probabilistic model. The model selects the next set of hyperparameters based on the results of previous evaluations. This process continues until you reach a threshold or hit the maximum defined number of iterations. The U-score on the validation set, described in Section III-D, was used as an objective function as it specifically credits the models ability for early prediction.

C. Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) is an Artificial Neural Network which features feedback connections, allowing the architecture to capture time-series dependencies within the input data [15]. This architecture was specifically designed to avoid the vanishing gradient problems encountered by less state-of-the-art recurrent neural networks, thus becoming a superior evolution of the recurrent neural network [26]. The LSTM's ability to capture the mentioned long-term dependencies through a form of memory is what makes it so

powerful, becoming increasingly more present for medical prediction tasks [27, 28]. Furthermore, there is a large body of research surrounding the early prediction and identification of sepsis which heavily relies on this technology. The following implementation of an LSTM to predict sepsis is inspired by the literature [29].

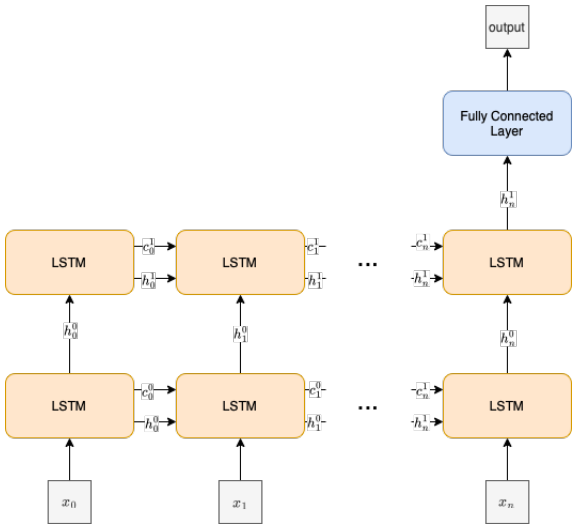

Fig. 7: A schematic diagram of stacked LSTM neural network. The LSTM has inputs x_n, the architecture features two LSTM layers, a fully connected layer and an output layer.

1) Model Architecture: After testing multiple model architectures, the two most successful architectures are presented. Both predict whether a patient has sepsis, using data from the last n hours as an input. The first, more basic model, uses multiple stacked LSTM layers. Each layer contains a hidden state comprised of 20 dimensions with a dropout rate between LSTM cells of 0.2. The final hidden state is fed into a fully connected layer which, when applied to a sigmoid function, produces a probability that the patient has contracted sepsis.

The more complex model, based on an existing solution, adds a convolution layer before the stacked LSTM layers [26]. The kernel width is equal to the number of features per snapshot, so the filter does not convolve within the readings of a given snapshot. This is because separate health readings do not have relationships with each other that can easily be expressed spatially. The filter has a custom height which dictates how many consecutive time series readings are applied within the convolution, with the aim of capturing time series differences within a convolution over this data. The convolutional layer produces 40 features, which are then given to the stacked LSTM layer. Limiting the hidden layers size and the number of LSTM layers can help to prevent over-fitting and improve accuracy.

2) Training: The model was trained over 50 epochs using a mini-batch size of 1024. We have compared the effectiveness of the Adam and RMSProp optimisation algorithms.

Comparing two different optimisation algorithms, we find that Adam extends stochastic gradient descent by using adaptive per-parameter learning rates, minimising the need for fine-tuning of hyper-parameters. RMSProp, short for Root Mean Square Propagation, favours fast convergence to a loss minima by using a decaying average of network gradients to determine the size of the step for each parameter. The algorithm can therefore forget early gradients and instead focus on the most recent partial gradients. Training with Adam produces a train ROC AUC score of 0.696 after five epochs and 0.801 after 25 epochs, while training with RMSProp produces a train ROC AUC score of 0.676 after five epochs and 0.753 after 25 epochs, so while both strategies displayed similar performance Adam was chosen.

During model training, model loss is measured by Binary Cross Entropy (BCE). BCE is well suited for binary classification tasks such as sepsis prediction. Using BCE with Logits Loss as the loss function allows for the use of the log-sum-exp trick, enhancing numerical stability.

