Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-17828-y

Unleashing the power of explainable AI: sepsis sentinel's clinical assistant for early sepsis identification

Snehashis Chakraborty[1] · Komal Kumar[1] · Kalyan Tadepalli[2] · Balakrishna Reddy Pailla[3] · Sudipta Roy[1]

Received: 15 May 2023 / Revised: 5 October 2023 / Accepted: 6 December 2023

Abstract
Sepsis is a severe and potentially life-threatening condition that occurs when the body's immune response becomes excessively intense in reaction to an infection. If not promptly treated, it can result in organ failure and even death. So, early identification of patients at risk for sepsis is crucial to improve the patient's outcome in critical care. The main objective of this work is to create a highly accurate model named XAutoNet that utilizes optimal number of clinical features to predict sepsis 6 h before its onset, also providing diagnostic map behind its prediction that will help health workers in better treatment. The importance of this work is heightened in resource-scarce settings, where not all tests are available, or the turnaround time is excessive. A novel convolutional neural network based autoencoder architecture is also implemented to augment the performance of XAutoNet by reducing the input dimensions into an optimal number of dimensions. For explaining the participation of features in feature extraction, Gradient-based Class Activation Map is used to visualize the gradients in individual layers of the encoder block via heatmaps. For the explainability of XAutoNet, a visualization tool named SHapley Additive exPlanations (is used to interpret the features' contribution in the model's global and local prediction. The proposed XAutoNet model has an accuracy of 93%, Precision of 90%, F1 score of 92%, and Recall of 94%. The performance of the convolutional neural network -based autoencoder was also compared with its other variants, including Principal Component Analysis, which showed its high feature extraction power. The XAutoNet has also outperformed other comparable method by a significant margin. The performance of XAutoNet is instrumental in predicting sepsis in advance by understanding the non-linearity and complexity of the data of Intensive Care Unit patients with the help of the proposed autoencoder.

Keywords Precision medicine · Machine learning · Deep learning · Sepsis sentinel · Informative bottleneck · Transparent AI

Extended author information available on the last page of the article

Published online: 16 December 2023

1 Introduction

Sepsis is a life-threatening condition often resulting from an exaggerated body response to an invading pathogen. In 2017, nearly 48.9 million people were affected by sepsis, with a mortality of 11 million [1]. Children under five accounted for almost half of global sepsis cases in 2017, with an estimated 20 million cases and 2.9 million deaths [1]. Sepsis poses a 30% mortality risk, severe sepsis 50%, and septic shock 80%. The high morbidity and mortality of sepsis, especially in the ICU, make it a global public health issue. Early detection and immediate treatment are critical. The mortality rate increases by 4–8% per hour delay in treatment. So, to diagnose and detect sepsis at an early stage, many hospitals use sepsis-related ICU scoring systems or indicators like Systemic Inflammatory Response Syndrome (SIRS) score, Modified Early Warning Score (MEWS), Glasgow Coma Scale (GCS), Quick Sequential Organ Failure Assessment (qSOFA) and many more. Though these scoring systems have a high sensitivity, their specificity is extremely poor due to the high overlap of the symptoms of sepsis with other entities leading to high false alarms.

Artificial Intelligence (AI) can help medical practitioners streamline tasks, improve process efficiency and simplify complicated processes. With the increasing ubiquitousness of Electronic Medical Records (EMR), medical imaging, pathophysiology, and other data, AI can help healthcare experts maximize their efficiency [2]. Framework based on ML can be used to predict sepsis at an early stage with the help of the patient's physiological data which can help the health workers to start treatment as early as possible [3]. Sometimes patient's data can be in unstructured clinical text format which can still be used for getting valuable insights, prediction, early detection, and identification of sepsis [4]. Also, there are AI-based algorithms that consider both structured and unstructured clinical notes to predict and diagnose sepsis [5]. However, previous research indicates several barriers to widespread adoption of sepsis prediction systems. These include the lack of generalizability across institutions, high false alarm rates, and the risk of automation bias due to the models' lack of interpretability. Another factor is the use of excessive features in model building, making them complex and increasing turnaround time, as hospitals may not have all the necessary tests available. Because of these drawbacks in the field of early prediction of sepsis, development of an innovative intelligent AI/ML model is very much required which will utilize an optimal number of features to reduce the false alarm rate as low as possible while attaining a high accuracy and can also provide explanations behind its prediction, thereby building up the trust in the medical practitioners which it lagged. Therefore, this paper focuses on building a predictive model capable of predicting sepsis 6 h in advance, considering the false positive cases that it encounters and providing diagnostic maps that explain its predictions to the practitioners to assist them.

This paper has the following contributions:

- Developed a highly accurate predictive model to identify patients with sepsis at an early stage by taking a minimal number of features as input.
- Considering the complexity of EMR data of ICU patients, a novel convolutional neural network (CNN)based autoencoder is introduced for extracting useful information, which is appropriately utilized by the model, helping it classify better.
- Implementation of proper and detailed data pre-processing to retain important information as it belongs to critical care patients.

- Use of Gradient-based Class Activation Map (GradCAM) for explaining the feature extraction process by Autoencoder and SHapley Additive exPlanations (SHAP) for digging out the hidden reasons behind the model's prediction.

The paper is divided as follows: Sect. 2 describes the literature review, and the complete methodology, from data description to explainable AI methods, is discussed in Sect. 3. Section 4 shows the results and analysis of the methodology section. The comparative study of the proposed architectures with other models is shown in Sect. 5. Section 6 takes us through the explainability of the proposed architectures in a graphical way, followed by Discussion in Sect. 7. Finally, we conclude our paper in Sect. 8.

2 Literature review

Over the last few years, ML has gained significant attention in the field of sepsis prediction [6–10]. Several studies have proposed various ML models for early sepsis detection using patients' clinical data. In one study [3], the authors introduced an ensemble ML model comprising light Gradient Boosting Machine (light GBM), RF, and XGBoost. This ensemble model takes 30 features as input and can predict sepsis 6 h in advance. They demonstrated that their ensemble model outperformed individual models to some extent, achieving a good AUC score. However, the model lacked parameter and structure optimization, and its overall performance was not high enough for clinical deployment. Another investigation [11] evaluated temporal models such as Recurrent Neural Network (RNN) and bidirectional Long-Short-Term Memory (LSTM) networks for sepsis detection and blood culture detection. The models exhibited quick and steady recognition of different sepsis levels through an automated procedure. The short-term classifier showed better capability in identifying the onset of sepsis, while the long-term classifier provided better predictions of future sepsis outbreaks by relying on general signs of the disease. However, their performance suffered in terms of early prediction.

In a different approach, authors in [12] analysed clinical text data from narrative notes provided by healthcare workers to predict sepsis using ML and Natural Language Processing (NLP) techniques. The inclusion of clinical text and structured data improved the model's predictive power and accuracy. However, the sensitivity to detect sepsis in this data varied, as it depended on timestamps that differ between hospitals. In [13], the author developed a classifier that uses patients' EMR, demographics, and vital signs to predict sepsis 6 h in advance. They introduced a new filling algorithm called mixed filling to address missing values in the dataset. Experimenting with various ML models like Logistic Regression (LR), XGBoost, RF, Neural Network, light GBM, and LSTM, they found that light GBM trained on the mixed filling dataset outperformed other models in terms of Receiver Operating Characteristics (ROC) curve. Despite its good predictive power for early sepsis prediction, the use of light GBM was prone to overfitting due to leaf-wise split. Similarly, in [14], the authors proposed an ensemble model comprising RF, LR, Naive Bayes (NB), and Support Vector Machine (SVM) trained on vital signs and clinical laboratory values of ICU patients. The data imputation technique used was average mean, which improved the model's performance. While the proposed model showed efficiency by outperforming individual models, it lacked explainability.

Authors in [15] utilized mean processing and feature generation methods along with ML algorithms like Light GBM and XGBoost to predict sepsis 6 h in advance. The mean

processing method addressed class imbalance, while the feature generation method created 146 additional features. Both algorithms performed well in the feature generation method, with Light GBM selected over XGBoost for its speed and generalization ability. However, the high number of generated features led to the "Curse of Dimensionality," resulting in poorer model performance, increased computational time, and visualization problems [16]. The same issue was seen in [17] where the authors first optimize a total of 1080 features to 660 by concurrent use of a multi-objective genetic algorithm optimization approach and ANN then they train a DL classifier to accurately classify patients with sepsis. In [18], a comparative study was conducted, including a novel genetic algorithm optimized rule-based system and ML algorithms like k-Nearest Neighbour (KNN), LR, SVM, ensemble classifier, and Neural Network. The neural network with 65 hidden neurons performed the best, but physicians preferred the rule-based system due to its interpretability and transparency. However, the rule-based system lacked generalizability as it was trained on a small dataset. To compare the predictive power of features, the authors in [19] trained various ML models like LR, SVM, RF, Adaptive Boosting (AdaBoost), and NB on different subsets of features, including biomarkers, EMR data, and a combination of both. SVM and AdaBoost achieved the highest AUC score, with little difference between the dataset containing both types of features and the one containing only biomarkers. However, the overall performance of their best method was low compared to other studies.

In [20], an RNN model was proposed to predict sepsis onset using the MIMIC-III dataset, and its performance was compared with the InSight algorithm, designed for the general patient population [21]. Though their model outperformed InSight by achieving a better AUROC score for different lengths of look-back, the performance was low compared to other work. Similarly, in [22], the authors developed the Artificial Intelligence Sepsis Expert (AISE) for predicting sepsis 4 to 12 h in advance and identifying factors impacting its prediction. AISE used 65 high-resolution vital signs and EMR data for training and showed good predictive performance over different validation cohorts. However, AISE had a higher false positive rate, making it more sensitive but also prone to false alarms. In [23], the authors proposed the Kernel Extreme Learning Machine (KELM) model trained on data extracted from blood samples of healthy individuals and sepsis patients. They introduced a new learning mechanism and used RF for feature selection, showing promising results in predicting sepsis. However, the model's lack of specificity resulted in high false alarms.

In summary, the aforementioned studies have shown encouraging outcomes in the early detection and differentiation of diverse sepsis levels, achieved through the exploration of various ML algorithms, data imputation techniques, and feature generation approaches (as outlined in Table 1). Nevertheless, significant challenges persist, such as the variability in model decision-making due to the absence of clinically relevant biomarkers [24–27] in the training datasets, data loss owing to missing values, and issues related to class imbalance. Furthermore, these works also grapple with the complexity arising from the adoption of numerous features during training, which contributes to overfitting, prolonged execution times, and reliance on features that might be unavailable during testing, especially in resource-scarce scenarios. Additionally, the "black box" nature of these models presents another hurdle, as medical practitioners often encounter difficulties in comprehending the insights and interpreting the information generated by these opaque systems.

