scientific reports

Check for updates

OPEN

Bio inspired feature selection and graph learning for sepsis risk stratification

D. Siri[1], Raviteja Kocherla[2]✉, Sudharshan Tumkunta[3], Pamula Udayaraju[4], Krishna Chaitanya Gogineni[5], Gowtham Mamidisetti[6] & Nanditha Boddu[7]

Sepsis remains a leading cause of mortality in critical care settings, necessitating timely and accurate risk stratification. However, existing machine learning models for sepsis prediction often suffer from poor interpretability, limited generalizability across diverse patient populations, and challenges in handling class imbalance and high-dimensional clinical data. To address these gaps, this study proposes a novel framework that integrates bio-inspired feature selection and graph-based deep learning for enhanced sepsis risk prediction. Using the MIMIC-IV dataset, we employ the Wolverine Optimization Algorithm (WoOA) to select clinically relevant features, followed by a Generative Pre-Training Graph Neural Network (GPT-GNN) that models complex patient relationships through self-supervised learning. To further improve predictive accuracy, the TOTO metaheuristic algorithm is applied for model fine-tuning. SMOTE is used to balance the dataset and mitigate bias toward the majority class. Experimental results show that our model outperforms traditional classifiers such as SVM, XGBoost, and LightGBM in terms of accuracy, AUC, and F1-score, while also providing interpretable mortality indicators. This research contributes a scalable and high-performing decision support tool for sepsis risk stratification in real-world clinical environments.

Keywords Wolverine optimization algorithm, Generative pre-training-graph neural networks, Synthetic minority over-sampling technique, Support vector machines, Sepsis prediction

Sepsis results from the intricate interactions between an invading microbe and the host's immunological, inflammatory, and coagulation responses, making it a potentially fatal condition for critically ill patients in the US[1]. Over 1.7 million adults in the US experience sepsis each year, with approximately 270,000 deaths. Around one-third of hospitalized individuals develop sepsis[2]. While standardized criteria are still under development, several identifying methods are currently available[3]. Multiple factors contribute to disparities in critical care[4]. Healthcare providers often exhibit biases when treating patients from diverse socioeconomic backgrounds[5]. Such disparities are increasingly studied with the rise of data-driven and AI-powered healthcare systems[6]. As Electronic Health Records (EHRs) advance, machine learning applications for sepsis risk prediction continue to strengthen[7]. However, the interaction between biases and inequalities in risk prediction models has not been clearly described[8]. This study examines a cohort of critical care patients, highlighting variations in sepsis incidence and outcomes among different social determinant groups. It also investigates how different subpopulations of sepsis patients fare when mortality prediction using machine learning classifiers fluctuates[9].

EHRs, imaging data, genomic information, and other large volumes of medical data can be better understood with the use of artificial intelligence (AI)[10]. Decision trees, support vector machines (SVMs), and ensemble approaches like XGBoost and LightGBM have found extensive applications in predictive analytics, particularly in early diagnosis and prognosis. Additionally, CNNs and RNNs, two types of deep learning models, have demonstrated remarkable effectiveness in clinical image analysis, speech recognition, and time-series prediction[11]. Cutting-edge AI paradigms, such as Graph Neural Networks (GNNs) and Generative Adversarial Networks (GANs), have opened new opportunities in drug development, personalized medicine, and predictive modeling[12]. These models enable researchers to draw insights from diverse data sources, describe complex

[1]Department of CSE, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India. [2]Department of Computer Science and Engineering, Malla Reddy University, Hyderabad 500043, India. [3]Meta, Bellevue, WA, USA. [4]Department of Computer Science and Engineering, SRM University-AP, Amaravati, Andhra Pradesh, India. [5]Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India. [6]Department of Computer Science and Engineering (AI & ML), St. Martin's Engineering College, Hyderabad, India. [7]Department of Information Technology, Vidya Jyothi Institute of Technology, Hyderabad, India. ✉email: tejakcse@gmail.com

medical interactions, and synthesize new hypotheses. Moreover, chatbots and other automated systems for patient engagement and clinical documentation are being developed using Natural Language Processing (NLP) technologies, while reinforcement learning optimizes treatment regimens[13,14]. To further advance preventative medicine and real-time health management, AI models are also being integrated into wearable technology and remote monitoring systems. Despite its promise, AI adoption in healthcare faces challenges related to data privacy, model interpretability, and regulatory compliance[15]. Addressing these issues is essential to build confidence and ensure the safe and effective application of AI technologies in clinical practice. By improving patient outcomes, reducing inefficiencies, and equipping practitioners with data-driven tools, AI is transforming healthcare[16]. The integration of AI into healthcare systems holds immense potential to revolutionize global health standards as the science continues to advance.

These advancements motivate the adoption of generative pre-training in Graph Neural Networks (GNNs) to enhance model generalization across heterogeneous clinical data. Traditional GNNs often require extensive labeled data and are constrained by limited interpretability and scalability. To overcome these limitations, we introduce a novel Generative Pre-Training Graph Neural Network (GPT-GNN) framework that reconstructs both graph structure and node attributes in a self-supervised manner, allowing the model to capture underlying clinical patterns from unlabelled patient data.

In this study, the GPT-GNN is combined with the Wolverine Optimization Algorithm (WoOA) for feature selection and the TOTO metaheuristic for model fine-tuning—forming a unified pipeline tailored for sepsis risk stratification. Unlike existing models that rely solely on feature-based predictions, our graph-based approach leverages inter-patient relationships and medical context encoded in graph topologies. This integration enables improved accuracy, interpretability, and scalability on real-world datasets. To the best of our knowledge, this is the first study to apply a generative graph pre-training framework with metaheuristic optimization for sepsis outcome prediction.

Key contributions of this manuscript are as follows:

- Developed a novel Generative Pre-Training Graph Neural Network (GPT-GNN) framework to model graph structures and attributes simultaneously for improved sepsis outcome predictions.
- Employed the Wolverine Optimization Algorithm (WoOA) for optimal feature selection, enhancing model accuracy and interpretability.
- Designed the GPT-GNN to handle large-scale graphs using sub-graph sampling, addressing limitations of existing graph pre-training methods.

This paper is structured as follows: Sect. "Related work" reviews relevant literature; Sect. "Proposed methodology" details the proposed methodology; Sect. "Results and discussion" analyzes results; and Sect. "Conclusion" concludes the findings.

Related work

Machine learning researchers Gao et al.[18] improved the interpretability of their model and addressed the shortcomings of earlier research by creating a model that better predicts sepsis outcomes with a smaller collection of features. Using the MIMIC-IV database, this study examines the outcomes for critical care patients with sepsis in adults. In this study, 38 features were chosen using modern data extraction techniques like Google BigQuery and strict selection criteria. A thorough evaluation of the literature and clinical expertise also played a role in this selection. Handling variables and balancing the data using the Synthetic Minority Over-sampling Technique (SMOTE) were part of the data preprocessing. Decision Trees, XGBoost, Gradient Boosting, LightGBM, MLP, SVM, and Random Forest were among the machine learning models tested. Hyperparameter tuning was conducted using the Sequential Halving and Classification (SHAC) algorithm, and performance and computational efficiency were assessed using both the train-test split and cross-validation approaches. A confidence interval of ± 0.01 and an area under the curve (AUC) of 0.94 were achieved by the Random Forest model, making it the most successful.

One study that attempted to use machine learning to predict early death in sepsis patients was conducted by Zhang et al.[5]. However, not much is known about the benefits of different machine learning approaches or how to build predictive variables using ML. The study aimed to determine whether machine learning could reliably predict deaths from sepsis at various time points through a systematic review and meta-analysis. Up until August 9, 2022, the databases PubMed, Embassy, Cochrane, and Web of Science were searched. The Forecast Model Risk of Bias Assessment Tool (PROBAST) was used to evaluate the potential for bias in the prediction models. Additionally, the study summarized the current predictive variables used to build models for sepsis death prediction and conducted subgroup analysis based on time of death and the type of model utilized. In all, 104 different models were covered by 50 separate original investigations. With a combined Concordance index (C-index) of 0.799, sensitivity of 0.81, and specificity of 0.80 in the training set, and 0.774, 0.71, and 0.68 in the validation set, respectively, machine learning models demonstrated superior C-index, sensitivity, and specificity compared to traditional clinical scoring techniques. The best machine learning models were eXtreme Gradient Boosting (XGBoost) and Random Forest (RF) due to their superior accuracy with respect to comparable modeling variables.

Using large-scale data, Bao et al.[2] trained machine learning representations to forecast the death rate of sepsis patients. With the Medical Intensive Care IV (MIMIC-IV) serving as the training set and the Philips eICU Collaborative Research Database serving as the testing set, sepsis patients were retrieved from both databases. LightGBM, GBM, and XGBoost were the top three algorithms in terms of AUC in the test set. After making adjustments to the model's parameters, the LightGBM model achieved an AUC value of 0.99 on the training set and 0.96 on the test set, representing high performance across both sets of data. Clinical decision aids that use

models trained on real-world sepsis patients' EHRs using the LightGBM method can improve patient prognoses and reduce the likelihood of negative outcomes by properly predicting whether sepsis patients will die.

The effect of a deep-learning model (COMPOSER) on patient outcomes for early sepsis prediction was evaluated by Boussina et al.[19]. Within the UC San Diego Health System, the study conducted a before-and-after quasi-experimental study at two separate emergency departments (EDs). From 1/1/2021 to 4/30/2023, 6,217 adult septic patients were included. A COMPOSER-triggered Best Practice Advisory (BPA) directed at nurses was the exposure that was examined. Both the 705 days before and the 145 days after the intervention were used to assess in-hospital compliance, the number of ICU visits, the number of days without sepsis, and the change in sequential organ failure scores (SOFA) 72 h after sepsis onset. A Bayesian structural time-series approach was used for the causal effect analysis, which included confounder adjustments to determine the exposure's significance at a 95% confidence level. According to the causal inference analysis, there was a 4% decrease in the 72-hour SOFA change after sepsis onset, a 5% increase in sepsis bundle compliance, and a 1.9% decrease in mortality. The corresponding relative decrease was 17%. The 95% confidence intervals for these outcomes were 0.3–3.5%, 2.4–8.0%, and 1.1–7.1%, respectively. These findings indicate that early sepsis prediction using COMPOSER led to a considerable decrease in mortality and an increase in compliance with the sepsis bundle.