D. U-score

For tuning hyperparameters and evaluating model performance, we use the so called U-score. It is a novel metric created by PhysioNet that specifically credits early prediction [6].

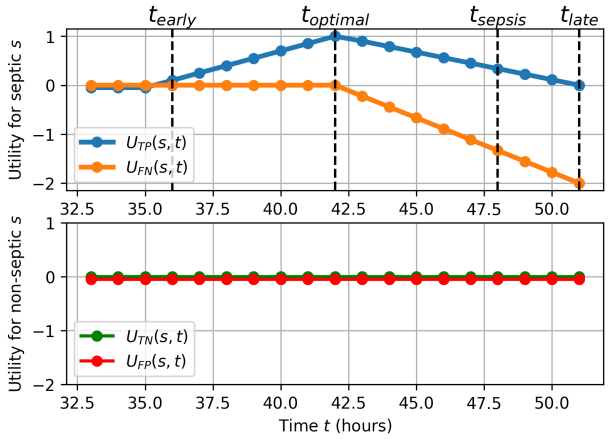

Fig. 8: Demonstration of the utility functions used for computing the U-score. Taken from [6].

Each patient s is assigned a U-score made up of scores $U(s,t)$ for every t, given by

$$U(s,t) = \begin{cases} U_{TP}(s,t) & \textit{septic \& predicted septic,} \\ U_{FP}(s,t) & \textit{non-septic but predicted septic,} \\ U_{FN}(s,t) & \textit{septic but predicted non-septic,} \\ U_{TN}(s,t) & \textit{non-septic \& predicted non-septic.} \end{cases} \quad (1)$$

The functions $U_{TP}(s,t)$, $U_{FP}(s,t)$, $U_{FN}(s,t)$, and $U_{TN}(s,t)$ are demonstrated in Fig. 8. The figure shows how sepsis predictions at least 12 hours before the time of medical diagnosis t_{sepsis} are rewarded, with a maximum reward of $U_{TP} = 1$ coming 6 hours before t_{sepsis} at $t_{optimal}$. Any prediction of sepsis earlier than 12 hours before onset is

slightly penalised. Failure to predict sepsis is also punished, with the penalty increasing to a maximum of $U_{FN} = -2$ at t_{late}, 3 hours after medical diagnosis. For non-septic patients, there is a false positive penalty of $U_{FP} = -0.05$, but no reward for a true negative prediction.

$$U_{total} = \sum_{s \in S} \sum_{t \in T(s)} U(s,t), \tag{2}$$

The total U-score is given in equation 2 where S is the set of all patients and $T(s)$ is the whole time window for each patient s. This is normalised according to $U_{norm} = (U_{total} - U_{no.pred})/(U_{optimum} - U_{no.pred})$, with $U_{no.pred}$ being the U-score associated with assuming no sepsis for all time, and $U_{optimum}$ being the best possible predictions.

IV. Results

Having developed and fine tuned the models, the next task is to evaluate their performance on the test set. The respective results are displayed in Table II.

TABLE II: Comparison of the test set results for the two models.

	U-score	AU-ROC	F1 (macro)
Random Forest	0.312	0.828	0.547
XGBoost	0.376	0.854	0.528
Convolutional LSTM	-0.073	0.619	0.464

The Receiver Operating Characteristic (ROC) curve is a commonly used representation of a binary classifiers performance. The area under ROC curve (AU-ROC) is a way to transform the representation to a quantitative measure. Perfect classifiers will have an AU-ROC of 1, given by a line that passes through the top left corner of the plot, and a random classifier will give a diagonal line and an AU-ROC of 0.5. Figure 9 shows the ROC curve for the three models. The ensemble methods significantly outperformed the LSTM, which achieved a AU-ROC score of just 0.62. XGBoost was marginally superior to random forest with an AU-ROC of 0.85 and 0.83 respectively.

