IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 6, 2020, accepted July 15, 2020, date of publication July 20, 2020, date of current version July 30, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3010556*

# Intensive Care Unit Mortality Prediction: An Improved Patient-Specific Stacking Ensemble Model

**NORA EL-RASHIDY[1], SHAKER EL-SAPPAGH[2,3], TAMER ABUHMED[4], SAMIR ABDELRAZEK[5], AND HAZEM M. EL-BAKRY[5]**

[1]Machine Learning and Information Retrieval Department, Faculty of Artificial Intelligence, Kafrelsheikh University, Kafr El-Sheikh 33516, Egypt
[2]Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain
[3]Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, Banha 13518, Egypt
[4]College of Computing, Sungkyunkwan University, Seoul 561-758, South Korea
[5]Information Systems Department, Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt

Corresponding author: Tamer Abuhmed (tamer@skku.edu)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) under Grant NRF-2016R1D1A1A03934816.

**ABSTRACT** The intensive care unit (ICU) admits the most seriously ill patients requiring extensive monitoring. Early ICU mortality prediction is crucial for identifying patients who are at great risk of dying and for providing suitable interventions to save their lives. Accordingly, early prediction of patients at high mortality risk will enable their provision of appropriate and timely medical services. Although various severity scores and machine-learning models have recently been developed for early mortality prediction, such prediction remains challenging. This paper proposes a novel stacking ensemble approach to predict the mortality of ICU patients. Our approach is more accurate and medically intuitive compared to the literature work. Data were prepared and feature selection was processed under the supervision of the ICU domain expert. The data were split into six modalities based on the expert's decisions. For the prediction process, a separate classifier was selected for each modality based on the performance of the classifiers. We utilized the most popular and diverse classifiers in the literature, including linear discriminant analysis, decision tree (DT), multilayer perceptron, k-nearest neighbor, and logistic regression (LR). Then, a stacking ensemble classifier was constructed and optimized based on the fusion of these five classifier decisions. The framework was evaluated using 10,664 patients from the medical information mart for intensive care (MIMIC III) benchmark dataset. To predict patient mortality, extensive experiments were conducted using the patients' time series data of different lengths. For each patient, the first 6, 12, and 24 hours of the first stay were tested. The results indicate that our model outperformed the state-of-the-art approaches in terms of accuracy (94.4%), F1 score (93.7%), precision (96.4%), recall (91.1%), and area under the receiver operator characteristic (ROC) curve (93.3%). These results demonstrate the ability and efficiency of our approach to predict ICU mortality.

**INDEX TERMS** Ensemble classifier, intensive care unit, information fusion, machine learning, mortality prediction.

## LISTS OF ABBREVIATIONS

| Term | Abbreviation |
|---|---|
| ICU | Intensive Care Unit |
| ICT | Information Communication Technology |
| CSRU | Cardiac Surgery Recovery Unit |
| SICU | Surgical Intensive Care Unit |
| MICU | Medical Intensive Care Unit |
| MIMIC-III | Using Medical Information Mart for Intensive Care III |
| EHR | Electronic Health Record |
| LOS | Length of Stay |
| APACHE | Acute Physiology and Chronic Health Evaluation |
| SOFA | Simplified Acute Physiology Score |
| MPM | Mortality Probability Model |
| BIDMC | Beth Israel Deaconess Medical Center |

The associate editor coordinating the review of this manuscript and approving it for publication was Pengcheng Liu.

| | |
|---|---|
| MCAR | Missing Completely at Random |
| MAR | Missing at Random |
| MNAR | Missing Not at Random |
| KNN | K-Nearest Neighbor |
| DT | Decision Tree |
| LR | Logistic Regression |
| MLP | Multilayer Perceptron |
| LDA | Linear Discriminate Analysis |
| SVM | Support Vector Machine |
| WSRT | Wilcoxon Sign Rank Test |

## I. INTRODUCTION

Most hospitals are moving towards replacing the traditional infrastructure with smart systems to maximize the utilization of information and communication technology [1]. Smart health is expected to significantly improve the quality of service in the healthcare sector. The intensive care unit (ICU) is a special department in the health care sector that typically helps people recover from life-threatening injuries and illnesses [2]. Patients in the ICU require consistent supervision from medical staff and caregivers to ensure the stability of their health. Therefore, early- and reliable-prediction tools for sensitive medical conditions would be useful caregiving aids. Mortality prediction is one of the most important tasks in critical care research [3]. The purpose of mortality prediction is not only related to identifying high-risk people and to making the right decisions but also to saving ICU beds for patients in need [4]. Several scoring systems, such as the acute physiology and chronic health evaluation system (APACHE) [5] and simplified acute physiology (SAPS), have been developed recently [6]. These systems are not always appropriate for all patients in the ICU because they are not sufficiently accurate [7]. To improve the accuracy of the mortality prediction, researchers have proposed models and scoring systems under specific conditions. For example, Moridani *et al.* [8] proposed a mortality scoring system for people with heart problems. They concluded that the model provides acceptable results when risks are predicted early. However, selecting the right model for a patient, who may be diagnosed with various diseases and then admitted to ICU, is still considered a challenge. Therefore, a more general and efficient mortality prediction model is required, and this is the focus of our study.