Patients with sepsis-associated acute kidney injury (SA-AKI) have an increased risk of dying while hospitalized, and Li et al.[3] developed and tested a machine learning (ML) model to predict this risk. The Medical Information Mart IV was utilized to gather SA-AKI patient data from 2008 to 2019. Following feature selection using Lasso regression, six ML techniques were employed to construct the model. After calculating the AUC and accuracy, the best model was selected. Additionally, the best model was analyzed using Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) values. Among 8,129 patients (median age 57.9%, 4,708 males), 24 out of 44 clinical parameters collected during ICU admission were used to build ML models for prognosis. The eXtreme Gradient Boosting (XGBoost) model outperformed the other five models with an AUC of 0.794. Based on SHAP values, the four most important variables were age, respiration rate, and simplified acute physiology score. The LIME algorithm was used to clarify personalized predictions. XGBoost outperformed the other ML models in early mortality risk prediction for SA-AKI.

The study by Koozi et al.[4] sought to determine how well SAPS-3 and SOFA scores predicted mortality in ICU sepsis patients and develop a more user-friendly model. Adults admitted to four general ICUs who met the Sepsis-3 criteria were analyzed retrospectively. By utilizing backward stepwise multivariate logistic regression, a basic prognostic model was developed. SAPS-3, SOFA, and the basic model's AUC were evaluated. In total, 1984 admissions were included. SOFA exhibited an AUC of 0.67 (95% CI 0.64–0.70).

The study by Alanazi et al.[1] explored a novel machine learning method. ICU patients' vital signs and clinical laboratory findings were analyzed to identify and forecast the earliest symptoms of sepsis. The study utilized data mining methods and proportional hazards modeling to examine survival rates and make predictions. Patients admitted to the ICU between April and October 2018, aged 14 and above, were included in the BESTCare data analysis at KAMC. A total of 1,178 individuals diagnosed with sepsis had their medical records examined. Two methods for predicting sepsis in ICU patients were investigated. With only three variables, the survival-based regression model had moderate predictive power ($p = 0.02$). Since linear classification algorithms assume variable independence, other data mining algorithms may have drawbacks. To progress in this field, data properties must be carefully cleaned and selected.

The Double Fusion Sepsis Predictor (DFSP) is an innovative early warning model introduced by Duan et al.[17]. DFSP integrates early and late fusion methods into a single framework. Initially, a hybrid deep learning model combining recurrent and convolutional neural networks extracts deep features. Early fusion merges deep features with clinical scores to create a joint feature representation. Late fusion combines this representation with multiple tree-based models to produce sepsis onset risk ratings. A retrospective study of ICU patients in Shanghai, China, demonstrated DFSP's superior performance compared to state-of-the-art methods in early sepsis prediction.

An extensive retrospective cohort from the University of Michigan (2018–2020) was utilized by Kamran et al.[20] to assess the Epic sepsis model (ESM). The study evaluated the model's ability to forecast sepsis before clinical recognition and treatment plans. Antibiotic administration, fluid intake, blood culture results, and lactate measurements were identified as treatment indicators. Among 77,582 hospitalizations, sepsis occurred in 4.9% (3766 cases). Combining predictions before and after clinical recognition yielded an AUROC of 0.62 (95% CI, 0.61–0.63). Excluding predictions post-recognition reduced the AUROC to 0.47 (95% CI, 0.46–0.48).

Researchers Mangalesh et al.[16] analyzed 267 cases of blood-culture-confirmed sepsis. Data included clinical and laboratory information acquired at admission. Outcomes, such as mortality and ICU stay duration, were examined. Sequential Organ Failure Assessment (SOFA), Systemic Immune-inflammation Index (SII), Neutrophil-to-Lymphocyte Ratio (NLR), and Platelet-to-Lymphocyte Ratio (PLR) were calculated. Multivariate regression identified factors predicting length of stay and death. Optimal cut-offs were determined using AUROC curves. Of 267 patients, 76 (28.5%) died. SII, NLR, and PLR independently predicted mortality, with SII showing the highest AUROC (0.848).

To forecast ICU sepsis cases, Moor et al.[13] created and validated a deep learning system using 136,478 admissions from four international ICU databases (2001–2016). The dataset included hourly-resolved sepsis annotations using Sepsis-3 criteria. The model achieved an AUC of 0.761 (95% CI, 0.746–0.770) in external tests and 0.846 (95% CI, 0.841–0.852) internally. Early detection provided a 3.7-hour window for intervention (95% CI, 3.0-4.3).

Sun et al.[21] developed an AI model for sepsis risk prediction in trauma patients, using 244 patients from their hospital and 717 from a US database. Machine learning methods like logistic regression, decision trees, XGBoost, and LightGBM were employed. The Neural Network (NN) model achieved the highest AUC (0.932, 95% CI: 0.917–0.948). LightGBM and XGBoost followed with AUCs of 0.912 and 0.891, respectively.

Wang et al.[6] compared presepsin to other blood indicators for early-stage SAP prognosis. The study included 48 septic shock patients and 53 septic prevention patients. Presepsin levels rose significantly in sepsis patients over time. Its prognostic accuracy surpassed procalcitonin and SOFA scores, making it a valuable biomarker for secondary sepsis and mortality prediction.

Research gap

Despite significant advancements in sepsis prediction models, existing approaches often lack comprehensive interpretability and scalability to large-scale datasets. Many models focus solely on prediction accuracy without addressing the integration of meaningful clinical insights, limiting their practical applicability. Furthermore, the challenge of balancing highly imbalanced datasets and ensuring robustness across diverse patient populations remains inadequately addressed. This study bridges these gaps by proposing an interpretable and scalable framework, leveraging advanced graph-based neural networks and optimization techniques to improve sepsis outcome prediction.

Proposed methodology

This study proposes a comprehensive and clinically relevant methodology to predict sepsis outcomes using high-dimensional and imbalanced ICU data. The core objective is to develop an interpretable and accurate model that can capture subtle clinical signals associated with mortality risk. To achieve this, we begin by extracting and preprocessing data from the MIMIC-IV database to ensure clinical consistency. The class imbalance issue, common in mortality prediction, is addressed using SMOTE. Clinically significant features are selected using the Wolverine Optimization Algorithm (WoOA), which mimics adaptive survival strategies—suitable for the high complexity of ICU data. These features are structured into a patient similarity graph and processed using a self-supervised GPT-GNN, enabling the model to understand intricate inter-patient relationships. Finally, the model is fine-tuned with the TOTO algorithm to optimize prediction performance in a diverse and noisy healthcare setting. In this section, the prediction process is carried out using an advanced deep learning model with balanced input data, as illustrated graphically in Fig. 1.

The workflow of the proposed framework for sepsis risk stratification consists of the following key stages:

- *Data Extraction and Selection*: Patient records were extracted from the MIMIC-IV database using BigQuery, applying inclusion criteria based on ICU stay, age, and SOFA scores.
- *Data Preprocessing*: Missing values were imputed, and categorical variables were grouped. Race and antibiotic types were encoded based on clinical relevance. Outliers were removed.
- *Data Balancing*: To address class imbalance, SMOTE was applied, expanding the dataset and improving minority class representation.
- *Feature Selection*: WoOA was used to identify the most informative features. It mimics scavenging and hunting behaviors to iteratively optimize the feature subset.
- *Graph Construction*: Selected features were used to build an attributed graph where nodes represent patients and edges capture similarities based on clinical attributes.
- *Generative Pre-Training of GNN (GPT-GNN)*: The graph was used to pre-train a GNN using a self-supervised generative framework to capture latent structural and attribute-based patterns.
- *Fine-tuning with TOTO*: The pre-trained model was fine-tuned using the TOTO optimizer, which integrates six different search strategies for enhanced convergence.
- *Model Evaluation*: The performance was evaluated using standard metrics (Accuracy, AUC, Precision, Recall, MCC) and compared with baseline models.

Data source and inclusion criteria

An authoritative and comprehensive database, the Medical Information Mart for Intensive Care IV (MIMIC-IV)[22], provided the data used in this investigation. Over forty thousand distinct individuals from intensive care units are included in this database, which contains health records from the Beth Israel Medical Center spanning 2008 to 2019. Data from the admissions process, including demographics, laboratory findings, and intensive care unit details, were organized into multiple tables. This dataset offers a more up-to-date representation of patient care compared to MIMIC-III by increasing the breadth of data collected and incorporating updated patient information. Using MIMIC-IV ensures that this study is based on the most recent data available, enabling a more precise and relevant investigation into the factors impacting ICU patient outcomes.

The following criteria were used to select target patients. To provide sufficient data for a comprehensive study, the criteria specified that only patients aged 18 or older and with an ICU stay exceeding 24 h were eligible for inclusion. Patients with a Sequential Organ Failure Assessment (SOFA) score above a threshold and a presumed infection, as recorded in the MIMIC-IV database, were also included. The diagnosis was based on the Sepsis-3 criteria, the Third International Consensus Definition for Sepsis. BigQuery, a data extraction tool, was employed to identify the patients for analysis.

Pre-processing and balancing the data

The dataset was preprocessed as follows: (1) addressing missing values and duplicates in numerical and categorical data, and (2) creating new features to facilitate encoding by grouping categorical variables (e.g., antibiotics and race). For this study, racial categories were defined, and the original 25 antibiotic classifications were reorganized into seven groups based on chemical structure, mechanism of action, spectrum of activity, toxicity, and side effects. Categories such as aminoglycosides and carbapenems were included in these groups.