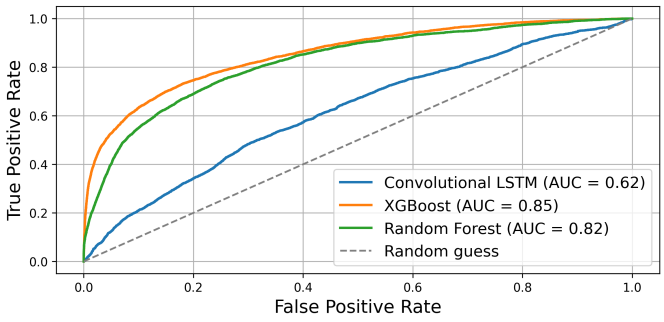

Fig. 9: ROC curves for three models

The most significant metric in evaluating the models is the normalised U-score, as it is specifically designed for crediting early prediction. On this metric, XGBoost outperformed the other models, achieving a score of 0.372, very close to that of the competition winners [22]. The random forest scored slightly less at 0.312, but was significantly better than the LSTM which only managed -0.07.

To further evaluate the models ability for early prediction, we can observe how the F1 score changes in the hours leading up to and after sepsis diagnosis at t_{sepsis}, as shown in Fig .10. As expected, the two ensemble methods had higher accuracy as sepsis was more developed, with XGBoost and random forest achieving a maximum F1 score of 0.86 and 0.81 respectively at $t_{sepsis} + 3$ hours. XGBoost also outperformed random forest in the hours prior to sepsis diagnosis, scoring 0.62 at $t_{optimal}$ compared with just 0.43 for the random forest model. The F1 score for the LSTM did not improve as sepsis developed, but instead plateaued at an F1 score of around 0.35.

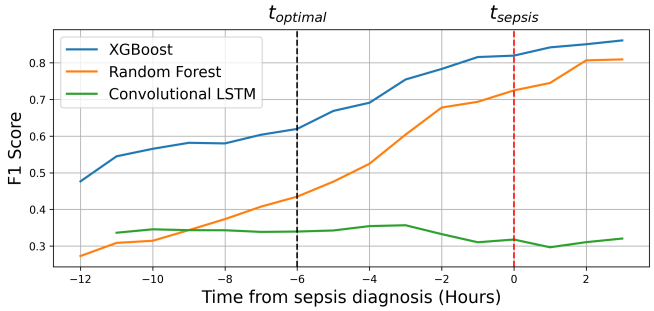

Fig. 10: F1 score of the models for sepsis prediction in the hours prior to and after sepsis diagnosis at t_{sepsis}.

V. Dashboard

To fulfil the end goal of this project, it was important for us to develop a clear way of visualising the results of our research. The solution to this was to create a dashboard that could be utilised by medical professionals to help them when diagnosing patients. Before doing this, we surveyed current thinking about data visualisation dashboards used in the medical industry. The key finding is that dashboards should aim to limit cognitive overload as much as possible. Specifically in intensive care units, doctors have a huge amount to cope with due to the constant care required by patients. Research shows that poorly designed bedside data visualisation devices are the greatest contributor to their cognitive load [30]. One solution called MIVA was developed with the primary aim of preventing cognitive overload [31]. It makes use of temporal organisation in the arrangement of the dashboard, including a timeline to provide perspective from the current readings. It also utilises colour extensively. Using this to code graphical displays can "facilitate rapid interpretation" of the data with red, orange and green representing alarm, warning and safety respectively [32].

Our chosen solution utilises the Dash framework for Python, which facilitates the building of visualisation web applications. Plotly can be used to generate a range of charts that are updated by callbacks from interactive elements in the application. Our dashboard focuses on patient-by-patient analysis with a dropdown selection box controlling the displayed data.