Therefore, to address the issues mentioned above, this research focuses on developing a deep learning model named XAutoNet to predict sepsis 6 h in advance, which takes an optimum number of features for which the false positive rate is minimal, along with other performance metrics. To address the non-linearity of the data, an autoencoder-based

Table 1 The performance of various models in early prediction of sepsis

Method name	Dataset used	Number of features used	Performance	XAI framework used
Ensemble model [10]	M3	30	Acc: 72.7% AUC: 79.2%	-
Bidirectional LSTM [3]	M3 and GUH	9	AUC: 84%	-
Light GBM [12]	M3	38	Acc: 98% F1 Score: 98%	LIME
Ensemble model [13]	SH	13	Acc: 96% F1 Score 98% AUC: 96%	-
Light GBM [14]	M3	146	F1 Score: 76% AUC: 97%	SHAP
DL classifier [16]	M3	660	Acc: 80% Sen: 78% AUC: 80%	-
Neural Network model [17]	DHC	10	Acc: 92% Sen: 92%	-
SVM [18]	CFH	15	AUC: 80%	Model specific
RNN [19]	M3	10	AUC: 81%	-
AISE [21]	M3	65	Acc: 67% AUC: 85%	-
RF-CFOA-KELM [22]	GCMS	5	Acc: 81% Sen:89%	-

dimension reduction is proposed to select this optimum number considering the above metrics. To address the class imbalance problem, a hybrid approach of upsampling followed by downsampling is experimented with to retain as much information as possible. For the interpretation of the Autoencoder, GradCAM is used to visualize each layer of the encoder block, which is responsible for feature extraction, whereas SHAP is used to explain and interpret the predictions made by XAutoNet by analysing the features.

2.1 Problem formulation

The early prediction of sepsis plays a vital role in initiating timely treatment, averting complications, and enhancing patient survival rates by intervening before the condition escalates. While early research achieved impressive results through the utilization of a high number of features, persistent challenges such as elevated false alarms, proper data preprocessing, achieving high performance with minimal features, and the absence of interpretability hinder the practical implementation of these models in real-world scenarios. Our work aims to comprehensively address these challenges by constructing a robust and transparent model capable of deployment in clinical settings. This model holds the potential to aid healthcare practitioners in both the detection and treatment of sepsis patients.

3 Methodology

Figure 1 portrays our comprehensive research workflow, spanning data collection, model prediction, and interpretation. The process commences with collected data passing through a pre-processing module, generating refined data. This refined data serves as the foundation for training an autoencoder, which effectively extracts an optimal number of features.

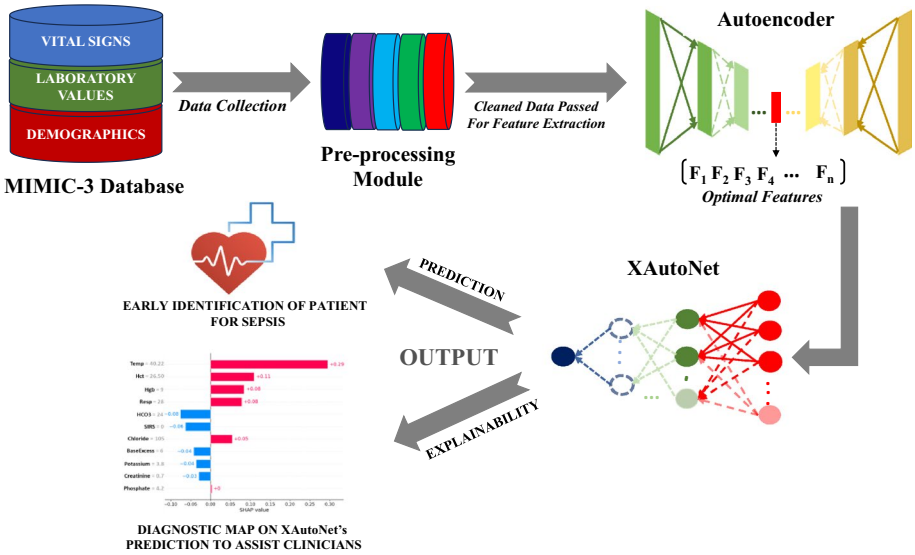

Fig. 1 The overall workflow of our proposed method

Subsequently, leveraging these features, XAutoNet is trained to precisely predict patients with sepsis onset, offering transparent explanations for its predictions.

3.1 Data description

The dataset used in this research was part of PhysioNet Challenge [28]. The original dataset consists of ICU patient's physiological records. However, each patient's data is in a single pipe-delimited text file. All the files have the same column headers, and each row represents a single hour's worth of patient data within that ICU hour stay. The patients were categorized between 0 and 1 based on their physiological records with 0 represents no sepsis and 1 represents that the patient will develop sepsis within the next 6 h. In total, there are 20,336 patient files, the minimum number of records a patient has is 8 whereas the maximum number is 336. To analyse the data, we have concatenated all the patient records and formed a dataset of 7,90,215 rows. The dataset has 40 independent features or indicators, including 8 vital signs, 26 laboratory values, and 6 demographic indicators. Most of the features (especially the laboratory values) have missing values, meaning that the data was not recorded hourly.

3.2 Data pre-processing

When dealing with deadly diseases like sepsis, adequate pre-processing of the data must be done as it belongs to patients that require critical care, and any falsity in the data will lead to biased modelling that may result in loss of life. The steps followed in pre-processing the data are shown in Fig. 2, where the raw data was passed from each block of the pre-processing pipeline, starting from handling missing values and ending in feature selection that

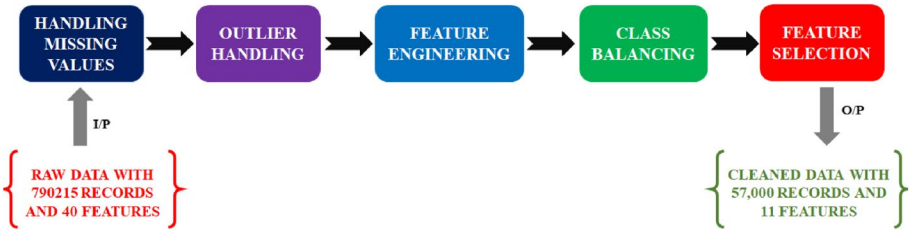

Fig. 2 The pipeline of sub-modules presents in the pre-processing module

resulted in cleaned data on which ML model was trained. The following sections discuss each part of the pipeline in detail.

3.2.1 A) Imputation of missing values

In this study, we have used Multiple Imputation Using Chained Equations (MICE) algorithm [29] as the percentage of missing data was very high. It is a technique by which we effortlessly impute missing values in a dataset by looking at data from other columns and trying to estimate the best prediction for each missing value. MICE compute missing values, accounts for the process that creates the missing data, and preserves the relation and its uncertainty.

Consider a complete dataset (η) following a multivariate distribution, mathematically shown in Eq. (1).

$$\eta \sim \mathcal{P}(\eta, \phi) \tag{1}$$

In Eq. (1), $\mathcal{P}(\eta, \phi)$ is a multivariate distribution specified by ϕ which is a vector of unknown parameter that models η. To get the multivariate distribution of ϕ, we solved the problem implicitly. MICE algorithm obtains the posterior distribution of ϕ iteratively from the conditional probability of the form as shown in the Eq. (2).

$$\mathcal{P}\left(\eta_1|\eta_{-1}, \phi_1\right), \ldots., \mathcal{P}\left(\eta_p|\eta_{-p}, \phi_p\right) \tag{2}$$

where p is the number of missing variables, η_{-i} denote the collection of p-1 variables excluding the i th

variable, and $\phi_1, \ldots, \phi_{\mathrm{p}}$ are conditional densities, respectively. By Gibbs sampler, η_j^T represents the j^{th} imputed value at T^{th} iteration, which is computed using the below-chained Eqs. (3) and (4),

$$\phi_j^{*(T)} \sim \mathcal{P}\left(\phi_{\mathrm{j}}|\eta_{\mathrm{j}}{}^o, \eta_1^{(T)}, \eta_2^{(T)}, \ldots, \eta_{j-1}^{(T)}, \eta_{j+1}^{(T)}, \ldots, \eta_p^{(T)}\right) \tag{3}$$

$$\eta_j^{*(T)} \sim \mathcal{P}\left(\eta_{\mathrm{j}}|\eta_{\mathrm{j}}{}^o, \eta_1^{(T)}, \eta_2^{(T)}, \ldots, \eta_{j-1}^{(T)}, \eta_{j+1}^{(T)}, ..., \eta_p^{(T)}, \phi_j^{*(T)}\right) \tag{4}$$

For $\forall j \in [1, p]$, $\eta_{\mathrm{j}}{}^o$ is the observed value of j^{th} missing value, and $\eta_j^T = (\eta_j^{*(T)}, \eta_{\mathrm{j}}{}^o)$.

To further validate the correctness of the imputation we ran the algorithm to get five variations of the given dataset with different imputed feature values and compared their descriptive statistics.

3.2.2 B) Outlier handling

After the imputation of the missing values, we closely analyzed each feature's distribution. Some features had outliers that must be appropriately handled or could make the model biased. Sometimes outliers can be valid, based on the type of problem that we are solving. In that case, including them becomes crucial as it may provide helpful information in decision-making [30]. And sometimes outliers are invalid, which may be caused due to human mistakes, instrument mistakes or even intentional, making the model less accurate and leading to biased performance [31].

After imputation, the data contained outliers, which were validated before proceeding further as they belonged to ICU patients who needed critical care, and any mistakes while fixing them may lead to loss of life. We use Interquartile Range (IQR) and Z score to find the outliers. The IQR is used to identify outliers by dividing the data into the lower quartile (qr_1), median (qr_2) and upper quartile (qr_3) and then finding the difference between qr_3 and qr_1. The equation for finding the outliers using IQR is mentioned below in Eq. (5) and (6),

$$O_l = qr_1 - 1.5(qr_3 - qr_1) \tag{5}$$

$$O_u = qr_3 + 1.5(qr_3 - qr_1) \tag{6}$$

where O_l denotes the lower range outliers and O_u denotes the upper range outliers. The boxplot works like IQR, which is used to visualize the outliers.

Z score on the other hand is used to measure the spread of data points from their mean. Data points beyond 3rd standard deviation have a Z score of more than 3, considered outliers. The Z score value is formulated below in Eq. (7).

$$Z_{score} = \frac{x - \varepsilon}{sd} \tag{7}$$

where x is the data point, ε and sd is the mean and standard deviation of the distribution to which x belongs. We also fixed some invalid outliers by clipping them within their possible range as advised by the domain expert, while some were genuine, so we kept them as it is.

3.2.3 C) Feature engineering

To add more information in the data, a new feature named SIRS [32] score was generated from the existing features such as: Temperature, Heart rate, Respiration rate, and WBC. SIRS score determines whether a patient can develop sepsis or not by giving a score. The criteria for the SIRS score are mentioned below.

- When patient's body temperature is greater than 38 °C or less than 36 °C.
- When patient's respiration rate is greater than 20 breaths per minute or Partial Carbon dioxide content is less than 32 mmHg.
- When patient's heart rate is greater than 90 beats per minute.
- When patient's white blood cell count is either less than 4000 or greater than 12,000 per mm^3.

The number of above criteria met is the SIRS score for that patient. A patient with SIRS scores as 0 or 1 does not meet the sepsis criteria. Whereas patients with a score of 2, 3 or 4 meets the criteria of sepsis with a higher score indicates higher chance of having the disease.