The growth of the electronic health record (EHR) in the ICU, where the IUC clinical measurements and other patient medical histories are preserved, presents a great opportunity to develop predictive and analytical tools that help both patients and physicians using machine learning (ML) techniques [9]. Using HER, various studies have utilized the ML advances to build accurate mortality prediction models. Graham *et al.* [10] provided a patient readmission prediction model based on logistic regression (LR). Their work relied on the patient's age, arrival mode, and triage category. Gentimis *et al.* [11] utilized lab test results to predict the patient's length of stay (LOS), and sepsis has been used for mortality prediction by [12], [13]. Liu *et al.* [14] provided a mortality prediction model using the support vector machine (SVM). References [15], [16], and [17] endorsed the use of LR to develop a clinical prediction model for mortality while [18] and [4] supported the use of random forest (RF) and decision tree (DT), respectively. Although these studies proposed various ML models for mortality prediction, the resulting models have suffered unstable performance, especially in different ICU categories [15]. This is because mortality prediction in the ICU encounters many challenges relating to data diversity, high dimensionality, irregular sampling, and data imbalance [14], [19]. The high dimensionality increases the computational complexity and decreases the model accuracy. The irregular sampling creates a challenge for extracting data and reflects the large amount of missing data [20]. Highly imbalanced data also negatively impact our ability to obtain satisfactory results as it may lead to classifier bias in favor of the majority class [21]. All these factors impact single-classifier models differently. As discussed by Anand *et al.* [16], each classifier has its advantages and limitations. Accordingly, using a single classifier with high diversity and high-volume datasets may increase both the search-space area and misclassified data, which may in turn degrade the classifier's accuracy [22], [23].

Therefore, instead of training on a large database, Breiman [24] proposes dividing data into small pieces, building classifiers on each small piece, and then combining all the predictors. This is known as ensemble learning. Ensemble learning methods are statistical and computational learning methods that are reminiscent of human learning behavior that involves considering multiple options before making a crucial decision. An ensemble classifier consists of a set of ML techniques that combine learning algorithms, decisions, or other characteristics to provide more accurate and reliable decisions [25]. It is well known that ensemble classifiers provide more efficient classification models by taking advantage of used classifiers and avoiding their limitations [26]. Generally, the ensemble model outperforms all constituent classifiers, but the participating weak learners should be accurate and diverse. Accuracy is achieved by optimizing the performance of each base learner, and diversity is achieved based on the ensemble type. For a homogeneous ensemble, this is achieved by using different samples (e.g. bagging), different weights (e.g. boosting), different feature sets, and samples (e.g. RF). For a heterogeneous ensemble, this is achieved by using different ML algorithms or different hyperparameters for the same algorithm. There are many combination strategies to construct the ensemble classifier including voting and stacking. Given the fact that ensemble classifiers outperform single classifiers, various theoretical and empirical studies recommend using ensemble classifiers as an effective model because they can improve the prediction accuracy for the following multiple reasons [27], [28]. (1) The ensemble classifier provides a handy solution by training different basic classifiers with different data partitions and then combines their outputs. In the case of insufficient training data, resampling techniques can be used to create overlapping random subsets from the available data. The data generated from resampling is used to train different classifiers, and then to create an ensemble output; such a method is well known to improve the ensemble classifier [29]. (2) It is extremely difficult for a single classifier to solve extremely complex problems that involve a decision boundary that lies outside the space of the function implemented by the selected classifier model [30]. Therefore, a suitable combination of ensemble classifiers can learn this non-linear boundary. (3) The ensemble classifier reduces the risk of a poor selection of a single that cannot generalize the performance, and therefore, combining several classifiers by averaging the output may decrease the risk of the poor-performance of the selected single classifier. Moreover, this contributes to decrease the risk of making a poor selection [31]. (4) Since medical data is a fusion of heterogeneous data obtained from different sources, a single classifier cannot be used to learn all the information included in this data. For example, ICU data are of several types (e.g., time series, multivariant, Electroencephalogram (EEG) recording, image scanning, blood tests, etc.). In such cases, data from each type can naturally be trained with different classifiers then combined for the final decision. The process of combining data from multiple sources to provide a more informative decision is called data fusion [32]. Ensemble-based models have been successfully used for applications adopting the data fusion process [33]– [35].

Maintaining the diversity between the base learner classifiers is the cornerstone of building an efficient ensemble classifier. There are several intuitive approaches to building an ensemble model with different decision boundaries, and to achieve the highest diversity, these can be summarized as follows [36]. (1) Dividing training data into several subsets with each subset used for learning with a single base classifier—this division can be performed using resampling techniques such as bootstrapping, which creates a training data subset with a replacement, or it can be performed manually. (2) Using different training parameters and different classifiers, as certain types of classifiers are not suitable for all types of data, selecting several algorithms and combinations can also serve as a better way to build classifiers with different decision boundaries.

This research investigates the use of a proposed ML-based ensemble classification technique to develop an accurate and medically intuitive model for early mortality prediction. Our research was conducted under the supervision of an ICU medical expert. Accordingly, multiple preprocessing steps were made to improve the quality of the data extracted from the MIMIC-III dataset. [37]. We investigate the role of time-series data (i.e. vital signs, laboratory tests, and demographic) on the performance of the proposed ML model. Our classifier combines the prediction powers of many well-known and diverse algorithms such as multi-layer perceptron (MLP), linear discriminate analysis (LDA), K-nearest neighbor (KNN), DT, and LR.

To optimize our ensemble model, the data are vertically divided into feature subsets based on the suggestions of the domain expert. The results show a superior performance of the proposed approach compared to that of the state-of-the-art models. To summarize, the study focuses on answering the following questions. (1) What are the most important factors in the early prediction of ICU mortality? (2) Do ML techniques, especially ensemble models, outperform the current mortality scoring systems?

The contributions of this study are summarized as follows.