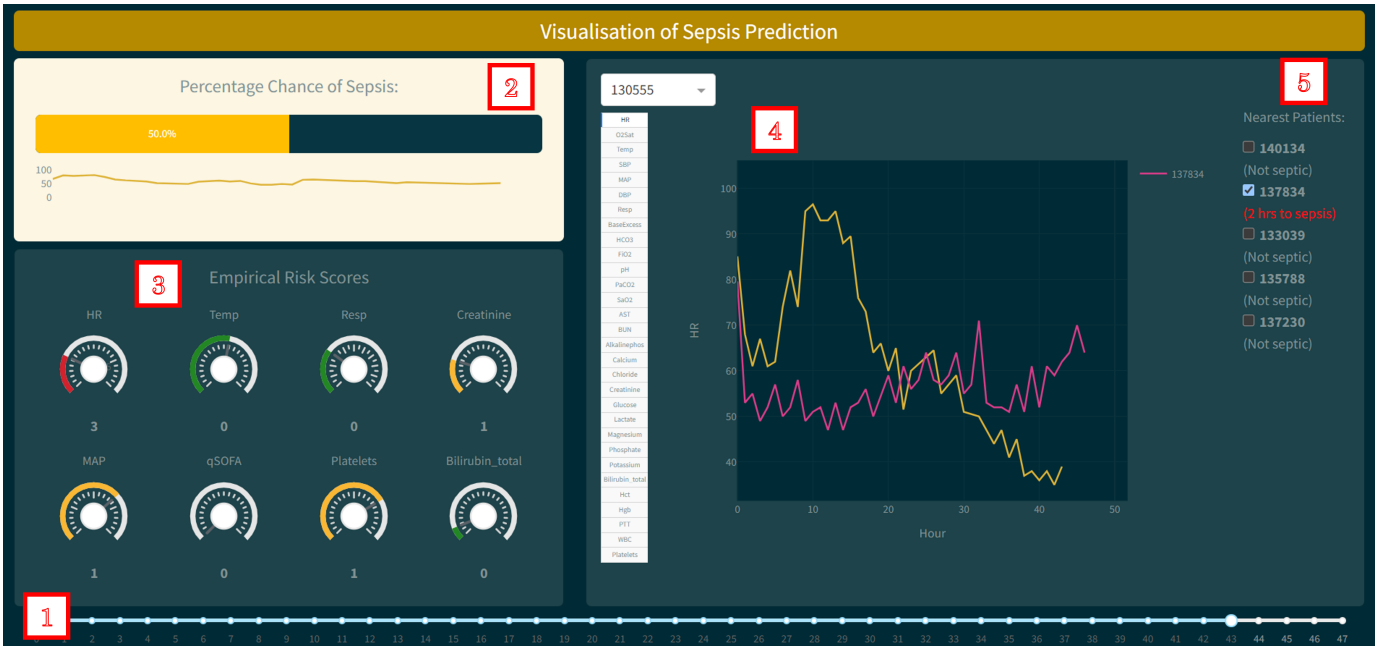

Fig. 11: Dashboard with annotations corresponding to the list of widgets

This updates a number of widgets around the dashboard, summarised as follows:

1) **Timeline:** A Dash 'Slider' at the bottom of the page allows the selection of data for each of the hours a patient is in ICU. It will resize depending on the maximum number of hours for a chosen patient. Toggling between hours will change the data on the dashboard to all points up to the selected hour.
2) **Model Predictions:** A Dash 'Progress' bar and bare-bones line chart are used to display the probability a patient has sepsis at the chosen timestamp, as produced by our model. The bar will fill and gradually change from green to red as the probability increases. The line chart shows the trend of the model output over time so that patient history can be determined more easily.
3) **Empirical Risk Scores:** A collection of gauges that visualise the empirical risk scores of eight features important for recognising sepsis. These are metrics that doctors would take and recognise without access to our model, so the idea of including them is that they can be compared with our predictions to increase trust in the accuracy of the tool.
4) **Feature Plots:** A large Plotly chart that presents each feature in our processed and imputed dataset. They are plotted on a line chart, with tabs used to select the desired feature. The axes will automatically scale depending on the selected feature, with a ten per cent margin above and below the minimum value for both x and y.
5) **Nearest Patients:** The ability to compare the selected patient with others that are similar is integrated into the features plots. A series of checkboxes exhibit five options. They are found using the Hausdorff distance which is a measure of distance between two point sets, corresponding to the sets of hourly readings for two patients in this dataset [33]. Selecting a checkbox will plot that patient's data on the line chart as well, but for all their hours, so the user can identify potential future trends. Coupled with each patient listed is a metric determining whether they ever had sepsis and, if so, how many hours there are until the diagnosis.