3.2.4 D) Class balancing

The class distribution in the dataset was unequal as only 2% of the records were marked as sepsis, whereas the remaining 98% belonged to the normal class. To handle the class imbalance problem, both over and under sampling was performed, aiming to keep as much information as possible. The Synthetic Minority Over-Sampling Technique (SMOTE) is used to over-sample the sepsis class resulting in a total of 25,000 data points by slightly moving them toward their neighbour. By doing this, we ensure that the new data points are not exact copies of the existing points and are not too different from them [33]. Finally, to under sample the majority class, we used Cluster-based sampling where unimportant instances of the majority class are removed by using Feature-Space Geometry, which is used to delineate important and unimportant instances. The concept of finding cluster centroid is used by averaging the feature vectors for all the features, over the majority class data points in feature space. The instances that lie farthest from the cluster centroid are considered the most unimportant and are removed based on their importance and number of instances to be resampled [34]. Figure 3 represents the approach of solving the class imbalance problem of the dataset where the records of the positive class were up-sampled, followed by down sampling of the normal records.

The dataset size was reduced to 57,000 records which contained 32,000 records as 0 and 25,000 records as 1 after solving the class imbalance problem. To avoid overfitting the model, we split the dataset into train and test sets and used k-fold cross-validation.

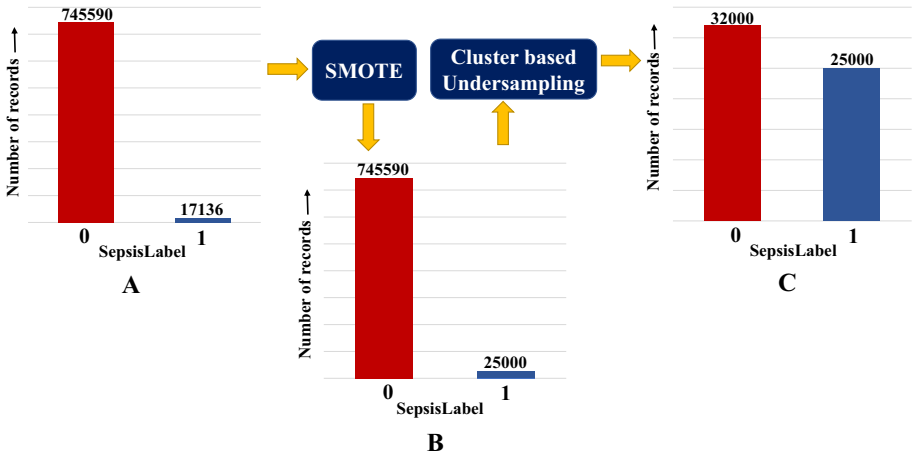

Fig. 3 Class imbalance problem of dataset. A) Initial class distribution of the dataset, B) Class distribution of the dataset after SMOTE and C) Final class distribution of the dataset after Cluster based under sampling

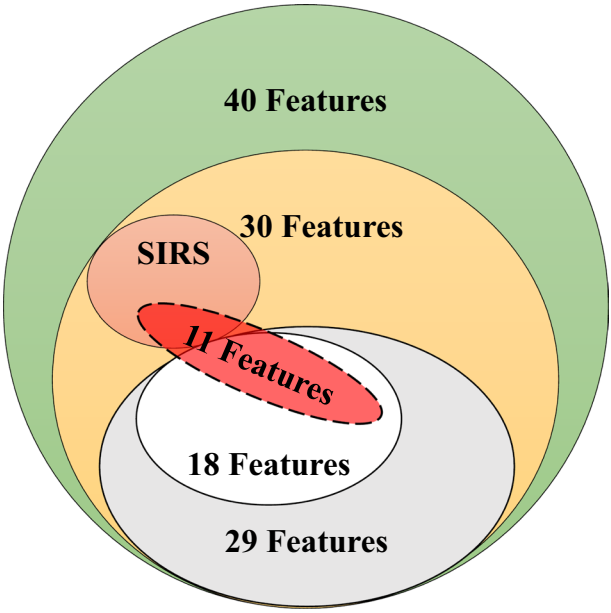

Fig. 4 The Venn diagram of the selection of the optimal number of features

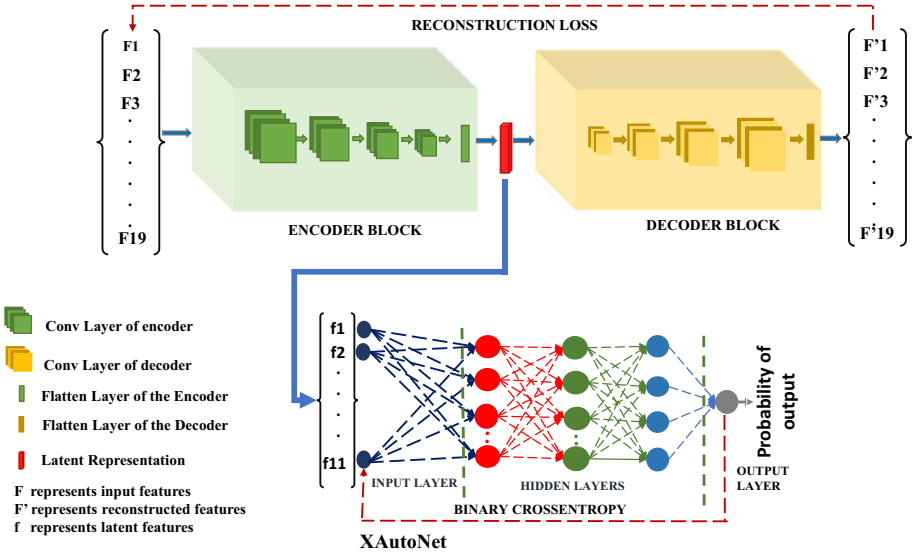

Fig. 5 The complete architecture of the CNN-CNN based Autoencoder with a multilayer neural network for final classification

3.2.5 E) Dimensional reduction

Our work focuses on reducing the feature space by feature selection and then by feature extraction so that the reduced dimension represents most of the original data's information improving the model's predictive power [365]. The Venn diagram in Figs. 4, 5, 6 and 7 represents

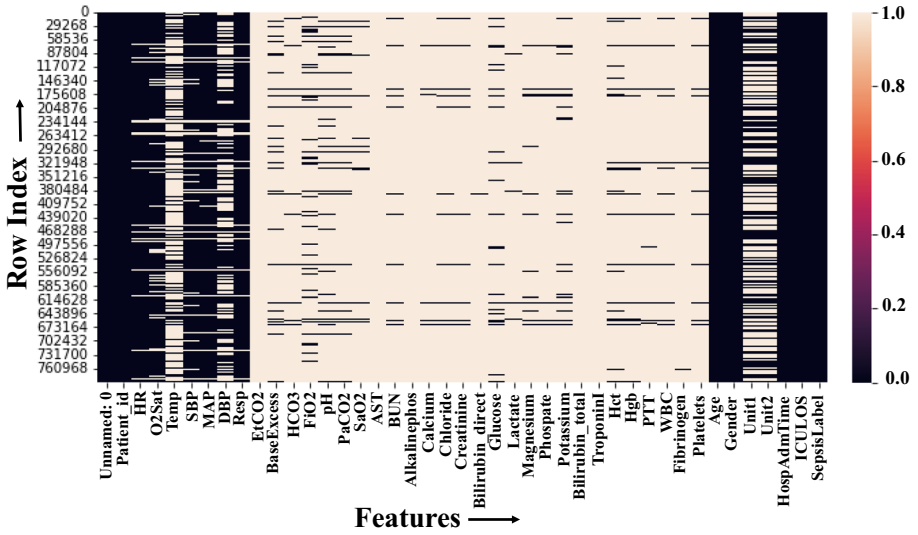

Fig. 6 Heatmap showing the distribution of the missing value in the dataset

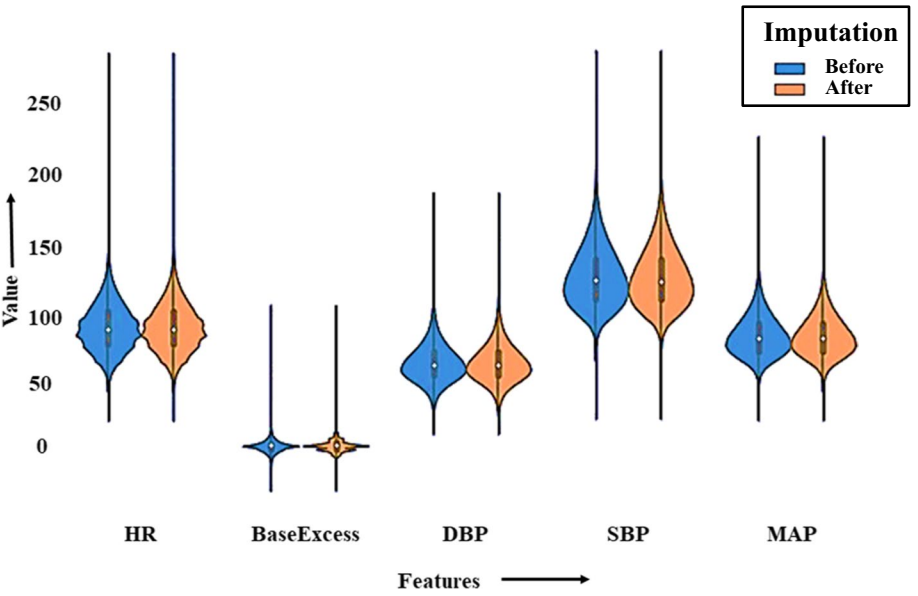

Fig. 7 Distribution of 5 random features before and after imputation of their missing values

the selection of the optimal features where thirty features were extracted out of forty during pre-processing, including direct deletion of empty or unimportant features shown as green circle. These thirty features comprise twenty-nine preexisting features marked as grey circle and the newly generated feature (SIRS) marked as light brown circle. We first used filter methods to select the best nineteen features, including eighteen from grey circle and SIRS, then

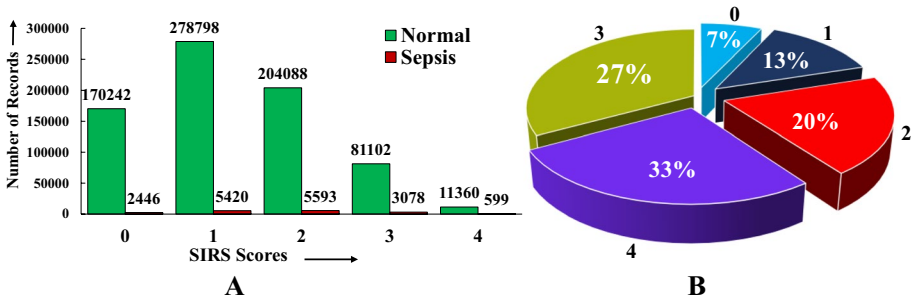

Fig. 8 SIRS analysis. A) Count of records falling in each category of SIRS with respect to SepsisLabel, B) The percentage of patients having sepsis in each category of SIRS

further reduced these nineteen features to eleven optimal features (marked in red ellipse) by feature extraction that convey the most information of the original data. The advantage of this approach is shown in Fig. 8 of Sect. 4.3. The above process is elaborated below in detail.