Healthcare datasets are often skewed, as confirmed during assessment. To address this imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was employed. Unlike the cluster centroids method used

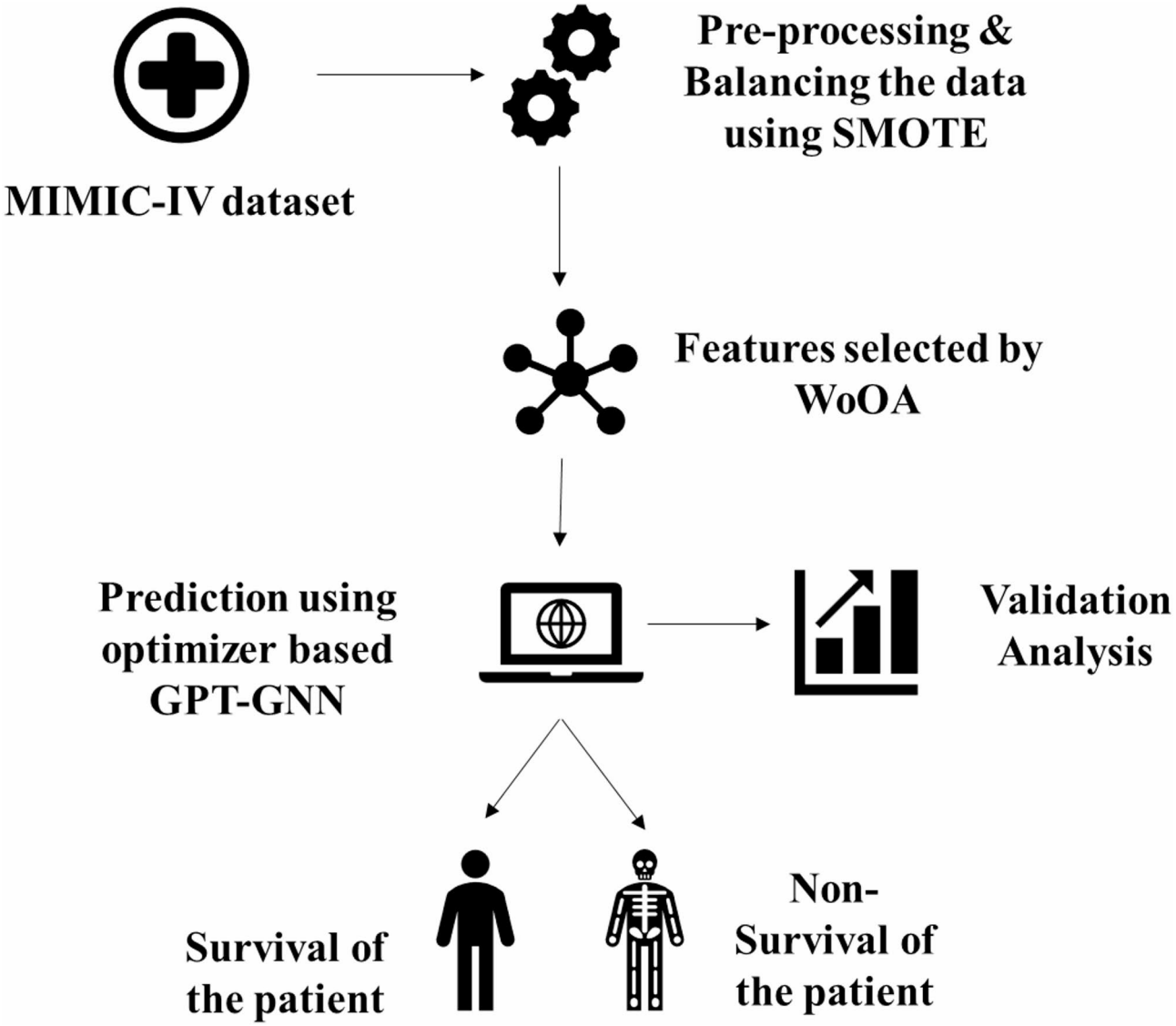

Fig. 1. Workflow of the research framework.

in prior research[23], SMOTE increases the number of data points for the minority class through oversampling. This approach enhances the model's ability to generalize to unseen data and reduces the risk of overfitting. By applying SMOTE, the dataset expanded from 6,401 to 7,304 data points, resulting in a balanced dataset with 53 columns and 7,304 entries. This balanced dataset improves the reliability and generalizability of predictive models.

The use of SMOTE is crucial in addressing the inherent class imbalance present in the sepsis dataset, where mortality cases are fewer than survival cases. Traditional classifiers trained on such imbalanced data often become biased toward the majority class, resulting in poor sensitivity and recall for the minority (deceased) class. SMOTE enhances the representativeness of the minority class by generating synthetic examples using k-nearest neighbors, thereby creating a more balanced class distribution. This not only improves the classifier's ability to recognize patterns indicative of sepsis mortality but also enhances evaluation metrics such as AUC, F1-score, and recall, which are critical for clinical decision-making in high-stakes scenarios.

Clinical insights were incorporated during preprocessing by grouping antibiotic categories and demographic variables based on medical relevance. SMOTE was applied to balance mortality classes, improving the model's ability to generalize and enhancing recall for minority cases.

Feature selection using wolverine optimization algorithm (WoOA)

Feature selection was informed by a clinical expert's suggestions and a comprehensive literature review. The selection process considered two main factors: (1) elements identified as critical for mortality prediction in previous studies, and (2) features frequently cited in studies, indicating their widespread use in critical care. To select features from the balanced dataset, this study employed the Wolverine Optimization Algorithm (WoOA)[24]. Inspired by the natural behaviors and strategies of wolverines, WoOA leverages scavenging and

hunting techniques for optimization. WoOA selected features based on both statistical significance and clinical relevance. Variables such as SOFA score, GCS score, urine output, and glucose were prioritized, ensuring that the model's predictions are both accurate and interpretable for clinicians. The study also provides a detailed outline of the mathematical modeling approach used to translate these biological principles into computational methods suitable for various optimization problems.

Algorithm initialization

WoOA is an evolutionary procedure inspired by wolverine populations. Through an iterative process, WoOA generates effective solutions for optimization tasks by utilizing individual members to explore the solution space. Each wolverine represents a decision variable and serves as a potential solution. Mathematically, the problem is represented as a vector, and the population of wolverines forms a collective behavior described by a matrix, as expressed in Eq. (1). During algorithm initialization, Eq. (2) is used to randomly position the wolverines within the problem-solving space.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \iddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N \times m} \tag{1}$$

$$x_{i,d} = lb_d + r.\,(ub_d - lb_d) \tag{2}$$

In this scenario, X signifies the populace matrix of WoOA, Xi means the ith wolverine, $x_{i,d}$ signifies its value in the dth search space. The variable N designates the sum of wolverines, while m characterises the total quantity variables. On, r is a haphazard sum in the interval [0, 1], lb_d, and ub_d denote a decision variable, correspondingly.

This theory states that every wolverine's placement inside the problem-solving area represents a potential answer to the situation at hand. In light of this knowledge, the objective function of the problem may be better evaluated with the help of the recommended values of the decision variables for each wolverine. A vector representation would be an excellent way to display the evaluated function.

$$F = \begin{bmatrix} F_1 \\ ? \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N \times 1} \tag{3}$$

where F is the vector of the assessed standards and F_i is the assessed function charge based on ith wolverine.

It can tell a lot about the solutions' quality from the evaluated values function. Each member of the population solution; the one with the highest assessed value of the objective function represents the best possible answer, and the one with the lowest evaluated value corresponds to the worst possible solution. Every time WoOA runs, the goal function is revaluated for every wolverine since their position in updated. So, it is important to compare the assessed function in each iteration and update the top separate in the population periodically.

Mathematical modelling of WoOA

The WoOA method is based on mimicking the wolverine's natural feeding behaviour in order to update the population's position in space. A wolverine's diet consists of two main components: hunting and scavenging. The scavenging approach is used by wolverines to feast on dead animals by following the tracks of other predators. Wolverines use a specific hunting method in which they ambush their target, engage in a struggle with it, and then consume its flesh. It is assumed in WoOA's architecture that each wolverine has an equal chance of randomly selecting one of these two tactics in each iteration, and its solving space is updated based on the simulation of the strategy it has chosen. The wolverine uses Eq. (4) to model its decision-making process in order to choose between hunting and scavenging. This implies that the first or second approach is the only determinant of how each wolverine's in each iteration.

$$Update\ process\ for\ i'\ th\ wolverine\ X_i : \begin{cases} based\ on\ scavenging\ strstegy, & r_p \leq 0.5 \\ based\ on\ hunting\ statergy, & else \end{cases} \tag{4}$$

Here, r_p is a random sum from the intermission [0, 1].

Strategy 1: scavenging strategy (exploration stage)

By mimicking the wolverine's normal feeding behaviour on carrion, the first WoOA technique updates the populace's location in the space. In this tactic, the wolverine tracks the tracks of other predators to find their leftover carrion. One way to approach exploring the solution space is by reaction to predators. By drastically shifting the population's places, this method changes the members' paths through the problem-solving domain. The algorithm accomplishes a more thorough investigation of possible solutions by utilising this approach. The search area is expanded and the algorithm's ability to find various and optimal answers is enhanced by these thorough positional alterations. As a result, the algorithm is better equipped to conduct comprehensive global

searches, which increases its capacity to handle large problem spaces and find novel solutions. One part of the scavenging method is an exploring phase that mimics the way a wolverine finds and consumes carrion.

According to Eq. (5), in the WoOA design, the predators' position is taken into account as the superior value, with the aim of releasing the remaining kills for each wolverine.

$$CP_i = \{X_k : F_k < F_i \; and \; k \neq i\}, \; where \; i = 1,2,\ldots,N \; and \; k \in \{1,2,.,N\} \tag{5}$$

In the project of WoOA, it is supposed that the wolverine selects the site of a marauder from the CP_i goes in the direction of it after being set at random. This is a simplified model of the wolverine's natural behaviour as it searches for prey and approaches the spot where the prey dropped food. A new proposed location for the corresponding member has been calculated using Eq. (6) as the wolverine approaches the chosen predator in pursuit of the carrion. According to Eq. (7), this proposed location takes precedence over the previous one of the matching member if the value of the goal function is improved.

$$x_{i,j}^{S1} = x_{i,j} + r_{i,j}.(SP_{i,j} - I_{i,j}.x_{i,j}) \tag{6}$$

$$X_i = \begin{cases} X_i^{s1}, & F_i^{s1} \leq F_i \\ X_i, & else \end{cases} \tag{7}$$

where SP_i describes the selected wolverine, while $SP_{i,j}$ signifies the jth predator. X_i^{s1} refers to the ith wolverine's recently intended location using the scavenging strategy used in the suggested WoOA. Similarly, $x_{i,j}^{S1}$ the jth dimension. The term F_i^{s1} designates the value associated with the new position. Additionally, $r_{i,j}$ are random statistics from the intermission [0, 1], besides $I_{i,j}$ are statistics 1 or 2.