Careful decisions were made regarding the design of each of the widgets, along with their placement within the dashboard. The model prediction was presented in as simple a way as possible to provide a quick reference point for doctors. If they are running low on time, it is important for them to be able to quickly identify the key piece of information from the dashboard. The history of the probability score over time was included so that it is clear whether the likelihood has suddenly spiked, or has gradually risen over time. Gauges were chosen to show the empirical risk scores due to their simplicity. The graphic can display the actual value of the feature, and the colour, combined with a value underneath between zero and three, can display the score. This abstraction of the danger into colour and categorical values reduces the cognitive load on the user.

The visualisation on the right of the dashboard is intended to give users the ability to perform some of their own research into the reasons behind the model and risk score predictions. Using tabs rather than a dropdown to select a feature allows users to see more clearly what is available, and toggle quickly between options. The ability to view the most similar patients to the one selected adds another dimension to the dashboard. It enables a user to make their own conclusions about how the

patient's condition might develop by comparing it with others.

During the development of the dashboard, we debated including visualisation of the principal components of the data set. The idea of this would be to show users the features that are most correlated with sepsis, which might be useful when trying to make an informed diagnosis. However, adding another widget would increase the cognitive load on the user significantly so we decided against it. The information could have been included in place of the empirical risk scores, but it is important to keep these to provide doctors with metrics they are already familiar with so they have context. Another potential addition was a visualisation of the decision tree from one of our ensemble models. Unfortunately, any options for this were far too large and complicated to be effective, especially as the primary aim was to reduce cognitive overload.

VI. Discussion

The primary objective of this report was to develop both deep learning and ensemble models for early sepsis prediction. These models could then be integrated into a dashboard to serve as a tool for medical professionals to assess the risk of sepsis.

After careful consideration of how to impute the missing data, extract informative features, and deal with the class imbalance, we were able to build a random forest, XGBoost and LSTM model, optimised with the U-score for early prediction. However, across all methods, accurate prediction of sepsis was challenging to achieve. Despite efforts to deal with the imbalance of classes, models remained strongly biased towards non-septic prediction, with significantly higher F1 scores for `SepsisLabel` = 0, than `SepsisLabel` = 1 across all three models. We attempted to optimise the models specifically to focus on achieving a higher F1 score for `SepsisLabel`=1, but this led to a rise in false positives. Considering the context of the problem, this would be problematic for medical professionals as it would mean there could be a significant waste of resources.

This is also why the U-score was so central to our evaluation processes. Ultimately, we have created models to aid medical professionals in their pursuit to identify the early development of sepsis. The project was not about creating complex models with the highest possible F1 scores, but instead to optimise with a metric, designed by experts in computational physiology, that specifically addresses the needs of medical professionals seeking to improve early sepsis prediction.

The dashboard incorporates multiple methods of evaluating the risk of sepsis, helping a practitioner ultimately use their own judgement as to what is best for an individual patient. It was designed to help bridge the gap between the decision processes of physicians and machine learning models. As such, given the inherent explainability of decision trees that are central to both XGBoost and random forest, these models seem most appropriate for this purpose. Furthermore, they performed significantly better than the LSTM, which itself is a less interpretable classifier.

The LSTM model was in theory an ideal candidate given the structure of the data. However, despite using various techniques such as windowing, which yielded good results in literature, we were unable to replicate them [26]. Although investigations into stacking an LSTM with a convolutional and max pooling layer helped increase prediction accuracy to some extent, ultimately, it struggled to truly differentiate and classify variables. We predict the LSTM is unable to identify the relation between vitals readings and sepsis labels because of a lack of vitals readings post sepsis diagnosis. Future investigation into early sepsis prediction, using this method should investigate interpolating each patient's data over an assumed suitable range to homogenise the number of all patients' readings. However, currently its lack of explainability and poor performance makes it hard to justify its role in this system.

Finally, it is important to discuss the ethical implications of the methods used in this report. A huge proportion of data was imputed, and as such, it is vital that this data is only used for these models and not re-released into the public domain. Imputed data is inherently erroneous and inaccurate in comparison with real data, and if individuals were to use it without knowing its limitations, they may make false and potential harmful conclusions.