3.3 Feature selection

After pre-processing, the total number of input features is thirty which is further reduced to nineteen with the help of filter methods. The Mutual Information (MI) score and Chi-square ($\chi 2$) test are used as the data contains both continuous and categorical features. We used MI to select the top eighteen features with the highest scores. It measures the dependency between a feature and the label, which is a non-negative value ranging between 0 and 1, where high score means higher dependency of the label on that feature and 0 denotes independency between them. The formula of MI between a feature (X) and the label (Y) is stated in Eq. (8).

$$MI(X : Y) = En(X) - En(X|Y) \tag{8}$$

where $MI(X : Y)$ is the MI score between X and Y, denoting probability by $\mathbb{P}$, H(X) is the entropy for X defined as,

$$En(X) = -\sum\nolimits_{x \in X} \mathbb{P}(x) \times \ln(\mathbb{P}(x)) \tag{9}$$

and $H(X|Y)$ is conditional entropy for X given Y defined as,

$$En(X|Y) = -\sum\nolimits_{x \in X, y \in Y} \mathbb{P}(X = x, Y = y) \times \ln(\mathbb{P}(X = x)|Y = y)) \tag{10}$$

For selecting the best feature from the remaining two categorical features (SIRS and Gender) we used $\chi 2$ test. The $\chi 2$ tests the relationship between categorical features and gives two values, $\chi 2$ statistics and p-value. Features having high $\chi 2$ statistics score and a p-value less than 0.05 are notable features in determining the label. The formula of $\chi 2$ is shown in Eq. (11).

$$\chi^2 = \sum \frac{1}{E_i} DD^T \tag{11}$$

where D is the difference between mean of the observed vector and the mean of expected vector (E_i).

In our case, $\chi 2$ statistics score for SIRS and gender are 1537.4 and 32.62 and their p-values were 0 and 0.24. So, we selected SIRS feature since the $\chi 2$ statistics is highest along with p-value of less than 0.05. So, after feature selection we ended up with top nineteen features which includes FiO2, O2Sat, Magnesium, SaO2, Lactate, pH, BaseExcess, Potassium, Temp, Calcium, Phosphate, Creatinine, Chloride, HCO3, Resp, Hgb, Hct, BUN (from MI score) and SIRS (from $\chi 2$ test).

3.4 Feature extraction

An autoencoder based architecture was proposed to reduce the above nineteen features into eleven which considers the non-linearity of the data and gives reduced dimensions of features with minimum information loss. The main architectural blocks of the autoencoder comprise an encoder block and a decoder block shown in Fig. 5. The encoder block takes the input features, encodes, or compresses them into reduced dimensions. These reduced dimensions represent most of the input features' valuable information and are stored in the bottleneck layer. The decoder block on the other side takes these reduced dimensions as input and decodes or reconstructs them back to the input features that were feed to the encoder as input. The role of the autoencoder is to minimize this reconstruction loss as low as possible. The working and detailed description of this architecture is mentioned below.

a) The Encoding Block

The encoder block in our proposed architecture combines four 1D convolution layers with decreasing filters used in each layer. For X_t^0 input, the feature extraction of the l^{th} layer by convolutional operation is shown in Eq. (12).

$$CE_t^l = \sum_{k=0}^{N^l} X_{t-k}^{l-1} f_{-k}^l \tag{12}$$

where f_k^l*'s* are the value of the filters associated with the input at l^{th} layer for $k \in [0, N]$, $CE_t^l = X_{t-k}^l$, and N^l is the number of the filters with $N^l > N^{l+1}$ for $l \in [0, L^E]$ where L^E is number of convolutional layers.

Candidate value of encoder at l^{th} layer and t^{th} step (CE_t^l) is transformed into the minimum size of bottleneck (BN) say S for the classification with the help of linear transformation as stated in Eq. (13).

$$BN_t = CE_t^{L^E} W + B \tag{13}$$

For $W \in R^{seqlen \times S}$($seqlen$is the size of $CE_t^{L^E}$) and $B \in \mathbb{R}^S$ is the bias of the network that needs to be learned.

In Eq. (12), to remove the effect of stride, the filter size in each of these layers is set to the number of input features of the encoder which is nineteen. This is intended to remove the sequential effect of the convolution operation. At the end of the last convolution layer, a flatten layer is responsible for flattening the input from the last layer, which further gets connected to a layer with eleven neurons known as the bottleneck layer in the Eq. (13). The eleven neurons represent the input dimension's latent or optimal size of the input dimension which will then be fed to the classifier for classification. This bottleneck layer is responsible in caring most of the information of the input data thereby giving the encoder the privilege to encode anything with minimum loss. We selected this optimal size of the bottleneck from the variation of Recall with respect to the bottleneck size shown in Figs. 9 and 10 of Sect. 4.3.

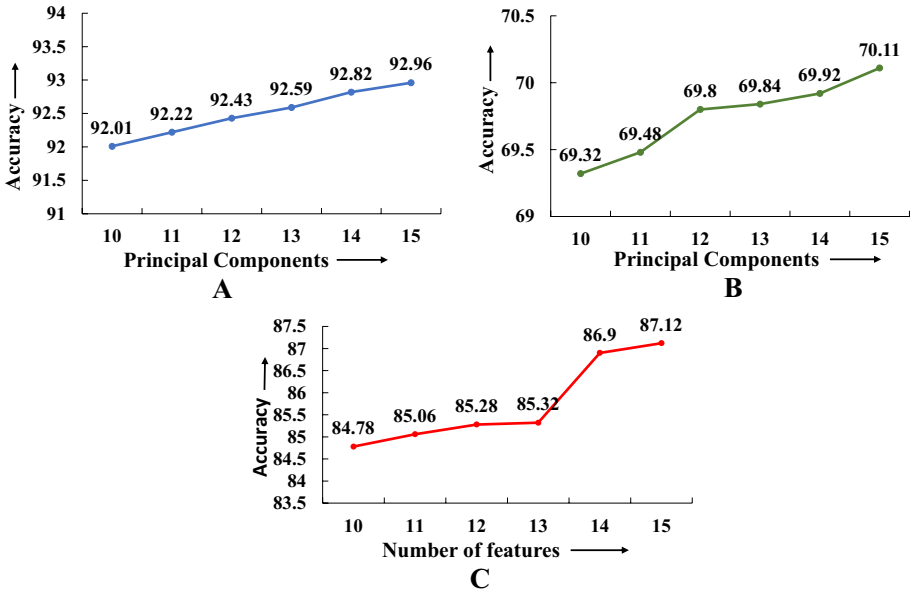

Fig. 9 Line plot showing the performance of SVM with respect to principal components after (A) feature selection followed by dimensional reduction by Principal Component Analysis (PCA), (B) dimensional reduction by PCA, and (C) feature selection by filter method

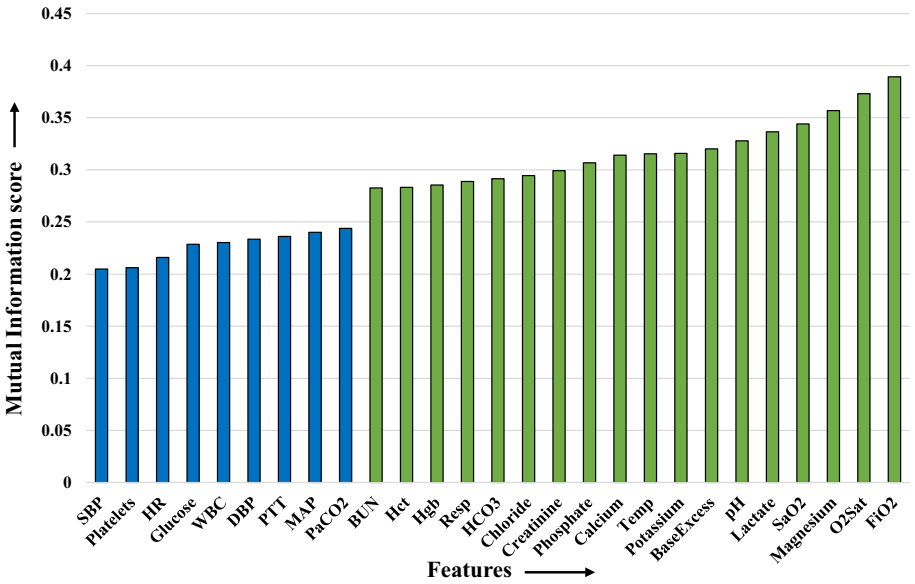

Fig. 10 MI score of all the 27 features where high score means high correlation with SepsisLabel

b) The Decoding Block

The decoder block on the other side is similar in architecture to the encoder block, but its role is to decode the latent dimension back to the original input dimension given to the encoder as an input. For L^D number of convolutional decoder layers, we have the convolutional operation defined as Eq. (14).

$$CD_t^{l\prime} = \sum_{k=0}^{Nl\prime} BN_t^{l\prime-1} f_{-k}^{l\prime} \tag{14}$$

where $N^{l\prime}$ is number of decoder convolutional layers with BN_t^0 is the output from the bottleneck and $N_t^{l\prime} = CD_t^{l\prime} \forall l\prime \in [0, L^D]$.

Before passing the latent dimension to the first convolution layer, it is further reshaped to a vector of size (11,1) to make it compatible with the input shape of the convolution layer. The decoder block also consists of four 1D convolutional layers as you can see in the Eq. (14), the filter size in these convolution layers is set to eleven which is equal to the latent dimension with the same intend of removing the sequentially just like in the encoder block. The convolution layers are also connected sequentially with increasing number of filters ($N^{l\prime} < N^{l\prime+1}$) making it symmetric with the convolution layers in the encoder block resulting in extraction of features from the compressed representation back to that related row of the input data. After the last convolution layer, a flatten layer is used to flatten its output into a single dimension which further gets connected to the final dense layer containing nineteen neurons which represents the corresponding input dimensions to be reconstruct shown in the Eq. (15) with the help of linear transformation,

$$Out_t = CD_t^{L\prime^D} W\prime + B\prime \tag{15}$$

For $W\prime \in \mathbb{R}^{seqlen\prime \times S\prime}$ ($seqlen\prime$ is the size of $CD_t^{L\prime^D}$) and $B\prime \in \mathbb{R}^{S\prime}$ is the bias of the network that needs to be learned.

The output from these $S\prime = 19$ neurons represent the reconstructed input data fed to the encoder initially with minimum information loss. Our objective is to minimize the total reconstruction error with the help of expected mean squared error mentioned in Eq. (16).

$$L(\theta) = E\left[(X - Out(\theta))^2\right] \tag{16}$$

where $Out(\theta)$ is value prediction with the Encoder-Decoder model which is the function of all the parameters (θ).

3.5 XAutoNet

Once the autoencoder was trained, its decoder block was removed with the motive of using the encoder side as dimensional reduction by feeding it those nineteen features and getting a reduced and optimal set of eleven dimension. This reduced dimension is then passed to a deep neural network classifier named XAutoNet which classify no sepsis or sepsis 6 h before, shown in Fig. 5.

The objective is to minimize the binary cross entropy between its output ($X\prime(\theta\prime)$) and ground truth (Q) with respect to all the parameters $\theta\prime$ as defined in Eq. (17).

$$L(\theta) = -\frac{1}{M}\sum_{t=0}^{M} Q_t \log\left(X_t\prime(\theta\prime)\right) + \left(1 - Q_t\right)\log\left(1 - \left(X_t\prime(\theta\prime)\right)\right) \tag{17}$$

where M is the length of dataset.