Strategy 2: hunting strategy (exploration and exploitation phases)

As for WoOA's second tactic, it mimics the wolverine's natural hunting behaviour to update the population's location in the space. This behaviour is similar to the way wolverines usually hunt: they assault live prey, then chase after them, fight them, and finally get their target before eating it. Two parts make up the populace's update hunting strategy: (i) exploration, when members mimic the wolverine's approach to prey, and (ii) exploitation, wherein members mimic the wolverine's fight-or-chase with its prey.

- Phase 1: Attack (exploration stage)

This part of the WoOA involves mimicking a wolverine's attack on its prey in order to move the population about in the space. Imitating the wolverine's motions during a hunt causes the population's position to alter significantly, which improves WoOA's exploring capabilities in the problem-solving area. The optimal population member's location is compared to the prey's site when designing the WoOA. A diagram depicting the typical actions of a wolverine as it attacks its prey. A new possible position is calculated using Eq. (8) by modelling the wolverine's tactic to the best member (the "prey") for each WoOA member. Then, according to Eq. (9), this new suggested site takes the place of the old one for the relevant member if the objective function shows an development.

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j}.\left(Prey_j - I_{i,j}.x_{i,j}\right) \tag{8}$$

$$X_i = \begin{cases} X_i^{P1}, F_i^{P1} \leq F_i \\ X_i, \quad else \end{cases} \tag{9}$$

where X_i^{P1} represents the newly computed established by the first phase of policy in projected WoOA, besides Prey is the best populace member as prey, with $x_{i,j}^{P1}$ being its jth dimension. Likewise, $x_{i,j}^{P1}$ refers to this new position's jth dimension. The function value of location is indicated by F_i^{P1}. Furthermore, $I_{i,j}$are binary numbers that are randomly assigned as either 1 or 2, and $r_{i,j}$ are random standards tested from the intermission [0,1].

- Phase 2: Fighting and chasing (exploitation stage)

In this stage of WoOA, the wolverine's hunt-and-pursue behavior is mimicked to move the population members in the solution space. To enhance WoOA's local search capabilities within the problem-solving area, the algorithm simulates the wolverine's motions during the chase and makes minor adjustments to the population's positions. These interactions are designed to occur near the hunting spot, as specified by the WoOA framework.

A diagram illustrates the typical wolverine behavior while pursuing prey. Each member of WoOA determines its new position using Eq. (10), which is based on a model of the hunt and pursuit behavior. According to Eq. (11), the member's previous position is replaced with the new one if it improves the value of the objective function.

$$x_{i,j}^{P2} = x_{i,j} + (1 - 2r_{i,j}).\frac{ub_j - lb_j}{t} \tag{10}$$

$$X_i = \begin{cases} X_i^{P2}, F_i^{P2} \leq F_i \\ X_i, \quad else \end{cases} \tag{11}$$

where X_i^{P2} represents the new location for the ith wolverine that was established in the anticipated WoOA based on the second policy. $x_{i,j}^{P2}$ represents the updated position. F_i^{P2} shows the charge of the objective

function connected to the ith new location. Furthermore, $r_{i,j}$ are random figures tested from the intermission [0, 1], and t signifies the repetition counter.

Repetition process of WoOA

The initial version of WoOA is complete after repositioning all wolverines in the problem-solving area. The method then proceeds to the next iteration using the modified values, making incremental adjustments to the wolverines' positions until the final iteration, which is governed by Eq. (4) to (11). In each iteration, the solution is updated by comparing evaluated values for the objective function. At the end of the procedure's runtime, the WoOA solution to the given problem is identified as the best solution found across all iterations.

The final dataset contains 38 unique features, including demographics, antibiotic use, medical history, and laboratory results. Factors such as respiration rate, glucose levels, sodium levels, average urine output, and the Sequential Organ Failure Assessment (SOFA) score are included as key variables. These variables were selected based on their proven predictive value for patient outcomes in relevant research. Incorporating these features into the dataset enhances the foundation for developing predictive models aimed at improving the accuracy of mortality and prognosis projections in critical care. A 200-point cutoff for the PaO2/FiO2 ratio was established to further refine the dataset. Patients with Glasgow Coma Scale (GCS) scores of 8 or below were classified as being in a coma, as suggested by clinical experts. After the feature selection process, the dataset was reduced to 6,401 records.

Classification using generative pre-training of GNNs

Here, to present the GPT-GNN for the prediction process and formalise the attributed graph generation problem.

The GNN pre-training problem

The credited graph G = (v, ε,x), where v besides ε represent the sets of nodes besides edges, and x represents the matrix of node features, is often features chosen to be fed to GNNs. A supervised downstream task, such node classification, teaches a GNN model to produce node representations. Most GNNs want adequate specialised labelled data for every task, which can be a problem when there are numerous tasks on a single graph. The training of generalized GNN is impeded when enough annotations are not readily available, especially for large-scale graphs. So, a pre-trained GNN classical that is able to generalise with few labels would be ideal.

Theoretically, this model should do two things: 1) benefit different downstream activities on this graph by capturing its internal capture these patterns. Our formal objective for GNN pre-training is to learn a generic GNN model.

f_θ purely graph $G = (v, \epsilon, x)$ in the absence of labelled data, in a way that f_θ serves as an effective starting point for different (hidden) subsequent graph, or graphs within the similar domain. A reasonable concern that emerges in this context is how to construct an unsupervised graph to pre-train the GNN model in the absence of labelled data on the graph.

The generative pre-training framework

A model that can capture the distribution of data can transfer onto many downstream tasks, thanks to recent breakthroughs in self-supervised learning, which have revealed that unlabelled data itself includes significant semantic knowledge. Our proposed GPT-GNN takes this idea a step further by pre-training a GNN using the structure and properties of the input graph that were reconstructed or generated.

Formally, given a graph G=(v,ε,x) and a GNN prototype f_θ, this graph is optimized by the GNN as $p(G, \theta)$, where nodes in G are associated. GPT-GNN aims to optimize the GNN model by exploiting the graph likelihood, i.e., $\theta^* = max_\theta\, p(G, \theta)$. Then, model $p(G, \theta)$.

It should be noted that most current methods for generating graphs use an auto-regressive approach to factorize the objective. This means that the nodes in the network are added in a specific order, created by linking each new node to an existing one. Similarly, the node ordering can be inferred from a permutation vector σ by examining the value of the node ID at the ith position in the vector. Therefore, the expected value over all feasible permutations is equal to the graph distribution p(G,).

$$p(G, \theta) = \mathbb{E}_\pi\, [p_\theta\, (X^\pi, E^\pi)] \tag{12}$$

where $X^\pi \in R^{|V| \times d}$ represents all edges connected with node , represents a collection of edges, and represents permutated node characteristics. The generating process for one permutation will be shown in the next sections without the subscript , assuming for simplicity that the probability of observing any node ordering π is equal. The log-likelihood can be factorized autoregressively, with one node generated per iteration, given a permuted order:

$$logp_\theta\, (X, E) = \sum\nolimits_{i=1}^{|v|} logp_\theta\, (X_i, E_i | X_{<i}, E_{<i}) \tag{13}$$

At each step $_i$, to use altogether nodes are shaped qualities $X_{<i}$, besides the construction (edges) among these nodes $E_{<i}$ to produce a novel node , counting attribute X_i besides its influence's nodes E_i.

Equation 13's goal essentially explains how an attributed graph is generated by an autoregressive process. So, how can to represent the conditional probability $p_\theta\, (X_i, E_i | X_{<i}, E_{<i})$?

Factorizing attributed graph generation

To calculate $p_\theta\ (X_i, E_i|X_{<i}, E_{<i})$, unique naive answer could be to only shoulder that X_i besides E_i are self-governing, that is,

$$p_\theta\ (X_i, E_i|X_{<i}, E_{<i}) = p_\theta\ (X_i|X_{<i}, E_{<i})\ .p_\theta\ (E_i|X_{<i}, E_{<i}) \tag{14}$$

This decomposition treats each node independently, ignoring the interdependence of its properties and connections. However, attributed graphs and GNN convolutional aggregation fundamentally rely on these dependencies.

Therefore, pre-training GNNs cannot be effectively informed by such a simplistic decomposition. To address this limitation, we propose a dependency-aware factorization method for generating attributed graphs.

Specifically, when estimating a node's attributes, we utilize knowledge about its structure, and vice versa. At this point in the process, a portion of the edges has already been observed (or created). Thus, the generation process can be divided into two linked components:

1. Creating node attributes and their associated edges
2. Creating the remaining edges using the observed edges and the previously generated node attributes

This approach allows the model to capture the relationship between each node's structure and attributes effectively.

Formally, we define a variable to represent the vector of all observed edges within E_i. Thus, $E_{i,0}$ highlights the edges that have been observed. Additionally, the index of all masked edges that need to be created is represented by ¬o. The conditional probability can now be expressed as the expectation over all observed edges using this factorization.

$$\begin{aligned} & p_\theta\ (X_i, E_i|X_{<i}, E_{<i}) \\ &= \sum_0 p_\theta\ (X_i, E_{i,\neg o}|E_{i,o}, X_{<i}, E_{<i})\ .p_\theta\ (E_{i,o}|X_{<i}, E_{<i}) \\ &= \mathbb{E}_{\mathrm{o}}\left[\mathrm{p}_\theta\ (\mathrm{X_i}, \mathrm{E_{i,\neg o}}|\mathrm{E_{i,o}}, \mathrm{X_{<i}}, \mathrm{E_{<i}})\right] \\ &= \mathbb{E}_{\mathrm{o}}\left[\underbrace{p_\theta\ (X_i, E_{i,o}|X_{<i}, E_{<i})}_{1)\ \mathrm{generate\ \ attributes}} \cdot \underbrace{p_\theta\ (E_{i,\neg o}|E_{i,o}, X_{<i}, E_{<i})}_{2)\ \mathrm{generate\ \ edges}}\right] \end{aligned} \tag{15}$$

This factorization project is talented to classical the addiction among node 's characteristics X_i and its related connections E_i. The term $p_\theta\ (X_i, E_{i,o}|X_{<i}, E_{<i})$ represents the cohort of attributes for edges $E_{i,o}$, to gather the info to make its attributes X_i. The $p_\theta\ (E_{i,\neg\ o}|E_{i,o}, X_{<i}, E_{<i})$ signifies the edges. Based experimental edges $E_{i,o}$ besides the generated attributes X_i, To create node 's representation and forecast within $E_{i,\neg\ o}$ exists.