VII. Conclusion

In conclusion, this project has revealed the difficulties involved in sepsis prediction. The nature of the condition means there are highly complex relations between variables, and although the ensemble methods were capable of relatively high prediction accuracy at the time of medical diagnosis, early prediction remains a challenging task. However, through thorough data wrangling processes such as imputation and feature extraction, we were able to develop and optimise two ensemble models, who achieved U-scores rivalling that of the highest achievers in the PhysioNet competition. These models were able to be successfully integrated into our dashboard, a front end visualisation tool designed to optimise usability and serve as a tool for medical professionals to make informed decisions on sepsis diagnosis. Further work is required to improve the accuracy of the models, especially the LSTM which despite seemingly being an ideal candidate given the structure of the data, was unable to produce positive results. It is clear machine learning has the potential to be a hugely influential factor in aiding the early diagnosis of sepsis and that research like this into creating tools for medical physicians is of great significance.

VIII. Code

The source code for our work is available in this repository: https://github.com/sbarnesthornton/sepsis-prediction

References

[1] Christopher W Seymour et al. "Assessment of clinical criteria for sepsis: for the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)". In: *Jama* 315.8 (2016), pp. 762–774.

[2] Kristina E Rudd et al. “Global, regional, and national sepsis incidence and mortality, 1990–2017: analysis for the Global Burden of Disease Study”. In: *The Lancet* 395.10219 (2020), pp. 200–211.

[3] Healthcare-associated Infections. *Centers for Disease Control and Prevention National Center for Emerging and Zoonotic Infectious Diseases (NCEZID) Division of Healthcare Quality Promotion (DHQP) Page last updated: November 9, 2011*.

[4] Carly J Paoli et al. “Epidemiology and costs of sepsis in the United States—an analysis based on timing of diagnosis and severity level”. In: *Critical care medicine* 46.12 (2018), p. 1889.

[5] Anand Kumar et al. “Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock”. In: *Critical care medicine* 34.6 (2006), pp. 1589–1596.

[6] Matthew A Reyna et al. “Early prediction of sepsis from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019”. In: *2019 Computing in Cardiology (CinC)*. IEEE. 2019, Page–1.

[7] Nithesh Naik et al. “Legal and ethical consideration in artificial intelligence in Healthcare: Who takes responsibility?” In: *Frontiers* (Feb. 2022). URL: https://www.frontiersin.org/articles/10.3389/fsurg.2022.862322/full.

[8] Shuo Feng, Celestin Hategeka, and Karen Ann Grépin. “Addressing missing values in routine health information system data: an evaluation of imputation methods using data from the Democratic Republic of the Congo during the COVID-19 pandemic”. In: *Population Health Metrics* 19.1 (2021), pp. 1–14.

[9] Stef Van Buuren and Karin Oudshoorn. *Flexible multivariate imputation by MICE*. Leiden: TNO, 1999.

[10] Joseph L. Schafer. “Multiple Imputation in Multivariate Problems When the Imputation and Analysis Models Differ”. In: *Statistica Neerlandica* 57.1 (2003), pp. 19–35. DOI: https://doi.org/10.1111/1467-9574.00218. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9574.00218. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9574.00218.

[11] Melissa J. Azur et al. “Multiple imputation by chained equations: what is it and how does it work?” In: *International Journal of Methods in Psychiatric Research* 20.1 (2011), pp. 40–49. DOI: https://doi.org/10.1002/mpr.329. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mpr.329. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mpr.329.

[12] Ali Jazayeri, Ou Stella Liang, and Christopher C. Yang. “Imputation of missing data in electronic health records based on patients’ similarities”. In: *Journal of Healthcare Informatics Research* 4.3 (2020), pp. 295–307. DOI: 10.1007/s41666-020-00073-5.

[13] RCP. *National Early Warning Score (NEWS) 2*. Dec. 2022. URL: https://www.rcplondon.ac.uk/projects/outputs/national-early-warning-score-news-2.

[14] Paul E Marik and Abdalsamih M Taeb. “SIRS, qSOFA and new sepsis definition”. In: *Journal of thoracic disease* 9.4 (2017), p. 943.