3.6 GradCAM

To explain the contribution of input features in the feature extraction process by autoencoder, we have used a gradient based class activation map (1D) in weakly supervision [35]. To get the heatmap $h \in \mathbb{R}^N$ for N feature, we first compute the gradient of class c, y^c where $y \in R^c$ is the output from bottleneck, with respect to the feature map vector ($F \in \mathbb{R}^{N\times C}$) from convolutional operation shown in the Eq. (18):

$$\nabla = \frac{\partial y^c}{\partial F} \tag{18}$$

where y^c is calculated using gradient ($\nabla \in R^{N\times C}$) via backpropagation with respect to feature maps($F \in \mathbb{R}^{N\times C}$) where C is number of channels in feature map. Then we calculate the heatmap $H = \{h_i : h_i \in \mathbb{R}^N, \forall i \in [1, N]\}$ by Eq. (19):

$$H = \frac{1}{C}\sum_{\forall i\in[1,C]} h_i \odot \nabla_i \tag{19}$$

where $\left\{\nabla_i : \nabla_i \in \mathbb{R}^N, \forall i \in [1, N]\right\} = \nabla$, C is the number of channels in feature map, and $\odot$ denotes the Hadamard product.

3.7 Deep SHAP

Encoder Visualization using Grad CAM can only tell the most useful feature for bottleneck. To check the contribution of features for being sepsis or normal, we have used deep SHAP [36]. Deep SHAP assigns each feature an importance value for a particular prediction. Deep SHAP is the combination of Deep Learning Important FeaTures (DeepLIFT) [37, 38] and Shapley value [39]. DeepLIFT proposed a DL prediction explanation. For each input x_i, it assigns an attribute $C_{\Delta x_i \Delta y}$ which represents the effect to the corresponding output. For DeepLIFT, this means that $x = h_x(x')$ transforms binary values into their original inputs, where 1 signifies the original input value, and 0 signifies the uninformative background value for the input feature chosen by the user. DeepLIFT states a "summation -to-delta" property as stated in Eq. (20).

$$\sum_{i=[1,n]} C_{\Delta x_i \Delta f(x)} = \Delta f(x) \tag{20}$$

where f(x) represents the model output for the original model, $\Delta f(x) = f(x) - f(r)$, $\Delta x_i = x_i - r_i$, and r is the uninformative background value for the feature.

There are three methods classic equation from game theory for explanation of model predictions, one of them is Shapley values. It assigns a value for each feature representing the effect of the model predictions. For the set of features F, we use a pretrained network f on feature subset ($S \subseteq F$) including that feature ($S \cup i$) and excluding that feature (S). Then, these two outputs are compared on current input i, shown in the Eq. (21),

$$OD_S = f_{SUi}(x_{SUi}) - f_S(x_S), \tag{21}$$

Weights for corresponding OD_S is calculated by the cardinality of the subsets (|S|) and set (|F|) as shown in the Eq. (22):

$$W_S = \frac{|S|!(|F|! - |S|! - 1)}{|F|!} \tag{22}$$

Shapley values (ϕ_i) are calculated using the weighted average of OD_S with weights W_S by Eq. (23):

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} OD_S W_S(A) \tag{23}$$

Property of local accuracy explain the model g(x′) matches the original model f(x) when $x = h_x(x')$ shown in the Eq. (24) below,

$$f(x) = g(x') = \phi_0 + \sum_{i=0}^{M} \phi_i x_i' \tag{24}$$

where $g(x')$ represents the explanation model which consists of Shapley values.

Property of missingness constrains the features where $x_i' = 0$ to have no impact i.e.,

$$x' = 0 \Rightarrow \phi_i = 0$$

SHAP values are the solutions to the equation (A) under $f(x_S) = E[f(x|x_S)]$. A Deep SHAP combines SHAP values calculated for smaller components into SHAP values for the whole network. It does so by recursively passing DeepLIFT's multipliers using the linear approximation as shown in the Eq. (25):

$$\phi_i(f, x) = w_{x_i f_j}(x_i - E[x_i]) \tag{25}$$

$w_{x_i f_j}$ are the weights of deep neural network and calculated using chain rule as shown in the Eq. (26):

$$w_{x_i f_j} = \sum_{k=0}^{j} w_{x_i f_k} w_{x_k f_j} (\text{byChainrule}) \tag{26}$$

3.8 System and environment

The presented methodology is implemented using Tensorflow 2.0. The training of both autoencoder and XAutoNet as well as all experiments are conducted on an NVIDIA RTX A4000 GPU with 16GB dedicated memory, and an Intel i9 processor. The training process involves Adam optimizer, running for 100 epochs with a learning rate of 0.001, and early stopping after 3 epochs. A batch size of 4 is chosen and mean squared error and binary cross-entropy serve as loss functions for autoencoder and XAutoNet, respectively. The training and testing loss for autoencoder are 0.07 and 0.05 and for XAutoNet are 0.07 and 0.08 respectively. Whereas the training and testing accuracy of XAutoNet are 0.96 and 0.94.

4 Results

4.1 Mice Imputation

Because of the high percentage of missing values in the data, which is evident from Fig. 6, MICE method was preferred as it tries to estimate a set of plausible values for the missing data using the distribution of the observed data. Figure 6 represents the occurrence of the missing values in the entire dataset in the form of a heatmap where missing values are represented in bisque colour and black represents actual values. From Fig. 6, we can see that out of 40 features, 36 had missing values. There were 20 features with more than 90% missing values.

After imputing the missing values with the help of MICE, the descriptive statistics of the features were compared with their initial descriptive statistics with the help of violin plot in Fig. 7. In a violin plot, the median is represented by a white marker, the box represents the interquartile range, and the probability density of the data, smoothed by a kernel density estimator. From Fig. 7, it was evident that MICE could estimate the missing values of the features accurately as the difference between their means and standard deviations before and after imputation was negligible.

4.2 Feature engineering

The analysis of the new feature (SIRS score) is shown in Fig. 8. The Fig. 8(A) shows that most of the patients admitted in the ICU has SIRS score less than 3 which implies that those patients might have some other complications which are not related to sepsis that is evident by the green bars that represent the total count of patients' records not developing sepsis in each SIRS score category. The susceptibility of a patient developing sepsis in each category of SIRS score is shown in Fig. 8(B). As mentioned in part C of Sect. 3.2 a patient with high SIRS score is more likely to develop sepsis which is evident from Fig. 8(B) where probability of developing sepsis is highest in score 4 and lowest in score 0.

4.3 Dimensional reduction of feature space

The approach of selecting optimal features by feature selection followed by feature extraction as discussed in part E of Sect. 3.2 helps us to reduce the feature space by considering minimum information loss helping the model to classify correctly. Figure 9 shows the comparison of the performance of SVM with the mentioned approach, with only dimensional reduction and with only feature selection. In Fig. 9(A), the accuracy of SVM was high since the irrelevant features of the training data was removed first by feature selection. Then on the remaining features, dimensional reduction was performed, resulting in reduced dimensional space but rich in valuable information that was utilized by the SVM properly. In Fig. 9(B), reducing the dimensions directly resulted in information loss as some irrelevant features were also considered during feature extraction, resulting in the SVM's poor performance. However, in Fig. 9(C), the performance of the SVM was better than its performance in Fig. 9(B) as the least essential features were removed directly but in comparison with Fig. 9(A), the information loss was high that resulted in its poor performance.

4.4 Feature selection

The selection of the top 18 continuous features with the highest MI score is shown in Fig. 10. Including more features leads to irrelevant information and increases the chance of overfitting and deleting many will lead to information loss. Therefore, we decided to keep the top 18 features marked as green bars in Fig. 10 as there was a sharp drop in MI score of features placed before BUN (marked as blue bars).

4.5 Selection of the bottleneck size

Figure 11 shows the selection of optimal input dimensions for XAutoNet with respect to Recall. We chose Recall over other metrics as we were more concerned about the false negative that the model can commit with different size of the input dimensions. Figure 11(B) shows the saturation of Recall after 11 for bottleneck size, therefore we decided to fix the optimal size of the input dimension as 11. We also further compared this variation with respect to linear and sigmoid kernel of non-linear PCA to cross validate our result in Fig. 11(A). Linear and sigmoid kernels were chosen out of the other kernels for dimensional reduction as these two kernels gave the best results regarding classification performance.

4.6 Performance of XAutoNet

We used K fold cross-validation on the entire dataset to ensure that the proposed XAutoNet is not over fitted and hence can perform well on the entire dataset ensuring its good predictive power. The performance of XAutoNet is shown with respect to accuracy, Precision, Recall and F1 score for fivefold cross-validation in Table 2. The table below shows that it performed well on each fold for all performance metrics with an overall mean of above 0.90 and standard deviation of less than 0.015, indicating good generalization ability.

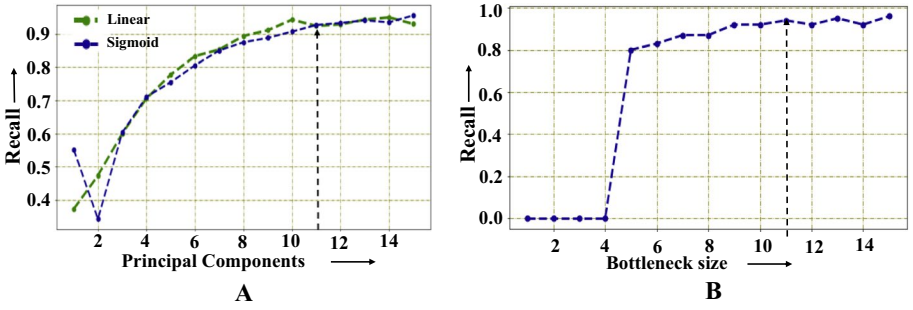

Fig. 11 Optimal features selection. A) Represents the variation of principal components with respect to recall for linear and sigmoid kernel by non-linear PCA, B) Represents variation of bottleneck of Autoencoder with respect to recall

Table 2 Performance scores of XAutoNet in fivefold cross validation

Fold	Accuracy	Precision	Recall	F1 Score
1	0.93	0.92	0.91	0.92
2	0.93	0.92	0.92	0.92
3	0.93	0.95	0.91	0.93
4	0.94	0.94	0.93	0.93
5	0.95	0.94	0.94	0.94
Mean ± SD	0.94 ± 0.0080	0.93 ± 0.012	0.92 ± 0.012	0.93 ± 0.007

5 Comparative study

5.1 With traditional ML algorithms

The performance of XAutoNet was compared with other traditional ML algorithms, which helped us infer how well our model is compared to other models. The performance comparison of the models with respect to Accuracy, Precision, Recall and F1 score is shown in Table 3, where XAutoNet achieves the scores marked in green. From the table below, it is evident that our proposed model outperformed all the traditional ML models with respect to Accuracy, Recall, and F1 Score, ensuring its good predictive power.