Efficient attribute and edge generation

To maximise efficiency, it is preferable to execute the GNN in order to calculate the loss of attribute generations. And to plan to generate both attributes and edges at the same time. Attribute generation may get unauthorised access to the node attributes used for edge generation. Our architecture separates each node into two kinds to prevent information loss:

- Cohort Nodes for attributes. To learn a shared vector $\mathrm{X_{init}}$ to characterise $X_{<i}, E_{<i}$ and nodes by substituting a dummy token for their attributes. In the masked language model, this is comparable to the [Mask] token trick.
- Nodes for Edge Generation. To retain these nodes' attributes and use GNN's input.

The adapted GNN is then applied to produce the output representations. The embedded outputs of the Attribute Node and the Edge Generation Node are represented by h^{Attr} and h^{Edge}, respectively. Because Nodes' attributes are hidden, h^{Attr} in over-all comprises less info than h^{Edge}. Therefore, when behaviour the GNN message Generation Nodes' output h^{Edge} on the outside. Then, using the two sets of node representations, attributes and edges are generated using separate decoders.

For Attribute Cohort, to signify its decoder as $Dec^{\mathrm{Attr}}(\cdot)$, which receipts h^{Attr} as input besides generates attributes. The modeling decision depends on the type of attributes. For example:

- If a node's input attribute is text, text generation can be performed using a text generator model like LSTM.
- If the input attribute is a standard vector, it can be generated using a multi-layer perceptron (MLP).

As a measure between the produced and real attributes, to also build a distance function, such L2-distance for vectors or perplexity for text. Therefore, the attribute generation loss is determined by:

$$\mathcal{L}_i^{Attr} = Distance(Dec^{Attr}\left(h_i^{Attr}\right), X_i) \tag{16}$$

By minimizing the distance among the produced and disguised characteristics, it is equal to characteristic, i.e., $p_\theta\ (X_i, E_{i,o}|X_{<i}, E_{<i})$, besides thus the retrained classical can graph.

In order to factorise the likelihood of Edge Generation, to undertake that each edge's generation is independent of the others.

$$p_\theta\ (E_{i,\neg\, o}|E_{i,o}, X_{\leq\, i}, E_{<i}) = \prod\nolimits_{j^+\in\, E_{i,\neg\, o}} p_\theta\ \left(j^+|E_{i,o}, X_{\leq\, i}, E_{<i}\right) \tag{17}$$

Next, after getting node illustration h^{Edge}, also model the likelihood connected with node by $Dec^{Edge}(h_i^{Edge},\ h_j^{Edge})$, where Dec^{Edge} is a function. Lastly, to adopt approximation to compute the node j^+. To prepare altogether the separate nodes as S_i^- besides compute loss via

$$\mathcal{L}_i^{Edge} = -\sum\nolimits_{j^+\in\, E_{i,\neg\, o}} log \frac{exp\left(Dec^{Edge}\left(h_i^{Edge}, h_{j^+}^{Edge}\right)\right)}{\sum\nolimits_{j\in\, S_i^-\cup\,\{j^+\}} exp\left(Dec^{Edge}\left(h_i^{Edge}, h_j^{Edge}\right)\right)} \tag{18}$$

By optimizing $\mathcal{L}^{Edge}$, The pre-trained model is graph's inherent construction because it is equivalent to optimising the probability of producing every edge.

Fine-tuning of the GPT-GNN using TOTO

This work employs TOTO, a swarm-based metaheuristic that utilizes different methods in each iteration to refine the suggested classifier[25]. Each strategy within TOTO has its advantages and limitations, and the multi-strategy approach aims to mitigate the latter while maximizing the former. In each cycle, every agent performs six searches to execute the various strategies. This means that each agent is required to complete all six searches. Unlike other metaheuristics that advocate population-level role separation, such as RDA, KMA, or SKA, TOTO does not adhere to the principle of task segregation.

Among the six searches, three are directed, and the other three are random. Each search generates a candidate solution. Only one of these six candidates is compared to the agent's current solution, and the best candidate is selected. If this candidate outperforms the agent's current solution, it replaces the existing one. Unlike metaheuristics such as DTBO, GPA, or NGO, which conduct their searches sequentially, TOTO evaluates all candidates concurrently. After evaluation, the best candidate is compared with the current global best solution. The global best is updated only if the new candidate is superior.

In the three guided searches, the global best solution and a randomly chosen agent serve as references. The first guided search generates a candidate by tracking the agent's progress toward the global best solution. The second search generates a candidate by considering the global best path but excluding the corresponding agent's current solution. The third guided search generates a candidate based on the relative movement of the corresponding agent toward the randomly chosen agent. If the corresponding agent does not move toward the random agent, the candidate is generated based on the agent's movement away from the random agent.

For the three random searches, the first two iterations are performed within the local search space, while the third does not require a local search space. The width of the local search space decreases linearly with the number of iterations. In the first random search, a candidate is generated within each agent's local search space. The second random search generates a candidate using the search space associated with the global solution. The third random search generates a candidate from the global search space without restrictions.

The integration of the TOTO algorithm with the GPT-GNN framework serves to refine the model's performance by enhancing the optimization process during fine-tuning. Graph-based models like GPT-GNN often involve high-dimensional and non-convex optimization challenges, where traditional single-strategy optimizers may get trapped in local minima. TOTO addresses this by executing six diverse search strategies in each iteration—three guided and three random—allowing it to thoroughly explore the solution space and exploit high-quality regions effectively.

This section summarizes the reasoning behind the proposed strategy as follows:

- *Multiple Search Strategies*: Metaheuristics using multiple search strategies have been shown to outperform those relying on a single strategy.
- *Balancing Exploration and Exploitation*: Combining directed and random searches aims to strike a balance between exploring new solutions and exploiting known ones.
- *Multiple References*: Using multiple references enhances the search capability and prevents reliance on a single reference point.
- *Rigorous Acceptance-Rejection*: A strict acceptance-rejection method is used to prevent the search process from converging on suboptimal solutions or locations.

Finally, the TOTO algorithm is developed based on this metaheuristic approach. Method 1 introduces this algorithm, and the associated process is formalized using equations (19) to (29). This publication employs the following annotations.

$$a = U(a_l, a_u) \tag{19}$$

$$a'_{best} = \begin{cases} a, f\,(a) < f\,(a_{best}) \\ a_{best},\ \ else \end{cases} \tag{20}$$

$$c_1 = a + r_1(a_{best} - r_2 a) \tag{21}$$

$$c_2 = a_{best} + r_1 \left(a_{best} - r_2 a\right) \tag{22}$$

$$a_{sel} = U\left(A\right) \tag{23}$$

$$c_3 = \begin{cases} a + r_1 \left(a_{sel} - r_2 a\right), f\left(a_{sel}\right) < f\left(a\right) \\ a + r_1 \left(a - r_2 a_{sel}\right), \ else \end{cases} \tag{24}$$

$$c_4 = a + r_3 \left(1 - \frac{t}{t_m}\right) \left(a_u - a_l\right) \tag{25}$$

$$c_5 = a_{best} + r_3 \left(1 - \frac{t}{t_m}\right) \left(a_u - a_l\right) \tag{26}$$

$$c_6 = a_l + r_1 (a_u - a_l) \tag{27}$$

$$c_{sel} = c \in \ C, \min\left(f(\,c\,)\right) \tag{28}$$

$$a' = \begin{cases} c_{sel}, f\left(c_{sel}\right) < f\left(a\right) \\ a, \qquad\qquad else \end{cases} \tag{29}$$

Algorithm 1: Chief procedure of TOTO

```
1 Begin
2 for all a in A
3 generate a using (19)
4 update a_best using (20)
5 end for
6 for t = 1 to t_max
7 for all a in A
8 first guided search using (21)
9 second guided search using (22)
10 third guided search using (23) and (24)
11 first random search using (25)
12 second random search using (26)
13 third random search using (27)
14 choose c_sel using (28)
15 update a using (29)
16 update a_best using (20)
17 ends for
18 ends for
19 end
```

Where, a is agent, A set of agents, a_{best} is global best agent, a_{sel} is arbitrarily selected agent a_{l} is lower boundary, a_{u} is upper boundary, c_1 is first candidate, c_2 is second candidate, c_3 is third candidate, c_4 is fourth candidate, c_5 is fifth candidate, c_6 is sixth candidate, c_{sel} is designated candidate, f is fitness function, r_1 is real uniform sum among 0 and 1, r_2 is integer uniform random sum among 1 besides 2, r_3 is random sum among −1 and 1, t is iteration, $t_{\max}$ is maximum iteration and u unchanging random.

The following provides an explanation of equations (19) through (29): The initial solution is determined through randomization within the solution space, as defined in Eq. (19). The global best value is updated based on the novel value of the associated agent, as stated in Eq. (20). If the associated agent outperforms the current global best solution, it replaces the global best. The candidate for the first guided search is determined by the agent's progress toward the global best solution, as described in Eq. (21). The candidate for the second guided search is determined by the global best solution's movement while avoiding the associated agent, as per Eq. (22). A random agent is selected from the population, as outlined in Eq. (23). The candidate for the third guided search is constructed based on the associated agent's movement relative to a randomly chosen agent, according to Eq. (24). The first random search candidate is generated by exploring the neighborhood of the associated agent, as described in Eq. (25). The candidate for the second random search is produced based on the neighborhood search's best solution, as per Eq. (26). The third random search candidate is obtained by conducting a search in a constant random space, as explained in Eq. (27). The optimal candidate is selected from these six options and compared to the associated solution, as stated in Eq. (28). If this candidate is superior to the associated solution, it replaces the solution, as defined in Eq. (29).

Theoretical comparison with existing methods

Traditional machine learning methods like SVM, XGBoost, and LightGBM treat patient records as independent feature vectors, lacking the capacity to model relationships between patients—a limitation in sepsis risk prediction where temporal and contextual factors are vital. In contrast, standard GNNs use graph structures to

model relationships but typically require large amounts of labeled data and suffer from limited generalization across patient subgroups.