- We propose an accurate and medically intuitive ICU mortality prediction framework based on a comprehensive list of critical features from ICU patients. The model is based on a novel stacked generalization ensemble algorithm. The constructed model is optimized to solve a binary classification based on an accurate pipeline of preprocessing and classifier optimization steps.
- The framework analyzes the role of multivariate time-series data on the performance of the prediction model. This includes the study of the effect of using different lengths of the temporal data (i.e. 6, 12, and 24 hours) of patients, starting from the admission time of the first stay.
- Our model incorporates six distinctive modalities based on the opinions of the medical expert. Each base classifier (i.e. KNN, MLP, LDA, DT, or LR) was optimized with the list of modalities, and the statistically significant classifier was selected for every modality. The resulting list of classifiers was used to build the proposed stacking model. Different meta-classifiers were tested, and the best performing classifier was selected.
- We evaluate the proposed approach by performing extensive experiments using a balanced dataset of 10,664 patients from the MIMIC-III benchmark dataset. The results of our framework are superior to those of the standard scoring systems, single classifiers, and standard ensemble techniques. The proposed model achieved encouraging accuracy and strong generalization performance that can adapt to the classification of various types of data in ICU.
- A comprehensive analysis was conducted to compare out stacking model with popular ML models such as KNN, MLP, LDA, DT, and LR. Moreover, we compare our model with similar ensemble models, including bagging, RF, boosting, and voting classifiers.

The rest of this paper is organized as follows. Section 2 presents a review of related work, whereas materials and methods are discussed in Sections 3. The research setting and experimental design for the proposed framework are detailed in Section 4, with the results presented in Section 5. We discuss our evaluation of the proposed solution in Section 6, and finally, we conclude our research with a brief outline of the main findings in Section 7.

## II. RELATED WORK

Intensive care is a complex department that always handles patients with critical cases, many of whom suffer from several diseases simultaneously [38]. Therefore, patients admitted to the ICU require close and continuous supervision to avert the potential for rapid degradation in their health status. Intensive monitoring through the ICU equipment results in large medical records that require efficient and accurate systems for assistance in data analysis. Using ICU data to predict future events, such as patient mortality, is considered one of the most critical topics in ICU research [39]. In this section, we discuss the related literature studies on this topic. We specifically focus on (1) traditional scoring systems and (2) ML-based systems for the mortality prediction.

### A. TRADITIONAL SCORING SYSTEMS

For traditional scoring systems, various models, such as APACHE (the most widely used and well-known model in critical care), have been developed. The first version of APACHE was developed in 1981; it is based on 34 physiological features extracted from the first 24-h period after ICU admission to determine the patient's health status. In 1999, the APACHE II scoring model was adopted, including only 12 physiological measures, in addition to the patient's age. This version was extended to APACHE III in 1993 by adding new features such as the gender, and LOS in ICU. In 2006, APACHE IV was developed to provide a risk-estimating score used to predict short-term mortality. SPAS II is another scoring system developed in 1993. This is frequently used for specifying the severity of diseases in patients in the ICU. However, it is only valid for adult patients (i.e. $age > 15$), and the values range between 0 and 163. One other conventional scoring system is the sepsis-related organ failure assessment (SOFA) score [40], which is used to assess six organs by evaluating the breathing, nervous system, liver, and coagulation related features. This score indicates the derangement for each organ, based on a number between 0 and 4, where 0 represents normal cases, and 4 represents critical cases [41]. Further details on the current scoring system can be found in the reports of Jeong [6] and Arabi *et al.* [42]. Overall, the traditional score-based systems have attempted to predict mortality using similar approaches based on specific numbers of vital signs and measurements. However, owing to their weakness in discrimination, there is no efficient up-to-date scoring system that can be used for mortality prediction. This has sparked the need for other assistant techniques that can help in realizing the early prediction of mortality.

### B. ML-BASED SYSTEMS

The studies are based on the regular ML techniques for mortality prediction [14], [43], [44]. Member *et al.* [45] illustrated the use of SVM results to improve the prediction of patient LOSLOS using association rules. Kim *et al.* compared the use of DT, artificial neural networks (ANN), and SVM with APACHE III to predict mortality [46], and reported that DT gave the best performance. In [18], Ghose *et al.* used RF, SVM, and LR, compared to SAPS, and concluded that RF achieved an area under the curve (AUC) of 87%; they also found that RF outperforms some of the state-of-the-art predictive models based on SVM and LR. Therefore, selecting the most suitable algorithm is considered a critical point in building an efficient model. Various studies used LR for the early prediction of mortality in the ICU [15], [47], [48]. For example, Ball *et al.* [17] used LR to develop a clinical prediction approach for mortality based solely on heart-rate features. Their work provides acceptable results for Canadian elderly patients who suffered from heart problems. Anand *et al.* [16] recommended LR to predict the mortality of adult patients admitted for cardiac surgery and to the coronary care unit. Sadeghi *et al.* recommended SVM as the most accurate algorithm for mortality prediction [4]. Dybowski *et al.* adopted an ANN-based system to predict mortality [49] and compared its performance with those of LR and SVM. They reported that ANN was more accurate than SVM and LR. Others in [62] used ANN for predicting various events such as mortality, LOS, and ICD 9 diagnosis. The study achieved AUC values of 0.93, 0.88, 0.87 for these tasks, respectively.

The limitation of the current ML-based studies is that most studies build prediction models for specific types of patients and use a single ML algorithm for their proposed system. The authors of [41], [43], [50] [51] and [52] developed mortality-prediction models for specific patients such as kidney-failure patients. Although accurate results were obtained in these studies, the application of their models is limited to specific domains. In the ICU, classifying a patient and selecting the right model, especially within the first 24 hours after admission, are challenging [53]. Ding *et al.* [54] attempted to overcome this shortcoming by developing a two-step framework—one for clustering and the other for mortality prediction. Their method uses the just in time learning algorithm (JITL) to collect relevant samples, after which the prediction process is conducted locally. The performance delivered by this method may be better than that delivered by conventional prediction systems. Other researchers have attempted to solve the problem of data unavailability in the first 24 hours by utilizing the correlation between specific patient's measurements and mortality. For example, Krishnan and Kamath [55] only used lab test features and genetic algorithm-based wrapper to do feature selection followed by an optimization of an ANN model for mortality prediction [56]. The authors reported that the developed model improved prediction accuracy by approximately 2–3% with an increase of 5% in terms of AUC. A similar idea has been considered by Miao *et al.* [57], where laboratory data have been used to predict the risk of mortality for heart failure patients. However, one drawback of using lab measurements is that these tests may not be available for all the ICU patients, and not all critical cases can be predicted using lab-test results. The same concept had been used in [4] and [42], where only heart-rate signals were used and then aggregated during the first hour for mortality prediction. In [58], the authors built a deep learning model that used only on nursing notes to predict mortality.