5.2 With other autoencoder architectures

Apart from the proposed Autoencoder, which was used for dimensional reduction mentioned in the methodology, we also compared its reductive power with 2 different architectures of Autoencoder that include multilayer perceptron (MLP-MLP) based Autoencoder and a hybrid autoencoder that include convolution neural network-based encoder and a multiplayer perceptron-based decoder (CNN-MLP). We first reduced the dimension of train and test set to the desired number with the help of the above autoencoders and then trained the classifier on this reduced dimension. AUC (Area Under Curve) score and ROC (Receiver Operating Characteristics) curve were used to compare the results. The separation of the two classes entirely depends on how well our classifier understands the reduced dimensional data, which further depends on how well the autoencoder can extract the vital information from it. In Fig. 12,

Table 3 Performance comparison of the XAutoNet with other state of the art ML models with respect to different metrices

Model	Accuracy	Precision	Recall	F1 Score
Naïve Bayes	0.62	0.59	0.43	0.49
Logistic Regression	0.64	0.62	0.45	0.52
Decision Tree	0.86	0.85	0.84	0.84
KNN	0.88	0.89	0.82	0.85
SVM	0.87	0.87	0.86	0.86
Random Forest	0.89	0.89	0.88	0.88
ADA Boost	0.88	0.88	0.82	0.85
Gradient Boost	0.89	0.88	0.86	0.87
XG Boost	0.90	0.90	0.89	0.89
XAutoNet	**0.93**	**0.90**	**0.94**	**0.92**

the ROC curves for all the autoencoder architectures are given, more the area covered by the curve, better will be the classifier in predicting the classes whereas an average performing model will be having a ROC curve same as the dashed blue line. As we can see from Fig. 12, the AUC score of the proposed autoencoder (CNN-CNN based Autoencoder) is better than the AUC score of other 2 architecture which implies that the proposed autoencoder was able to extract most of the information from the complete data which helped the classifier to classify the classes better.

6 Explainability

6.1 Encoder visualization using GradCAM

One of the most challenging tasks in AI is to explain the interpretability of these "black box" models. To explain the autoencoder, we have implemented a gradient-based class activation map to calculate the heatmaps shown in Fig. 13(A). First, we train the model to get the best possible bottleneck vector then we calculate the heatmap mentioned as (EB) for each encoder layer (E) with the help of GradCAM as discussed in Sect. 3.4. Based on our experiments, we can see that Hct is more active and FiO2 is less active among all the features for first encoder layer (E1). Similarly, second explainable block (EB2) corresponding to E2 shows that again Hct is most active, and Lactate is least active, EB3 shows that Phosphate is most active and again Lactate same as EB2 is least active feature for third encoder block (E3), and EB4 shows that Potassium is most active whereas pH and Calcium is least active features for the final encoder layer (E4). To check the overall impact of these features in the creation of the bottleneck, we combine the heatmaps of each encoder layer by discounting the value of heatmap from last encoder layer (E4) by Eq. (27):

$$DHM = \sum_{i=1}^{4} \gamma^i HM_i\left(x_1, x_2, \ldots, x_{19}\right) \tag{27}$$

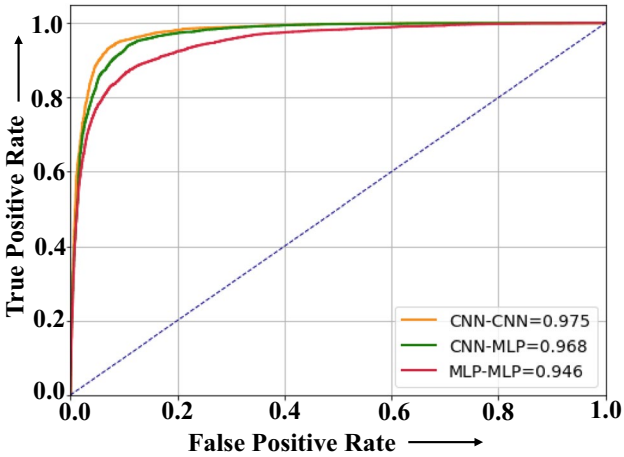

Fig. 12 The ROC curve of each variation of Autoencoder with their corresponding AUC score

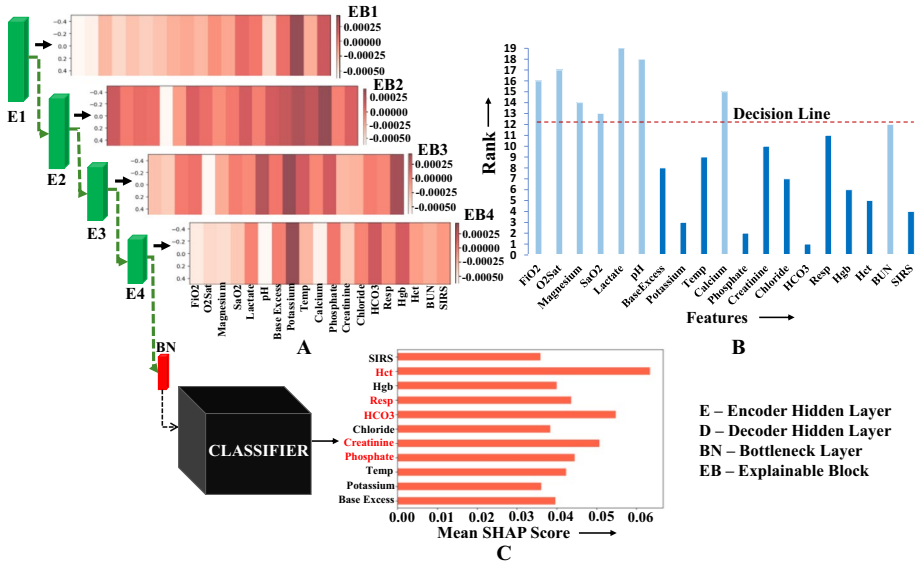

Fig. 13 represents the explain ability of autoencoder and classifiers. (A) heatmaps (EB1 to EB4) showing visual representation of the activation of the features in each convolutional encoder layers marked as green block using Grad-CAM. Darker the shade of a feature in the heatmap, more active that feature is and vice-versa. (B) Represents the rank of the input feature with respect to the bottleneck layer (BN) for the reconstruction using autoencoder. (C) A Blackbox classifier with its explanation using Mean SHAP Score

Based on Discounting HeatMap (DHM), we calculate the rank of the input features for discounting factor $\gamma = 0.9$ which is shown in the Fig. 13(B). Finally, we conclude that BN layer consists of top 11 ranked features like BaseExcess, Potassium, Temp, Phosphate, Creatinine, Chloride, HCO3, Resp, Hgb, Hct and SIRS from Fig. 13(B) based on DHM, which are present below the decision line, while the high ranked features are blurred out.

6.2 XAutoNet visualization using deep SHAP

For the explain ability of XAutoNet we have used deep SHAP which can interpret the relationship between the feature value and the result. We assume a hypothesis that features from the bottleneck are the top 11 feature based on GradCAM feature ranking method shown in Fig. 13(B). To check the features' overall contribution in determining sepsis, we calculated the mean SHAP value from test data shown in Fig. 13(C). Based on our hypothesis, we can conclude that Hct, HCO3, Creatinine, Phosphate, and Resp are the five most responsible features (marked in red color) in determining sepsis as the magnitude of their impact on the model is highest among all the features as shown in Fig. 13(C). On the other hand, the waterfall plot becomes very insightful for explaining the prediction of XAutoNet with respect to a single instance. It tells the magnitude of each feature's impact and whether it contributes to or offsets the model's output. The waterfall plots in Fig. 14 represent the comparison of the impact of each feature, whether contributing (red bar) or offsetting (blue bar), on the model's prediction for 4 different cases. For case A, correctly identified as normal by the model, 8 out of 11 features reduced the prediction probability of this patient being prone to sepsis, with Hgb (haemoglobin) value of 9.99 g/dL contributing the most

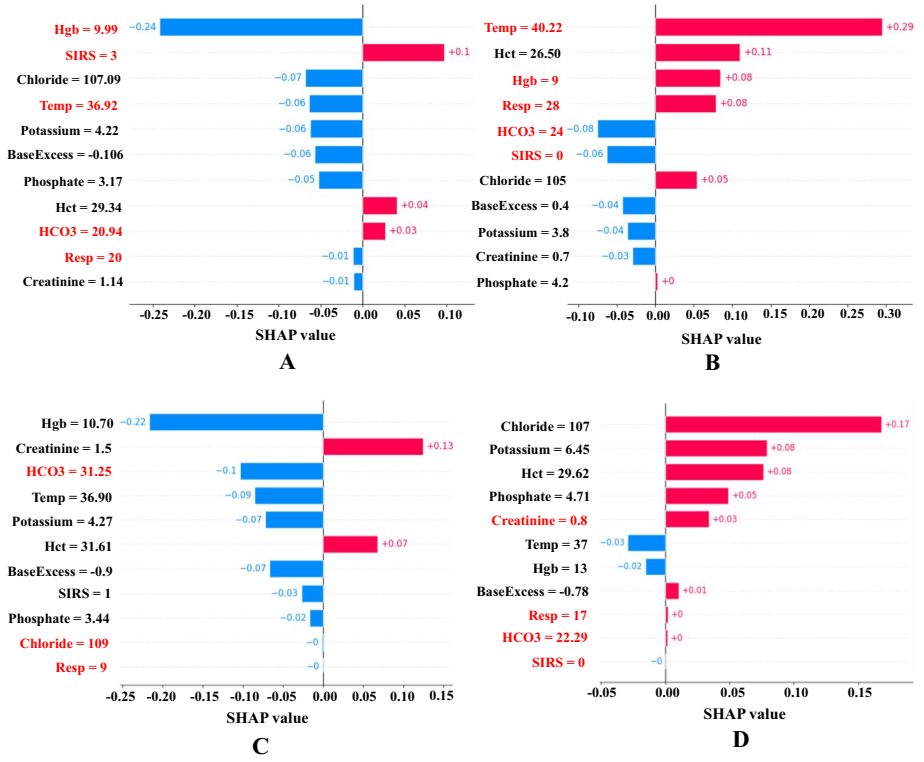

Fig. 14 Waterfall plots showing the explanations of the XAutoNet prediction on a (A) true negative instance, (B) true positive instance, (C) false negative instance and (D) false positive instance. Resp = Respiration rate, Temp = Body Temperature, Hgb = Haemoglobin, Hct = Haematocrit, HCO3 = Bicarbonate

and Creatinine and Resp contributing the least. Whereas for case B, correctly identified as septic 6 h before, 6 out of 11 features increased the prediction probability for being prone to sepsis, with the most contributing feature as Temp (Temperature) with a value of 40.22 °C and Phosphate being the least.

According to Figs. 14(A) and (B), features like Temp, SIRS, Hgb, Resp, and HCO3 of the patients (marked in red) played a crucial role in differentiating them. Other interesting facts with respect to the mentioned figures are that a SIRS score higher than 2 indicates a higher chance of sepsis as per its definition, which is also correctly depicted by the model. The mentioned fact is validated by a positive SHAP value of 0.1 as reported in Fig. 14(A). However, for a SIRS score of 0, a SHAP value of -0.06 is shown in Fig. 14(B). Regarding normal body temperature (Temp) that ranges between 36.1℃ and 37.2℃, a high temperature of 40.22℃ as observed in case B contributes highly to being sepsis positive, with a SHAP score of + 0.29 by XAutoNet. In contrast, the temperature in case A that falls within the normal range, the SHAP score of -0.06 offered by XAutoNet implies that the patient's temperature is not a point of concern for becoming positive. Similarly, for respiration (Resp) whose normal range is between 12 and 20, a high respiration rate of 28 in B indicates its high impact in becoming sepsis positive with a SHAP score of + 0.08 but for A whose respiration rate is considered normal, SHAP score of -0.01 infers that respiration rate is not a point of concern for this patient.