[15] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[16] Vili Podgorelec et al. “Decision trees: an overview and their use in medicine”. In: *Journal of medical systems* 26 (2002), pp. 445–463.

[17] Haewon Byeon. “Development of depression prediction models for caregivers of patients with dementia using decision tree learning algorithm”. In: *International Journal of Gerontology* 13.4 (2019), pp. 314–319.

[18] Haewon Byeon. “Is the random forest algorithm suitable for predicting parkinson’s disease with mild cognitive impairment out of parkinson’s disease with normal cognition?” In: *International journal of environmental research and public health* 17.7 (2020), p. 2594.

[19] Abhishek Sharma. *Random Forest vs decision tree: Which is right for you?* Apr. 2023. URL: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/.

[20] Tadao Ooka et al. “Random forest approach for determining risk prediction and predictive factors of type 2 diabetes: large-scale health check-up data in Japan”. In: *BMJ Nutrition, Prevention & Health* 4.1 (2021), p. 140.

[21] Jie Wang et al. “A descriptive study of random forest algorithm for predicting COVID-19 patients outcome”. In: *PeerJ* 8 (2020), e9945.

[22] Simon Lyra, Steffen Leonhardt, and Christoph Hoog Antink. “Early prediction of sepsis using random forest classification for imbalanced clinical data”. In: *2019 Computing in Cardiology (CinC)*. IEEE. 2019, pp. 1–4.

[23] Zheng-gang Fang et al. “Application of a data-driven XGBoost model for the prediction of COVID-19 in the USA: a time-series study”. In: *BMJ open* 12.7 (2022), e056685.

[24] Kartik Budholiya, Shailendra Kumar Shrivastava, and Vivek Sharma. “An optimized XGBoost based diagnostic system for effective prediction of heart disease”. In: *Journal of King Saud University-Computer and Information Sciences* 34.7 (2022), pp. 4514–4523.

[25] Shuhui Liu et al. “Dynamic sepsis prediction for intensive care unit patients using XGBoost-based model with novel time-dependent features”. In: *IEEE Journal of Biomedical and Health Informatics* 26.8 (2022), pp. 4258–4269.

[26] Chen Lin et al. “Early Diagnosis and Prediction of Sepsis Shock by Combining Static and Dynamic Information Using Convolutional-LSTM”. In: *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. 2018, pp. 219–228. DOI: 10.1109/ICHI.2018.00032.

[27] G Maragatham and Shobana Devi. “LSTM model for prediction of heart failure in big data”. In: *Journal of medical systems* 43 (2019), pp. 1–13.

[28] Bhargava K Reddy and Dursun Delen. “Predicting hospital readmission for lupus patients: An RNN-LSTM-based deep-learning methodology”. In: *Computers in biology and medicine* 101 (2018), pp. 199–209.

[29] Alireza Rafiei et al. “SSP: Early prediction of sepsis using fully connected LSTM-CNN model”. In: *Computers in biology and medicine* 128 (2021), p. 104110.

[30] Anthony J. Faiola, Preethi Srinivas, and Bradley N. Doebbeling. “A ubiquitous situation-aware data visualization dashboard to reduce ICU clinician cognitive load”. In: *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*. 2015, pp. 439–442. DOI: 10.1109/HealthCom.2015.7454540.

[31] Anthony J. Faiola, Preethi Srinivas, and Bradley N. Doebbeling. “A ubiquitous situation-aware data visualization dashboard to reduce ICU clinician cognitive load”. In: *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*. 2015, pp. 439–442. DOI: 10.1109/HealthCom.2015.7454540.

[32] BRYAN A. WILBANKS and PATSY A. LANGFORD. “A review of dashboards for data analytics in nursing”. In: *CIN: Computers, Informatics, Nursing* 32.11 (Nov. 2014), pp. 545–549. DOI: 10.1097/cin.0000000000000106.

[33] Abdel Aziz Taha and Allan Hanbury. “An Efficient Algorithm for Calculating the Exact Hausdorff Distance”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.11 (2015), pp. 2153–2163. DOI: 10.1109/TPAMI.2015.2408351.