Other studies have developed ensemble models to improve mortality-prediction accuracy. For example, Johnson *et al.* [59] provided an ensemble for survival prediction using a Bayesian ensemble schema that consists of 500 weak learners (DT); it achieved an area under the receiver operator characteristic curve ROC (AUC) of 86%. Using an ensemble of the same learner may not be practical and may not deliver the best accuracy. In [60], Awed *et al.* designed an ensemble model for mortality prediction, which includes RF, DT, and Naive Bayes (NB). They applied their ensemble model on 20 features extracted in the first 6 hours of the patient admission. It achieves an AUROC score of 82%. It may not provide the best performance in the first 6 hours as many values may not be available during this initial period. In [61], J. Xia *et al.* proposed an ensemble model based on the long short-term memory (LSTM) technique for mortality prediction. The idea behind this work is to use two LSTM layers based on 50 features extracted in the first 24 hours. They achieved an AUC score of 85.5%. Caicedo-Torres *et al.* proposed a deep learning model called ConvNet for mortality prediction [58]. Their system achieved 87.3% in terms of AUC, and they added further steps by using ConvNet in handling both static and dynamic data.

These inconsistencies in the results and performance reported in the literature clarify that no single algorithm outperforms others in terms of prediction, and none of the developed systems are commonly used for prediction owing to the low power of discrimination. Therefore, it is a challenge to provide an accurate prediction of hospital mortality. This work focuses on handling this challenge and dealing with a complex ICU dataset and improving mortality prediction in the ICU.

## III. MATERIALS AND METHODS

This section details the selected dataset, preprocessing steps, and extracted features in our experiments.

### A. MIMIC III DATABASE

The MIMIC III dataset is a benchmark ICU dataset developed by the MIT Lab for computational physiology. It comprises the EHR data related to patients admitted to the ICU at the Beth Israel Deaconess Medical Center (BIDMC) in Boston. MIMIC-III is an updated version of MIMIC-II released in 2010. MIMIC-III can be used once accessibility confirmation is obtained from the Physionet Organization. Privacy issues have been managed in all MIMIC versions by removing all the sensitive patient data such as names, phone numbers, and addresses.

### B. USED DATASET DESCRIPTION

MIMIC-III is a vast single-center database including information related to patients admitted to the ICU in a large tertiary hospital. It comprises data of 53,423 distinct ICU admissions in the period between 2001 and 2012. In MIMIC-III, 38,597 distinct patients are aged over 16. The mean age among adults is 65.8, and 55.9% are males. The means of 4,579 measurements and 380 laboratory tests are available in the MIMIC III tables. Table 1 presents the statistics of the dataset according to age and gender. There are different modalities in MIMIC III, including physiological measurements, medications, laboratory tests, descriptive details, nursing notes, and reports. The data are distributed as a set of commas separated files (CSV) that can be mapped to a relational database such as PostgreSQL. The resulting database consists of 26 related tables linked by unique identifiers such as SUBJECT_ID.

**TABLE 1. Dataset description.**

| Age group | Gender | Number of patients |
|---|---|---|
| Neonate | F | 3629 |
| Adult | F | 15476 |
| >89 | F | 1294 |
| Neonate | M | 4245 |
| Adult | M | 21179 |
| >89 | M | 697 |

In this work, we consider adult patients ($age > 15$ years) admitted to the cardiac surgery recovery unit (CSRU), medical ICU (MICU), or surgical ICU (SICU). Table 2 presents the distribution of patients according to the ICU type.

**TABLE 2. Utilized ICU types.**

| First care unit | Admission type | Survived | Deceased |
|---|---|---|---|
| SICU | Emergency | 2117 | 202 |
| SICU | Urgent | 65 | 15 |
| MICU | Emergency | 6387 | 1657 |
| MICU | Urgent | 104 | 34 |
| CSRU | Emergency | 2115 | 624 |
| CSRU | Urgent | 152 | 13 |

### C. DATA PREPROCESSING STEPS

The following challenges are addressed to prepare the MIMIC data for the ML process:

Imbalanced data distribution: Imbalanced data produce biased results, and the algorithms are optimized for the majority class. Our selected MIMIC dataset is imbalanced because it has 35000 patients who have survived (i.e. class 0) and 5400 dead patients who died in the ICU (class 1). The main class represents 15.43% of the total dataset.

Data redundancy: MIMIC has several redundant features with different names and units of measurement. Besides, many of these features are highly correlated.

Irregularity of time series: MIMIC data are primarily time-series data. However, not all features are collected at the same rate. Most features have many missing values. A certain feature may be lost or incomplete because sensors or instruments can break down or be improperly operated by the medical staff.

### D. COHORT SELECTION

We used the following three main inclusion criteria for the study cohort: (1) care unit type: only patients from CSRU, MICU, and SICU are considered; (2) age: only adult patients (age $>$ 15 years) are included; (3) the number of admissions: only the first admission is considered to prevent possible data leakage during analysis. Figure 1 details the number of patients in each step and the filtering steps performed.