In instances where XAutoNet provides inaccurate predictions, its associated explanations are illustrated in Figs. 14(C) and (D). For Patient C, who was erroneously categorized as normal by the model, several crucial features such as HCO3, Chloride, and Resp (highlighted in red) did not contribute significantly to indicating sepsis, despite their deviations from their respective normal ranges. This observation underscores the model's limitation in effectively capturing the significance of these features for this specific patient. In contrast, for Patient D, who was incorrectly labeled as septic, XAutoNet failed to appropriately highlight the importance of certain features, including Creatinine, Resp, HCO3, and SIRS (marked in red), even though these values fell within their respective normal ranges. In this case, Creatinine, instead of serving as an offsetting element for sepsis, was mistakenly identified as a contributing factor. Furthermore, the model was unable to recognize the significance of Resp, HCO3, and SIRS as relevant features. These two cases illustrate the variations in XAutoNet's explanations when it encounters challenges in correctly identifying patients.

The force plot is another valuable tool alongside SHAP. Similar to the waterfall plot, it presents key features influencing predictions, but linearly. Importance, reflected by SHAP values, is shown by bar length. Longer bars mean higher SHAP values and greater feature importance. Risk factors raising predictions are red, and protection factors lowering predictions are blue.

As seen in Fig. 15(A), a patient with low sepsis risk is depicted. Despite having three risk factors—elevated Potassium level (>5 mmol/L), reduced Chloride level (<98 mmol/L), and diminished HCO3 level (<22 mmol/L), this patient's Phosphate level, Hct level, BaseExcess level, and Temperature all fall within the normal range. Consequently, the predicted risk for this patient is lower than the average. In Fig. 15(B), a patient is forecasted as having a high sepsis risk. This assessment is primarily due to several factors: an elevated temperature (>37.5 degrees Celsius), decreased Potassium level (<3.5 mmol/L), low Hemoglobin (Hgb) level (<12.1 g/dL), with the diminished Potassium level being the most influential contributor to the model's estimation of sepsis.

7 Discussions

This XAutoNet predicts sepsis 6 h in advance using clinical data. A convolution-based autoencoder was also proposed to extract useful information from the ICU records by reducing the input dimensions of the data. Most of the research mentioned in the Literature

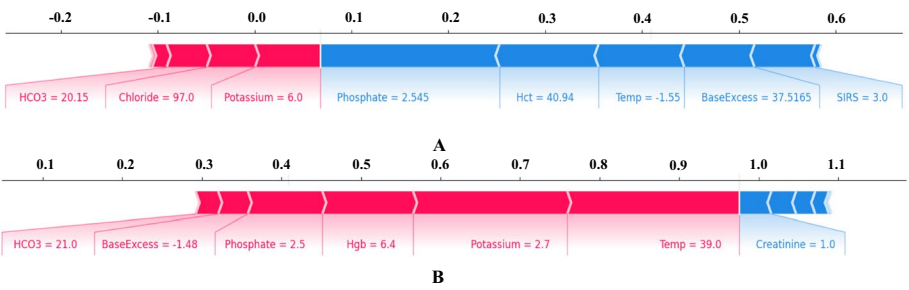

Fig. 15 Force plot showing the explanation of XAutoNet on a (A) low risk patient and (B) high risk patient

Review suffered from information loss that resulted in weak predictive modelling. So, to address this issue, we used the approach to reduce the total feature count by feature selection followed by dimensional reduction. This approach helped us to keep most of the information which is useful for the final model prediction rather than using only dimensional reduction or feature selection that results in information loss and leads to poor performance of the model, as shown in the result section. In feature selection, we used filter methods like the Mutual Information score and Chi-square test as the data contained both continuous and categorical features. We preferred filter methods over others as it considers the dependent variable, it is not model specific and low cost. The class imbalance problem was another issue encountered by the authors in the literature review. The dataset used in this research also has a class imbalance problem. Out of 20,336 patients, there were only 1,790 patients with sepsis (9.68%), and only 17,136 records (2.17%) out of a total of 7,90,215 records were labelled as sepsis. Most of the research preferred under-sampling of the majority class. But by doing this only a small subset of the data was retained, resulting in high information loss, whereas oversampling of the data may lead to an overfit model. So, the strategy of oversampling the minority class using SMOTE and then under sampling the majority class by Cluster based under sampling helped us get 57,000 data points, which was much more compared to other works that were useful to train a more accurate prediction model. To prevent overfitting SMOTE was used to up sample the minority class by a small amount whereas cluster based under sampling was used to remove the unimportant instances by using Feature-Space Geometry which is used to delineate important and unimportant instances. By doing this we make sure that the new data points are not exact copies of the existing points and are not too different from them.

On the other hand, missing values can bias the ML models' results and reduce the accuracy. In terms of data analysis, it has a more significant impact on two aspects: weakened statistics and biased estimation. So, this study uses the MICE algorithm to impute the missing values, as deleting them directly may lead to information loss, which has been proved in early research [40, 41]. The result after imputation by MICE which was discussed in the result section showed great imputation power inferring that the algorithm was able to estimate the probable values of most of the missing data that prevented further information loss as well as helped us dig out new feature (SIRS score) which can contribute to the prediction.

The comparative analysis of the deep neural network with other ML models in the result sections shows the strong approximation power of the network. In comparison to the models presented in Table 1, our approach demonstrated superior performance across most metrics, except for [13, 14], and [18]. Our method not only achieved strong performance but also showcased advantages in terms of the number of input features and interpretability, leveraging a compact subset of attributes. While [13] achieved a higher accuracy than our approach, its utilization of an extensive feature set (38) introduces delays in feature acquisition and raises concerns regarding its practicality in resource-limited settings. Additionally, employing LIME for explanation purposes can be problematic due to its limitations in capturing intricate interactions within complex or nonlinear decision boundaries. The method presented in [14] attained commendable accuracy using 13 features; however, its training on a non-publicly available dataset hindered our ability to validate the results. Conversely, [18] exhibited satisfying performance with 10 input features. Nonetheless, it fell short compared to our method, displaying a 2% lower sensitivity which contributed to higher instances of false alarms. A notable drawback shared among these models is their lack of transparency. Both [13] and [18] struggled to offer comprehensive explanations for their predictions, which casts doubts on their suitability for clinical deployment. In contrast, our

approach strikes a balance between accuracy, feature utilization, and interpretability, making it a promising candidate for practical clinical settings. A few previous works focused on developing sequential models by considering the time feature of this dataset. However, these methods are prone to overfit, require more data for training and their inability to handle complex relationships between inputs and target are the significant drawbacks that resulted in their poor performance. Hence our approach of not using time series modelling not only helped us to attain a high accuracy but also helped in outperforming most of the previous research. The high recall value of the models shows its efficient predictive ability to identify the positive cases, which is the need of the hour in sepsis prediction. In terms of generalizability, its performance in fivefold cross-validation shows great results, ensuring that it performs well on the entire dataset instead of specific sections. Proper pre-processing of the data, including information retention and noise reduction, was the main reason behind the model's performance that early research lagged.

The only limitation of our study is the further generalization of the model due to the unavailability of other important biomarkers and indicators like procalcitonin, C-reactive protein, qSOFA, and NEWS in the dataset are used extensively in the identification of sepsis.

8 Conclusion

In this study, we introduce a predictive model capable of forecasting sepsis 6 h in advance, requiring a streamlined input size of only 11 variables. This design choice renders our model less intricate than the previously proposed alternatives featured in the literature review. Moreover, our model exhibits noteworthy attributes such as strong generalizability and performance enhancement, particularly evident in the realm of recall sensitivity. These aspects collectively position our model favorably against both the mentioned methods and conventional machine learning approaches. Crucially, our method's advantage extends to its practicality within clinical settings, a facet that sets it apart from several existing models. Unlike most of these models, our approach not only offers predictive accuracy but also delivers transparency through meaningful explanations behind its predictions. This quality enhances its potential for real-world deployment, a characteristic often absent in other methods. Furthermore, our research delves into a novel autoencoder architecture, designed to grasp the intricacies of nonlinear data patterns. This architecture capitalizes on meticulous data preprocessing, enabling the extraction of pertinent information and culminating in the robust performance of our final classifier. Future work will be inclusion of more relevant biomarkers, like C-reactive protein, procalcitonin, etc., that are proven to be efficient in sepsis prediction, which can further improve the model's generalizability. Other approaches that include sequential models will be experimented with to exploit the time feature to outperform our current method. Exploration of other pre-processing techniques, including extracting more features from the preexisting ones, and other methods of handling missing data, can improve the model performance.

Abbreviations M3: MIMIC-3; GUH: Ghent University Hospital dataset; SH: Skaraborg Hospital; DHC: Detroit Medical Centre in Michigan; CFH: Carle Foundation Hospital; GCMS: Gas Chromatography Mass Spectrometry; SIRS: Systemic Inflammatory Response Syndrome; MEWS: Modified Early Warning Score; GCS: Glasgow Coma Scale; qSOFA: Quick Sequential Organ Failure Assessment; AI: Artificial Intelligence; EMR: Electronic Medical Records; ML: Machine Learning; CNN: Convolutional Neural Network; GradCAM: Gradient-based Class Activation Map; SHAP: SHapley Additive exPlanations; light GBM: Light Gradient Boosting Machine; RF: Random Forest; RF-CFOA-KELM: Random Forest-improved fruit fly Optimization Algorithm-Kernel Extreme Learning Machine; AUC: Area Under Curve; ANN: Artificial Neural Network; DL: Deep Learning; RNN: Recurrent Neural Network; LSTM: Long-Short-Term

Memory; NLP: Natural Language Processing; LR: Logistic Regression; ROC: Receiver Operating Characteristics; NB: Naive Bayes; SVM: Support Vector Machine; ICU: Intensive Care Unit; KNN: K-Nearest Neighbour; AdaBoost: Adaptive Boosting; AISE: Artificial Intelligence Sepsis Expert; KELM: Kernel Extreme Learning Machine; MICE: Multiple Imputation Using Chained Equations; IQR: Interquartile Range; SMOTE: Synthetic Minority Over-Sampling Technique; MI: Mutual Information; $\chi 2$: Chi-square; BN: Bottleneck; DeepLIFT: Deep Learning Important FeaTures; PCA: Principal Component Analysis; MLP: Multilayer Perceptron; DHM: Discounting HeatMap; M3: MIMIC-3; GUH: Ghent University Hospital dataset; SH: Skaraborg Hospital; DHC: Detroit Medical Centre in Michigan; CFH: Carle Foundation Hospital; GCMS: Gas Chromatography Mass Spectrometry; SIRS: Systemic Inflammatory Response Syndrome; MEWS: Modified Early Warning Score; GCS: Glasgow Coma Scale; qSOFA: Quick Sequential Organ Failure Assessment; AI: Artificial Intelligence; EMR: Electronic Medical Records; ML: Machine Learning; CNN: Convolutional Neural Network; GradCAM: Gradient-based Class Activation Map; SHAP: SHapley Additive exPlanations; light GBM: Light Gradient Boosting Machine; RF: Random Forest; RF-CFOA-KELM: Random Forest-improved fruit fly Optimization Algorithm-Kernel Extreme Learning Machine; AUC: Area Under Curve; ANN: Artificial Neural Network; DL: Deep Learning; RNN: Recurrent Neural Network; LSTM: Long-Short-Term Memory; NLP: Natural Language Processing; LR: Logistic Regression; ROC: Receiver Operating Characteristics; NB: Naive Bayes; SVM: Support Vector Machine; ICU: Intensive Care Unit; KNN: K-Nearest Neighbour; AdaBoost: Adaptive Boosting; AISE: Artificial Intelligence Sepsis Expert; KELM: Kernel Extreme Learning Machine; MICE: Multiple Imputation Using Chained Equations; IQR: Interquartile Range; SMOTE: Synthetic Minority Over-Sampling Technique; MI: Mutual Information; $\chi 2$: Chi-square

Acknowledgements The authors express their deep gratitude to Dr. Ketan Kargirwar, Department of Critical Care, Sir H. N. Reliance Foundation Hospital and Research Centre, Mumbai-400004, India for his invaluable guidance and insight in understanding the critical-care nuances of this article's subject matter.