Our proposed GPT-GNN framework differs by adopting a generative, self-supervised approach that pre-trains the model to learn latent graph representations without labeled data. This enables better initialization and generalization for downstream tasks. Additionally, the Wolverine Optimization Algorithm (WoOA) uniquely integrates clinical domain knowledge during feature selection through a biologically inspired search mechanism, improving model interpretability—unlike brute-force or filter-based methods.

The inclusion of TOTO fine-tuning introduces another layer of novelty, as it combines multiple guided and random search strategies to escape local optima—outperforming conventional optimizers that rely on a single update rule. Together, these three components form a synergistic pipeline that addresses real-world challenges such as class imbalance, sparse label availability, and clinical insight integration—setting this work apart from existing methods in both design and capability.

Results and discussion

The research was conducted using an Intel Core i5-7200 processor with 8 GB of internal memory. The processor operates at a clock speed of 2.7 GHz. A dedicated User Interface (UI) and a Notebook (Python 3.7) environment were used to perform the operations on Windows 10, a 64-bit operating system[26]. Figure 2 illustrates the training and testing accuracy of the proposed classifier.

Validation analysis of proposed feature WoOA

Figures 3 and 4 provide a graphical analysis of the proposed Feature Selection (FS) method compared with existing optimizers in terms of various metrics.

The performance of various optimization-based models is evaluated across multiple metrics. The WoOA model demonstrates superior performance, achieving the highest values in most metrics, including Accuracy (98.28%), Precision (98.32%), Recall (98.10%), Specificity (97.67%), and F1-Score (98.18%). It also achieves outstanding AUC-ROC (99.88%) and AUC-PR (98.71%), along with a strong MCC (98.09%). The Butterfly Optimizer follows closely, with a high Accuracy of 98.05% and comparable Precision (97.99%), Recall (97.95%), and F1-Score (97.96%), though it exhibits slightly lower Specificity (96.47%) and MCC (97.83%). The Wolf Optimizer shows moderate performance, with an Accuracy of 91.86%, strong AUC-ROC (99.56%), and AUC-PR (96.60%), but relatively lower Specificity (90.14%) and MCC (90.95%). The Ant Lion Optimizer records the lowest performance, with an Accuracy of 89.33%, a lower Recall (88.93%), and F1-Score (89.04%), though it

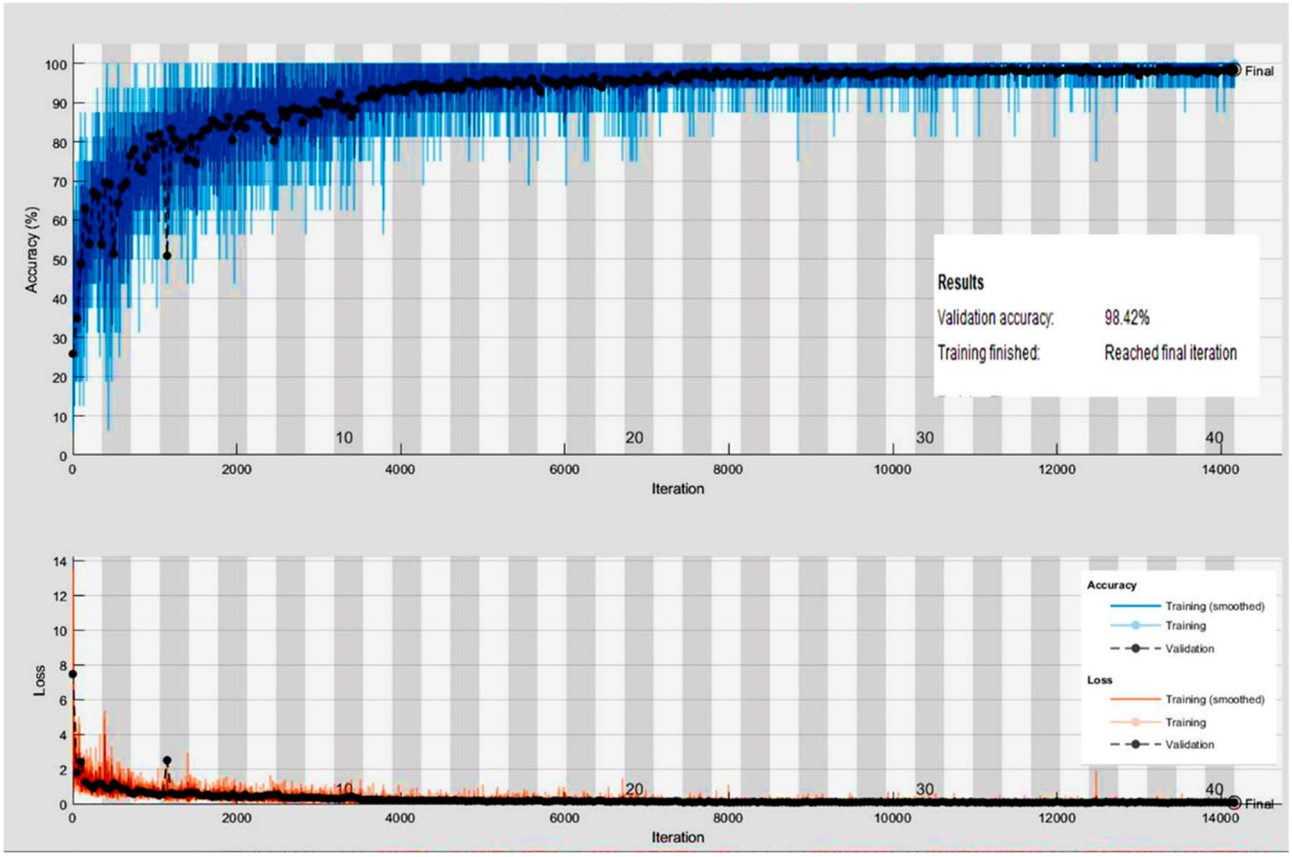

Fig. 2. Training curve for the proposed model.

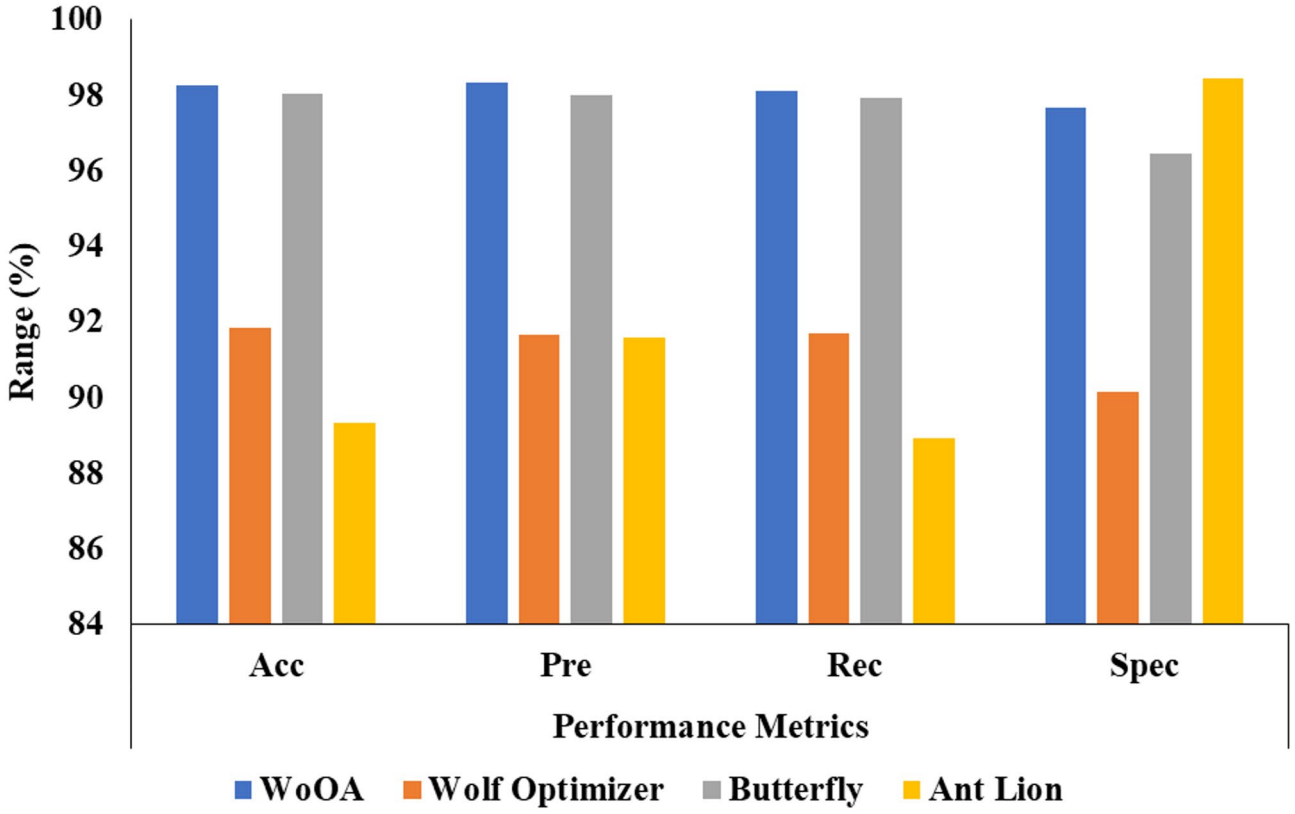

Fig. 3. Visual analysis of various algorithms.