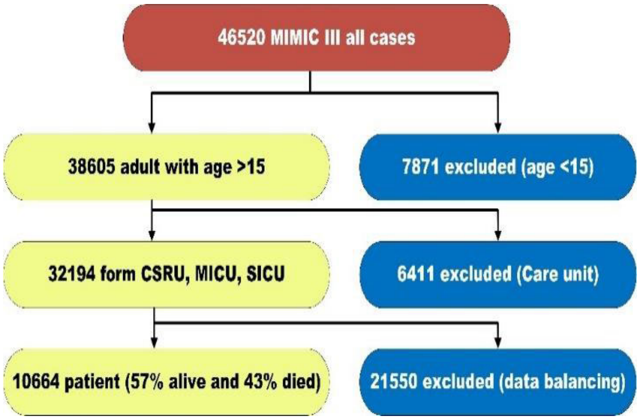

46520 MIMIC III all cases
38605 adult with age >15
7871 excluded (age <15)
32194 form CSRU, MICU, SICU
6411 excluded (Care unit)
10664 patient (57% alive and 43% died)
21550 excluded (data balancing)

**FIGURE 1.** **Inclusion and exclusion criteria of patients.**

### E. TABLE SELECTION

MIMIC III relational database consists of 26 tables that store patient demographics, vital signs, laboratory test results, diagnosis notes, etc. Our patient monitoring data are mainly extracted from the chartEvents, outputEvents, labEvents, d_itmes, and lab_items tables. To aggregate data from the MetaVision and CareVue medical information systems, we utilize the inputevents_cv and inputevents_mv tables. Table 3 shows the table selected from the dataset and the selected column of each table.

## IV. PROPOSED FRAMEWORK

This work presents a stacking ensemble classifier for ICU mortality prediction as a binary classification task (0: survived and 1: died). We performed experiments on the model using the MIMIC-III dataset. As shown in Figure 2, the proposed framework has a set of four modules. The first module collects the MIMIC data and performs the preprocessing steps to improve data quality. The outputs of this step include three datasets called feature sets A, B, and C. These data are collected for different time windows (i.e. at 6, 12, and 24 hours). The second module optimizes the five selected ML models for the mortality-prediction binary-classification task. The optimization process utilizes the datasets prepared by the first component. The third module fuses the optimized base classifiers of the previous module and explores the roles of the standard ensemble techniques. The tested ensemble models include RF, AdaBoost, bagging, stacking, and voting.

We propose a customized stacking ensemble model, which is developed based on the six optimized base classifiers. Our proposed ensemble classifier works as a committee of experts where each expert is specialized in one field. A committee can sometimes make a wiser decision than individual experts can. Opinions of all committees (classifiers) are aggregated for consideration using various mechanisms like weighted voting. Finally, the performance of different models is evaluated using unseen test datasets.

### A. DATA PREPROCESSING

This step aims to improve the quality of the collected medical data. After exploring the dataset, we found that the collected data contains missing and outlier values. Most of the ML models cannot work with missing data, and several are sensitive to outliers. Missing data occur due to many reasons including equipment failure and system/network errors. Furthermore, because vital signs and lab-test results may be recorded at different frequencies, some values may also be missing. The data preprocessing includes sample selections, data balancing, normalization, removing outliers, and handling missing data.

#### 1) DATA BALANCE

Class imbalance is a common problem in medical data [63]. In MIMIC III, a minority of patients died during ICU admission. There are many techniques for handling imbalanced data including data sampling. Two main techniques are used for data sampling: downsampling and oversampling [64]. Oversampling involves increasing the number of cases of the minority class. Various techniques can be used for oversampling such as random oversampling [65], Synthetic Minority Oversampling Technique (SMOTE) [66], adaptive synthetic sampling approach, or (ADASYN) [67]. Downsampling involves decreasing the number of records of the majority class, and various techniques are used for data downsampling, such as random undersampling [67], cluster and Tomek links [68]. As we mentioned in Section 3, we used three main inclusion criteria for choosing the dataset: age, care unit type, and data balancing. After excluding records according to age and ICU type, 32194 patients were included, 26320 survivors and 5874 fatalities. Therefore, in our study we used the undersampling technique to keep the data balanced, it works by removing records from the majority (the survival) class. The undersampling technique does not add any noise in the data but loses some knowledge from the majority class. However, our dataset is huge, so this loss will not affect the resulting dataset. The undersampling process cut the sample to 10,644 patients (57% survivors, 43% fatalities).

#### 2) UNIFICATION OF MEASUREMENT UNITS

The MIMIC III dataset is collected from both MetaVision and CareVue medical information systems. Therefore, we can find a single feature measured by different units. Each feature is unified and converted to a single unit according to some conversion rules. For example, the calcium chloride measurement is recorded in mg and ml, with a percentage

**TABLE 3. Lists the selected tables and columns.**

| Table Selected | Content | Column Selected | Content |
|---|---|---|---|
| Patient | Defines each patient with unique subject-ID. It contains patient demographics. | Subject-ID | Unique patient identifier. |
| | | Expire Flag | A binary flag indicates whether the patient died or survived. |
| ICUStay | Linked to the patient table by subject-ID. It includes care unit related data, such as care unit name, time of admission, time of discharge, and the number of days in each unit. | ICUStay-ID | Unique identifier for every single ICU-Stay in hospital. |
| | | First-care-Unit | The first care unit where the patient was admitted. |
| | | LOS | LOS as the number of days the patient stays. |
| ChartEvents | Contains all measurements recorded for patients in ICU, such as heart rate, respiratory rate, etc. | Chart-time | The time the measurement was recorded. |
| | | Item-ID | Unique identifier for measurement. |
| | | value | The result of the measurement. |
| InputEvents (MV/CV) | Inputs refer to any liquids that have been administered either by tube feeding or any other tool. Each row is associated with item_id, which is defined in the D_itmes table. | Chart-time | The time of recording. |
| | | Item-ID | Unique identifier for measurement. |
| | | Amount | Amount of entering fluid. |
| OutputEvents | Outputs include urine, stool, and sweat. Each row associated with item_id is defined in the D_itmes table. | Chart-time | Time the measurement is recorded. |
| | | Item_id | Unique identifier for measurement. |
| | | value | Amount of the output. |
| LabEvents | Stores all laboratory tests. | Chart-time | Time the lab test takes place. |
| | | Item_id | Unique identifier for measurement. |
| | | value | Result of the lab test. |
| D_itmes, Lab-items | Dictionaries for all details of items and lab tests. | Label | Name of measurement. |
| | | Unit name | Unit of measurement. |

of 89% for mg and 11% for ml. In this case, all the values are converted to mg. Table S5 in the Supplementary File provides further information about all the conversion rules for the measurements.