Authors' contributions Planning and Concept of this research: S.C. and S.R.; Data collection and cleaning: S.C. and K.T.; Implementation and study: S.C., K.T. and K.K, statistical analysis and data interpretation: S.C., B.R.P. and S.R.; Manuscript writing: S.C. K.T. and K.K. critical revision: B.R.P. and S.R. All authors have read and agreed to the published version of the manuscript.

Funding This research work is supported by the RFIER-Jio Institute's "CVMI-Computer Vision in Medical Imaging" research project fund (Grant No. 2022/33185004) under the AI for ALL" research centre.

Source code availability The complete GitHub source code is available at: https://github.com/Snehashis100/XAutoNet1

Data availability The dataset used in this research was part of PhysioNet Challenge (https://physionet.org/content/challenge-2019/1.0.0/). The details is included in the data description section.

Declarations

Competing interests Authors have no Competing interests to declare.

References

1. Rudd KE, Johnson SC, Agesa KM, Shackelford KA, Tsoi D, Kievlan DR, ... & Naghavi M (2020) Global, regional, and national sepsis incidence and mortality, 1990–2017: analysis for the Global Burden of Disease Study. Lancet 395(10219):200–211
2. Roy S, Meena T, Lim S (2022) Demystifying Supervised Learning in Healthcare 4.0: A New Reality of Transforming Diagnostic Medicine. *Diagnostics, 12*.
3. Fu M, Yuan J, Lu M, Hong P, Zeng M (2019). An ensemble machine learning model for the early detection of sepsis from clinical data. In *2019 Computing in Cardiology (CinC)* (pp. Page-1). IEEE.
4. Liu R, Greenstein JL, Sarma SV, Winslow RL (2019) Natural language processing of clinical notes for improved early prediction of septic shock in the ICU. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 6103–6108). IEEE.

5. Goh KH, Wang L, Yeow AYK, Poh H, Li K, Yeow JJL, Tan GYH (2021) Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare. Nat Commun 12(1):1–10
6. Teng AK, Wilcox AB (2020) A review of predictive analytics solutions for sepsis patients. Appl Clin Inform 11(03):387–398
7. Islam MM, Nasrin T, Walther BA, Wu CC, Yang HC, Li YC (2019) Prediction of sepsis patients using machine learning approach: a meta-analysis. Comput Methods Programs Biomed 170:1–9
8. Lauritsen SM, Thiesson B, Jørgensen MJ, Riis AH, Espelund US, Weile JB, Lange J (2021) The Framing of machine learning risk prediction models illustrated by evaluation of sepsis in general wards. NPJ Digit Med 4(1):1–12
9. Zargoush M, Sameh A, Javadi M, Shabani S, Ghazalbash S, Perri D (2021) The impact of recency and adequacy of historical information on sepsis predictions using machine learning. Sci Rep 11(1):1–12
10. Chakraborty S, Kumar K, Reddy PB, Roy S (2023) "An Explanable AI based clinical assistance model for identifying patients with the onset of sepsis," IEEE 24th International Conference on Information Reuse and Integration for Data Science, August 4 - August 6, Seattle, WA, US
11. Nedee JA (2017) Early Identification of Sepsis Risk through the Prediction of Positive Blood Cultures Using Temporal Models in Tensorflow. Ghent University.
12. Yan MY, Gustad LT, Nytrø Ø (2022) Sepsis prediction, early detection, and identification using clinical text for machine learning: a systematic review. J Am Med Inform Assoc 29(3):559–575
13. Shankar A, Diwan M, Singh S, Nahrpurawala H, Bhowmick T (2021) Early Prediction of Sepsis using Machine Learning. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 837–842). IEEE.
14. Singh YV, Singh P, Khan S, Singh RS (2022) A Machine Learning Model for Early Prediction and Detection of Sepsis in Intensive Care Unit Patients. J Healthc Eng 2022
15. Zhao X, Shen W, Wang G (2021) Early prediction of sepsis based on machine learning algorithm. Comput Intell Neurosci 2021
16. Debie E, Shafi K (2019) Implications of the curse of dimensionality for supervised learning classifier systems: theoretical and empirical analyses. Pattern Anal Appl 22(2):519–536
17. El-Rashidy N, Abuhmed T, Alarabi L, El-Bakry HM, Abdelrazek S, Ali F, El-Sappagh S (2022) Sepsis prediction in intensive care unit based on genetic feature optimization and stacked deep ensemble learning. Neural Comput Appl 1–30
18. Mohamed A, Ying H, Sherwin R (2020) Electronic-medical-record-based identification of sepsis patients in emergency department: a machine learning perspective. In 2020 International Conference on Contemporary Computing and Applications (IC3A) (pp. 336–340). IEEE.
19. Taneja I, Reddy B, Damhorst G, Dave Zhao S, Hassan U, Price Z, ... & Zhu R (2017) Combining biomarkers with EMR data to identify patients in different phases of sepsis. Sci Rep 7(1):1–12
20. Scherpf M, Gräßer F, Malberg H, Zaunseder S (2019) Predicting sepsis with a recurrent neural network using the MIMIC III database. Comput Biol Med 113:103395
21. Calvert JS, Price DA, Chettipally UK, Barton CW, Feldman MD, Hoffman JL, ... & Das R (2016) A computational approach to early sepsis detection. Comput Biol Med 74:69–73
22. Nemati S, Holder A, Razmi F, Stanley MD, Clifford GD, Buchman TG (2018) An interpretable machine learning model for accurate prediction of sepsis in the ICU. Crit Care Med 46(4):547
23. Wang X, Wang Z, Weng J, Wen C, Chen H, Wang X (2018) A new effective machine learning framework for sepsis diagnosis. IEEE access 6:48300–48310
24. Roy, Sudipta, Timothy D. Whitehead, James D. Quirk, Amber Salter, Foluso O. Ademuyiwa, Shunqiang Li, Hongyu An, Kooresh I. Shoghi. (2020) Optimal co-clinical radiomics: Sensitivity of radiomic features to tumour volume, image noise and resolution in co-clinical T1-weighted and T2-weighted magnetic resonance imaging. EBioMedicine 59
25. Roy, Sudipta, Timothy D. Whitehead, Shunqiang Li, Foluso O. Ademuyiwa, Richard L. Wahl, Farrokh Dehdashti, Kooresh I. Shoghi (2022) Co-clinical FDG-PET radiomic signature in predicting response to neoadjuvant chemotherapy in triple-negative breast cancer. Eur J Nucl Med Mol Imaging 1–13
26. Weiss SJ, Guerrero A, Root-Bowman C, Ernst A, Krumperman K, Femling J, Froman P (2019) Sepsis alerts in EMS and the results of pre-hospital ETCO2. Am J Emerg Med 37(8):1505–1509
27. Rodelo JR, De la Rosa G, Valencia ML, Ospina S, Arango CM, Gómez CI, ... & Jaimes FA (2012) D-dimer is a significant prognostic factor in patients with suspected infection and sepsis. Am J Emerg Med 30(9):1991–1999
28. Reyna, Matthew A, Josef, Christopher S, Jeter, Russell, Shashikumar, Supreeth P, Westover, M. Brandon, Nemati, Shamim, Clifford, Gari D, Sharma, Ashish (2020) Early Prediction of Sepsis From Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019. Crit Care Med 48(2):210-217. 10.1097/CCM.0000000000004145

29. Van Buuren S, Groothuis-Oudshoorn K (2011) mice: Multivariate imputation by chained equations in R. J Stat Softw 45:1–67
30. Caroline Cynthia P, Thomas George S (2021) An outlier detection approach on credit card fraud detection using machine learning: a comparative analysis on supervised and unsupervised learning. In Intelligence in Big Data Technologies—Beyond the Hype (pp. 125–135). Springer, Singapore.
31. Chakravarty S, Demirhan H, Baser F (2020) Fuzzy regression functions with a noise cluster and the impact of outliers on mainstream machine learning methods in the regression setting. Appl Soft Comput 96:106535
32. Chakraborty RK, Burns B. Systemic Inflammatory Response Syndrome. 2022 May 30. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2022 Jan–. PMID: 31613449.
33. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
34. Yen SJ, Lee YS (2009) Cluster-based under-sampling approaches for imbalanced data distributions. Expert Syst Appl 36(3):5718–5727
35. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision (pp. 618–626).
36. Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. Adv Neural Inf Process Syst 30
37. Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. In International conference on machine learning (pp. 3145–3153). PMLR.
38. Shrikumar A, Greenside P, Shcherbina A, Kundaje A (2016) Not just a black box: Learning important features through propagating activation differences. *arXiv preprint* arXiv:1605.01713.
39. Fryer D, Strümke I, Nguyen H (2021) Shapley values for feature selection: the good, the bad, and the axioms. IEEE Access 9:144352–144360
40. Raaijmakers QA (1999) Effectiveness of different missing data treatments in surveys with Likert-type data: Introducing the relative mean substitution approach. Educ Psychol Measur 59(5):725–748
41. Kim JO, Curry J (1977) The treatment of missing data in multivariate analysis. Sociol Methods Res 6(2):215–240

Authors and Affiliations

Snehashis Chakraborty[1] · Komal Kumar[1] · Kalyan Tadepalli[2] · Balakrishna Reddy Pailla[3] · Sudipta Roy[1]

✉ Sudipta Roy
sudipta1.roy@jioinstitute.edu.in

Snehashis Chakraborty
Snehashis1.C@jioinstitute.edu.in

Komal Kumar
komal2.Kumar@jioinstitute.edu.in

Kalyan Tadepalli
Kalyan.Tadepalli@rfhospital.org

Balakrishna Reddy Pailla
balakrishna.pailla@ril.com

[1] Artificial Intelligence & Data Science, Jio Institute, Navi Mumbai -410206, India

[2] Sir H. N. Reliance Foundation Hospital and Research Centre, Mumbai -400004, India

[3] Reliance Jio - Artificial Intelligence Centre of Excellence (AICoE), Hyderabad - 500081, India