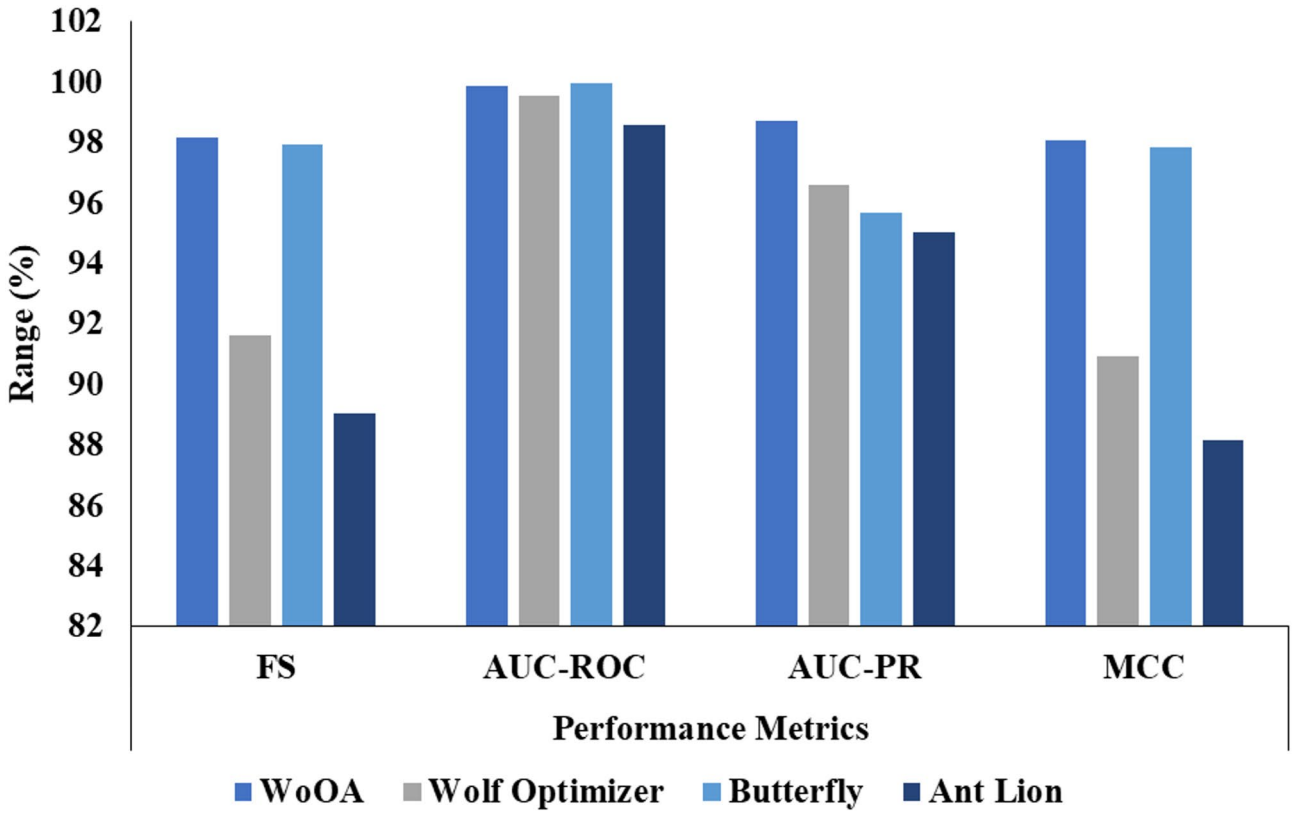

Fig. 4. Graphical analysis of different models.

achieves high Specificity (98.46%) and a good AUC-ROC (98.59%). Overall, the WoOA model outperforms the other models across most parameters.

Validation analysis of proposed classifier

Figures 5 and 6 present the visual representation of the proposed classifier compared with existing models across various metrics.

The performance of different models is compared across multiple metrics, showcasing the proposed model as the most effective among the four. The proposed model achieves the highest Accuracy (98.08%), Precision (99.06%), Recall (99.05%), and MCC (98.98%), along with an impressive F1-Score (96.05%), AUC-ROC (99.93%), and AUC-PR (99.43%). XGBoost follows with an Accuracy of 95.54%, high Precision (96.57%), Recall (96.55%), and a strong MCC (97.49%), indicating robust predictive capability. LightGBM performs slightly lower, achieving an Accuracy of 93.77%, Precision of 94.78%, and an MCC of 96.75%, while maintaining good AUC-PR (98.99%) and AUC-ROC (97.99%). SVM records the lowest performance, with an Accuracy of 91.63%, F1-Score of 90.70%, and MCC of 95.73%, though it maintains respectable Specificity (94.70%). Overall, the proposed model demonstrates superior performance across all metrics, significantly outperforming the other models in Precision and Recall. High recall (99.05%) and AUC (99.93%) confirm the success of SMOTE in handling class imbalance. The selected features align with known clinical indicators, demonstrating the model's practical relevance and interpretability.

Computational complexity of WoOA

A good indicator of the efficiency of the Wolverine Optimization Algorithm (WoOA) in solving optimization problems is the computational difficulty of the task. This section examines the challenges WoOA faces and evaluates its performance compared to other metaheuristic algorithms in this research.

Determinants of Computational Difficulty:

Number of Candidates (N): A population of NNN potential solutions is initialized by WoOA in each iteration.

- The algorithm's efficiency and rate of convergence are strongly influenced by the size of this population.

Number of Decision Variables (m): Assuming mmm decision variables, the difficulty of evaluating each potential solution is proportional to their number.

- The time required to evaluate potential solutions increases exponentially with the growth of the search space as m increases.

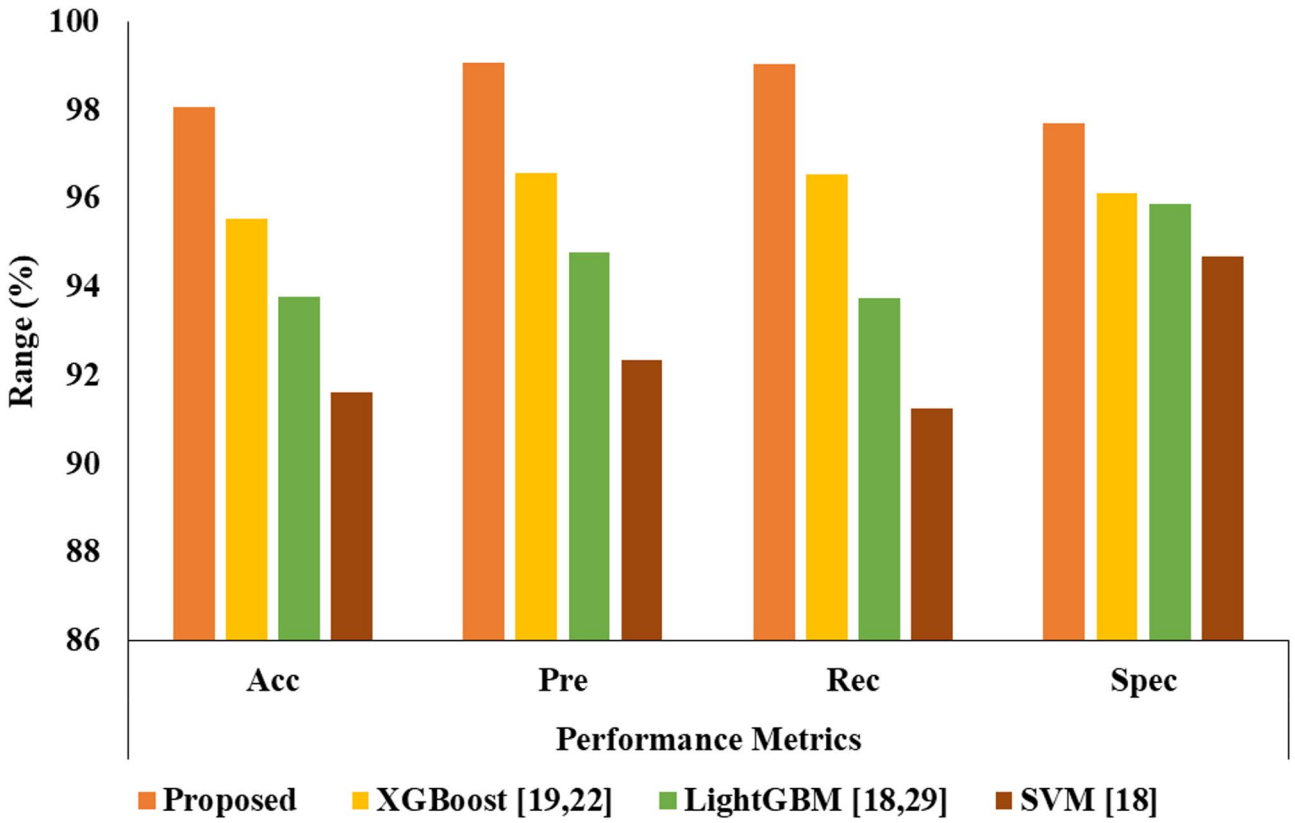

Fig. 5. Graphical comparison of various classifiers.

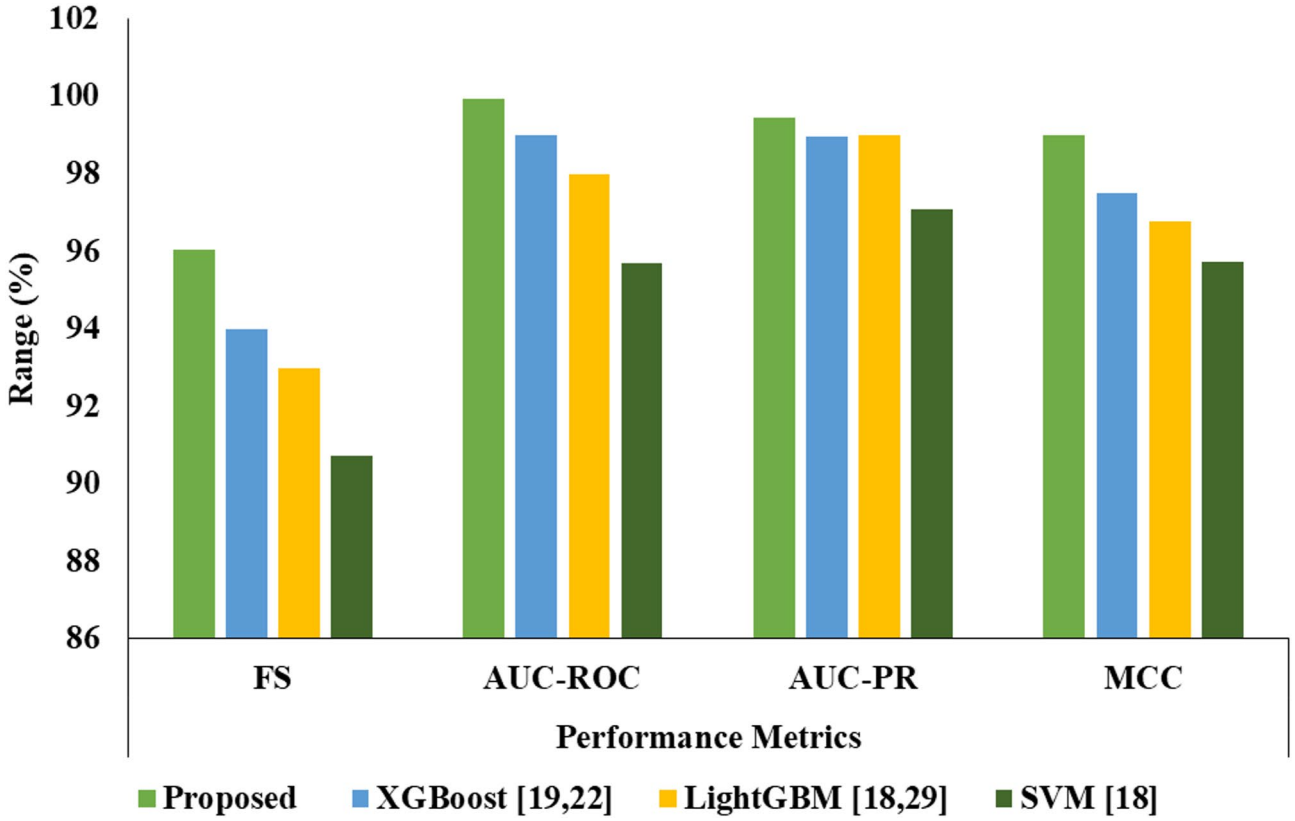

Fig. 6. Visual representation of different models.