### 3) REMOVING OUTLIERS

Many ML algorithms are sensitive to outliers [69]. Extreme outlier values are removed from our dataset. To detect outliers, the acceptable range of each feature is determined by our medical specialist, and all values outside the acceptable range are removed.

### 4) HANDLING MISSING VALUE

Many statistical analysis methods are used to impute missing values, including hot-deck and regression imputation [70]. These techniques deliver excellent performance when the percentage of missing data is in the range of 5–10%. Advanced techniques, such as expectation maximization [64], [71], and multiple imputations, are considered appropriate when the percentage of missing data lies in the range of 20–50%, resulting in more reliable data.

In this paper, features with more than 60% of missing data were entirely removed. However, time-series features, such as arterial blood pressure and temperature, which were missing in 40–60% of the data were retained owing to their medical importance. We selected patients that have at least three records for all time-series features. Missing values of time-series features are filled using (1) forward filling, (2) backward filling, and (3) the means of the same patient's data. Other features with missing data were filled using the expectation-maximization algorithm [72].

### 5) DATA NORMALIZATION

Data normalization unifies the roles of all features. We rescaled all the features using the min-max normalization technique (Equation 1), where $X'$ is the scaled value, and $X$ is the original value.

$$X' = \frac{X - min(X)}{max\,(X) - min(X)} \tag{1}$$

## B. FEATURE SELECTION AND EXTRACTION

For each patient, a set of features and measurements were recorded during the patient's admission in the ICU. For our study, features were extracted from four main tables: chartevents, inputevents_mv/cv, outputevents, and labevents. Of around 1200 features scanned with the help of a medical expert from four main tables, the most important 80 features were used. As verified by medical expert knowledge, we collected the medically relevant features for mortality prediction. Table S3 in the Supplementary File presents these features with their item IDs. The resulting time-series data are filtered for the first 24 hours for each patient, and we only consider the first admission. To support an exhaustive comparison study, we select three main feature sets, A, B, and C, which are described in the following subsections. Each feature was extracted using statistical and functional forms such as summation and maximum. Table 4 details the functional form used for each feature in every feature set A, B, and C. Features were extracted according to their importance. For example, for GSC it is only important to consider the minimum value. For the level of consciousness, the value when the patient arrived at the ICU is the most important, and the last conscious status before the prediction can be made. In addition to these data, demographic features, such

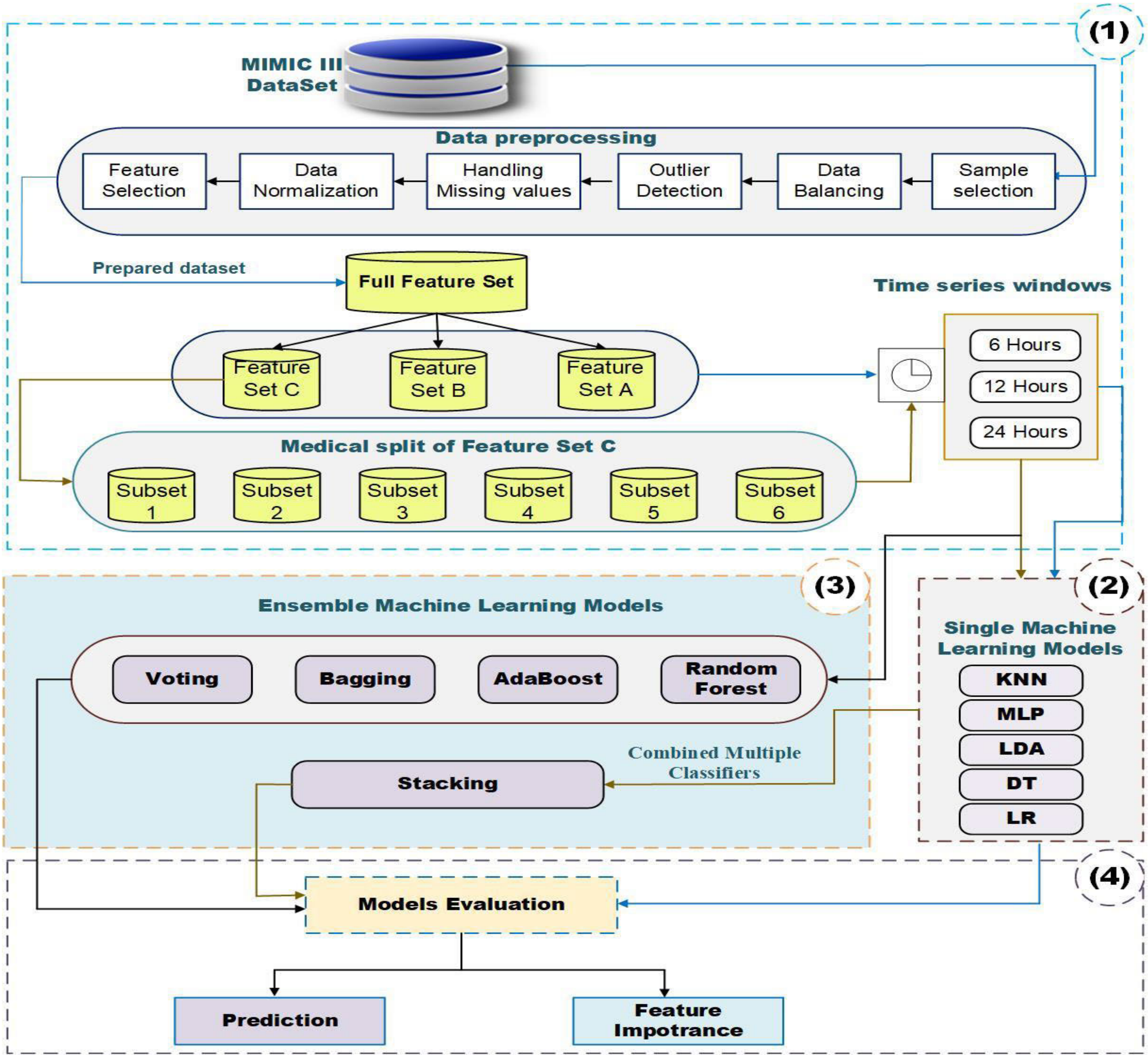


**FIGURE 2.** Architecture of the proposed framework.

as age, weight were extracted from each patient's table. The following points discuss the three feature sets and the number of features in each set.

Feature Set A contains 28 statistical features extracted from 12 features, including the time-series vital signs, such as heart rate and respiratory rate, as they provide the first indication of the patient's health status. A patient's heart rate exceeding the normal limit indicates tachyarrhythmia, which may be a fetal-like ventricular tachycardia or fibrillation, or it may be an atrial flutter or fibrillation. In contrast, a heart rate falling below the normal range may lead to a heart blockage or cardiac arrest [73]. All situations are critical and may lead to sudden death. The same can be said for oxygen saturation—a decrease in oxygen saturation is critical and may lead to lactic acid accumulation or cardiac arrest (in severe cases) [74].

Feature Set B contains 62 features extracted from 27 features, including time-series vital signs from the feature set A, in addition to some lab-test results, such as urine output and hemoglobin, and other endocrinal gland test results. For urine volume, we use the sum function to calculate the volume of urine, drainage from condom catheters, Foley catheters, etc. We use the sum function to calculate the total amount of urine per day. The amount of urine falling below the normal amount indicates oliguria or an acute kidney injury. For hemoglobin, we use the minimum and maximum functions as these are the most important for hemoglobin. High hemoglobin increases blood thickness, which may lead to heart attacks, clots, and strokes. Low hemoglobin may also be a risk indicator, especially for patients with kidney diseases.

For feature Set C, 322 features were extracted from 3 tables including input_events, outputevents, and labevents. This set

**TABLE 4. Extracting features from time-series data of all the feature set.**

| Feature extracted | Feature Name |
|---|---|
| | Feature Set A |
| Minimum, maximum, variance | Heart rate, respiratory rate, oxygen saturation, non-invasive/Invasive blood pressure (mean, diastolic, systolic), glucose |
| First, Last | Level of consciousness |
| Minimum | GSC score |
| | Feature Set B |
| Minimum, maximum, variance | Heart rate, respiratory rate, oxygen saturation, Non-Invasive/Invasive blood pressure (mean, diastolic, systolic), glucose |
| First, Last | Level of consciousness, motor response, eye-opening, respiratory pattern, urine color |
| Minimum, Maximum | sodium, potassium, endocrinal glands (magnesium, arterial CO2), kidney Functions (BUN, creatine kinase), hematocrit, temperature, urea nitrogen urine, hemoglobin |
| Minimum | GSC score |
| Sum | Urine out |
| | Feature Set C |
| Minimum, maximum, variance | Heart rate, respiratory rate, oxygen saturation, non-invasive /invasive blood pressure (mean, diastolic, systolic), glucose, Cardiac index, Mean airway pressure, Expiratory ratio, Inspiratory ratio |
| First, Last | Level of consciousness, motor response, eye-opening, respiratory pattern, urine color, arterial line site appears, Arterial PaO2, pupil response right, pupil response left, alveolar-arterial Gradient, INV#2 waveform Appear, Breath sounds |
| Minimum, Maximum | Sodium, potassium, endocrinal glands (HCo3, magnesium, arterial CO2), kidney functions (BUN, Creatine Kinase) Hematocrit, Temperature, Anion gap, PH, Urea nitrogen urine, alkaline phosphates, Albumin, Toprin, creatine, CVP, Urea Nitrogen, Urine, lactate, blood Flow, anion gap, prothrombin time, mean airway pressure, carboxin dioxide, Neutrophils, Monocytes, Eosinophils, Basophils, Alveolar-arterial, Urobilinogen, Cholesterol, HDL, Green top hold (plasma), Neutrophils, Monocytes, Eosinophils, Basophils |
| Minimum | GSC score, Hemoglobin, WBC, RBC, o2 flow, Tidal Volume (Spont), |
| Maximum | bicarbonate, pain score, PTT, sodium, potassium, cortisol, cholesterol, HDL |
| Sum | Urine out |

contains all the features used in feature sets A and B, as well as some other tests such as Arterial Blood Gases (ABG) tests. ABG includes different markers: PH, PO2, PCO2, HCO3, Anion Gap, and Base Excess (see table S3 in the Supplementary File). This indicates the levels of oxygen and carbon dioxide in the blood. ABG features are used to evaluate respiratory and kidney functions and provide a view of the metabolic state. They are also used as an indication of how efficiently the lungs provide oxygen and remove carbon dioxide. ABGs also measure the blood pH and the body's acid-base balance.

When selecting the time-series data, we focus on data collected from the first day. The time-series data from different measurements are often recorded and sampled at irregular periods. Therefore, we perform discretization on the time and measurement axis, resampling data at regular space periods (every hour). For each subset extracted earlier, three-time frames (6, 12, and 24 hours) are used to extract different feature sets with different sampling times. Figure 3 explains how we extracted feature sets according to time frames. The first three inputs depict the various ways a feature set can be sampled (6, 12, and 24 hours), leaving 1 hour as a time gap between training and prediction. The event of interest considers the time of prediction (time until the patient either died or was discharged from the ICU).