Iteration Count (T): Over T iterations, WoOA continuously improves its solutions.

- The number of iterations impacts the extent to which the algorithm explores and exploits the search space.

Objective Function Evaluation (O(f)): The evaluation of the objective function f determines which of several potential solutions is the most fit.

- The mathematical formulation and constraints of f significantly influence the computational effort required for O(f).

Time complexity analysis

Assuming a population size of N, the temporal complexity of WoOA iterations can be roughly expressed as $O(N \cdot O(f))$.

- For each possible solution, effort to assess the objective function is denoted as O(f).
- Therefore, total time difficulty of WoOA is $O(T \cdot N \cdot O(f))$.
- This formalism specifies how the procedure changes depending on the size of the populace, the sum of repetitions, and the computational cost of evaluating the objective function.

Ablation study

To evaluate the impact of individual components in the proposed framework, an ablation study was performed under three settings:

Model A: Without WoOA-based feature selection (using all original features).

Model B: Without GPT-GNN pre-training (replaced by a standard GCN).

Model C: Without TOTO fine-tuning (standard training post GNN).

The results in Table 1 show the comparative performance of each variant using key metrics.

These findings confirm that each component—WoOA, GPT-GNN, and TOTO—contributes meaningfully to the model's superior performance, especially in terms of AUC and F1-Score.

Conclusion

This study demonstrates substantial progress in predicting sepsis outcomes through the integration of an optimized graph-based CNN model and advanced data preparation techniques. Key contributions include

Model variant	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC (%)
Proposed Model	98.08	99.06	99.05	96.05	99.93
Model A (No WoOA)	95.21	95.77	94.82	95.29	96.88
Model B (No GPT-GNN)	93.87	94.12	93.50	93.80	94.32
Model C (No TOTO)	94.65	94.92	94.10	94.51	95.77

Table 1. Ablation study evaluating the impact of WoOA, GPT-GNN, and TOTO on model performance.

the application of grouping and balancing techniques on the MIMIC-IV database and the utilization of the Wolverine Optimization Algorithm (WoOA) for effective feature selection. The TOTO algorithm fine-tunes the proposed model, significantly enhancing classification accuracy. The model's efficiency is evident in its ability to deliver highly accurate predictions using a minimal set of features. A robust AUROC score, combined with improved stability indicated by a reduced confidence interval, highlights its predictive power. The analysis underscores critical factors such as female gender, reduced average urine output, and higher coma scores as significant indicators of sepsis-related mortality. These insights enable clinicians to adopt preventive measures, thereby reducing mortality rates.

Future work

While the current study leverages the MIMIC-IV database effectively, a major limitation lies in the lack of additional datasets for external validation. Future research will address this by testing the model on diverse, publicly available datasets to assess its generalizability across different populations. Enhancing the interpretability of the model using explainable AI (XAI) techniques will also be a focus to foster greater clinical adoption. Moreover, exploring real-time implementation of the proposed model in healthcare settings, integrating wearable devices, and remote monitoring systems will extend its practical utility. Finally, the scalability of the model for other critical care prediction tasks beyond sepsis, such as organ failure or infection severity, will be investigated, paving the way for broader applications of graph-based deep learning in healthcare.

Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Received: 25 March 2025; Accepted: 16 May 2025
Published online: 22 May 2025

References

1. Alanazi, A., Aldakhil, L., Aldhoayan, M. & Aldosari, B. Machine learning for early prediction of Sepsis in intensive care unit (ICU) patients. *Medicina* **59** (7), 1276 (2023).
2. Bao, C., Deng, F. & Zhao, S. Machine-learning models for prediction of sepsis patients mortality. *Med. Intensiva (English Edition)*. **47** (6), 315–325 (2023).
3. Li, X. et al. Machine learning algorithm to predict mortality in critically ill patients with sepsis-associated acute kidney injury. *Sci. Rep.* **13** (1), 5223 (2023).
4. Koozi, H., Lidestam, A., Lengquist, M., Johnsson, P. & Frigyesi, A. A simple mortality prediction model for sepsis patients in intensive care. *J. Intensive Care Soc.* **24** (4), 372–378 (2023).
5. Zhang, Y., Xu, W., Yang, P. & Zhang, A. Machine learning for the prediction of sepsis-related death: a systematic review and meta-analysis. *BMC Med. Inf. Decis. Mak.* **23** (1), 283 (2023).
6. Wang, C., Zhang, J., Liu, L., Qin, W. & Luo, N. Early predictive value of presepsin for secondary sepsis and mortality in intensive care unit patients with severe acute pancreatitis. *Shock* **59** (4), 560–568 (2023).
7. Yang, Z., Cui, X. & Song, Z. Predicting sepsis onset in ICU using machine learning models: a systematic review and meta-analysis. *BMC Infect. Dis.* **23** (1), 635 (2023).
8. Yang, Y. et al. Integrating fuzzy clustering and graph convolution network to accurately identify clusters from attributed graph. *IEEE Trans. Netw. Sci. Eng.* **12**(2), 1112–1125. https://doi.org/10.1109/TNSE.2024.3524077 (March-April 2025).
9. Baswaraju, S., Thirumalraj, A. & Manjunatha, B. Unlocking the potential of deep learning in knee bone Cancer diagnosis using MSCSA-Net segmentation and MLGC-LTNet classification. In: *Sustainable Development Using Private AI* (pp. 190–213) (CRC, 2024).
10. Schertz, A. R. et al. Sepsis prediction model for determining sepsis vs SIRS, qSOFA, and SOFA. *JAMA Netw. Open.* **6** (8), e2329729–e2329729 (2023).
11. Zhang, Y. et al. Development of a machine learning-based prediction model for sepsis-associated delirium in the intensive care unit. *Sci. Rep.* **13** (1), 12697 (2023).
12. Centner, F. S. et al. S-adenosylhomocysteine is a useful metabolic factor in the early prediction of septic disease progression and death in critically ill patients: A prospective cohort study. *Int. J. Mol. Sci.* **24** (16), 12600 (2023).
13. Moor, M. et al. Predicting sepsis using deep learning across international sites: a retrospective development and validation study. *EClinicalMedicine* **62** (2023).
14. Xu, C., Zheng, L., Jiang, Y. & Jin, L. A prediction model for predicting the risk of acute respiratory distress syndrome in sepsis patients: a retrospective cohort study. *BMC Pulm. Med.* **23** (1), 78 (2023).
15. Peng, M., Deng, F. & Qi, D. Development of a nomogram model for the early prediction of sepsis-associated acute kidney injury in critically ill patients. *Sci. Rep.* **13** (1), 15200 (2023).
16. Mangalesh, S., Dudani, S. & Malik, A. The systemic immune-inflammation index in predicting sepsis mortality. *Postgrad. Med.* **135** (4), 345–351 (2023).
17. Duan, Y. et al. Early prediction of sepsis using double fusion of deep features and handcrafted features. *Appl. Intell.* **53** (14), 17903–17919 (2023).

18. Gao, J. et al. Prediction of Sepsis mortality in ICU patients using machine learning methods. *BMC Med. Inf. Decis. Mak.* **24** (1), 228 (2024).
19. Boussina, A. et al. Impact of a deep learning sepsis prediction model on quality of care and survival. *Npj Digit. Med.* **7** (1), 14 (2024).
20. Kamran, F. et al. Evaluation of sepsis prediction models before onset of treatment. *NEJM AI.* **1** (3), AIoa2300032 (2024).
21. Sun, B. et al. Prediction of sepsis among patients with major trauma using artificial intelligence: a multicenter validated cohort study. *Int. J. Surg.* **111**(1), 467–480 (2024).
22. MIMIC-IV (Medical Information Mart for Intensive Care. *Version 4.0). Laboratory for Computational Physiology* (Massachusetts Institute of Technology, 2020). https://mimic.mit.edu/.
23. Kosolwattana, T. et al. A self-inspected adaptive SMOTE algorithm (SASMOTE) for highly imbalanced data classification in healthcare. *BioData Min.* **16** (1), 15 (2023).
24. Hamadneh, T. et al. Using the novel wolverine optimization algorithm for solving engineering applications. CMES Comput. Model. Eng. Sci. **141**(3), (2024).
25. Kusuma, P. D. & Dinimaharawati, A. Three on three optimizer: a new metaheuristic with three guided searches and three random searches. *Int. J. Adv. Comput. Sci. Appl.* **14** (1), 10–14569 (2023).
26. Stephe, S., Manjunatha, B., Revathi, V. & Thirumalraj, A. Osteosarcoma cancer detection using ghost-faster RCNN model from histopathological images. *Iran. J. Comput. Sci.* **8**, 217–231 (2024).

Author contributions
All authors contributed equally.

Funding
The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Ethics approval
The submitted work is original and has not been published elsewhere in any form or language.

Research involving human participants and/or animals
NA.

Additional information
Correspondence and requests for materials should be addressed to R.K.