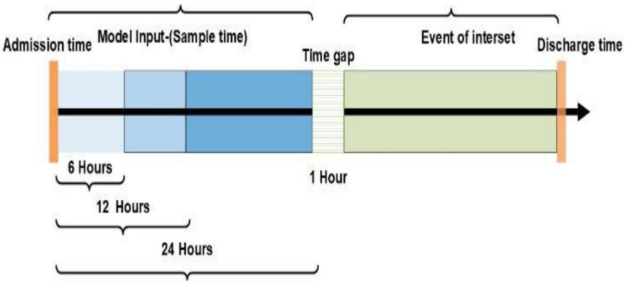


**FIGURE 3. Extraction of features according to time frames.**

### 1) SINGLE CLASSIFIER MODELS

First, we divided the whole feature set into the six modalities based on the domain expert decisions. Second, we evaluated and optimized five ML classifiers (i.e. LR, LDA, DT MLP, and KNN) based on the grid search technique and assigned the best performing classifier to each modality using its statistical significance.

We chose classifiers based on their diversity and popularity in the ICU domain [75], [76], see Table 5. The five selected base classifiers achieved the best performance for the six modalities, where the DT classifier had the best performance for two modalities. This justifies the selected types and number of classifiers.

**TABLE 5. Classifiers adopted in this study.**

| Classifier Type | Name | Label |
|---|---|---|
| Linear model | Logistic regression | LR |
| Statistical model | Linear discriminate analysis | LDA |
| DT model | Decision tree | DT |
| Neural Network model | Multi-layer perception | MLP |
| Distance-based model | k-nearest neighbors | KNN |

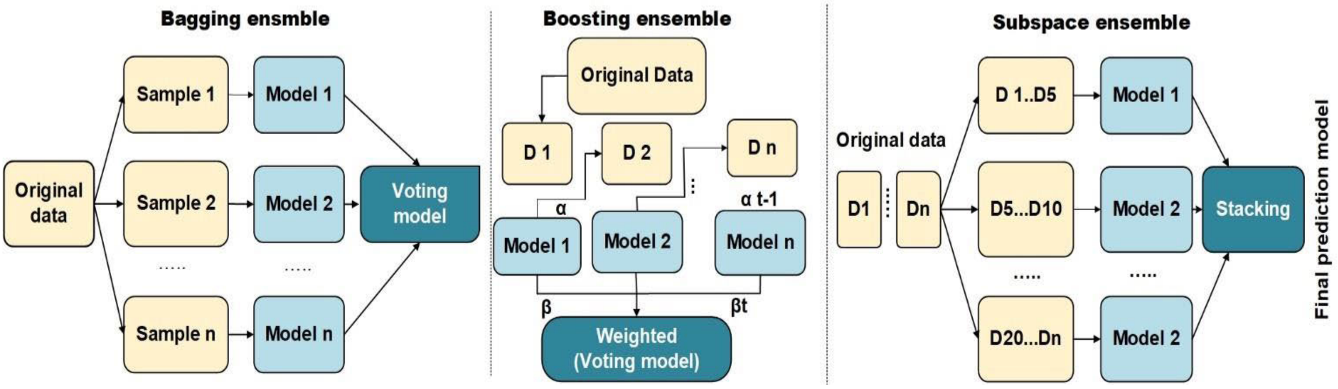


**FIGURE 4.** Standard ensemble Classifiers (bagging, boosting, and stacking).

### 2) ENSEMBLE CLASSIFIER

Using an ensemble classifier is a well-known ML technique that combines the prediction of a set of single classifiers (base or weak learner) using weighted or unweighted techniques and obtains a model that outperforms every base classifier. This type of learning is intuitive as it simulates the human nature of considering multiple perspectives before making a final decision. In conventional learning, a single classifier is used to solve the problem, but in ensemble learning, several classifiers work together to solve the problem. There are two main categories of ensemble classifiers: (1) homogeneous ensemble, which uses the same classifier such as RF, where RF is an ensemble of DT, and (2) heterogeneous ensemble, which uses a set of diverse classifiers such as SVM and DT [27]. The sequence of work of all the base classifiers and the method for individual decision combinations are important when building an ensemble classifier. The three major ensemble models—bagging, boosting, and stacking—are presented in Figure 4 [28].

## V. RESULTS

We performed extensive experiments to determine the efficiency of our proposed system using the MIMIC III dataset since it has the complete patient EHR, including the patient profile, daily vital signs, laboratory test results, summaries of admission and discharge, nursing, and caregiver notes. We only input specific patient information to the tested classification models, starting with 1200 features and ending with the 80 most essential features. We only include the 10644 patients (5000 survivors and 4644 fatalities) from the undersampling process to ensure data balancing. Our objective is to predict whether any individual patient will die in the ICU.

### A. EXPERIMENTAL SETUP

All the experiments were performed on a laptop workstation with an Intel Core i7, 16 GB RAM and a 1 Terabyte hard disk on a Windows 10 64-bit system. All algorithms of single classifiers and ensembles were implemented in the Python language using the scikit-Learn library. For each experiment, we make a pipeline optimization by sequentially applying the following steps. Initially, the datasets were divided into training sets (75%) and testing sets (25%). For the training set, we used stratified 10-fold cross-validation (CV) technique to train all the models. Training data used to tune hyperparameter for all classifiers using a grid search algorithm. Using 10-fold CV helps us to avoid overfitting and generates an evaluation matrix report that is based on generalization performance. The unseen testing sets were used to measure the generalization performance of the trained models. Figure 5 shows the structure of the cross-validation on training and testing data. We ensure that no admission data exist in both training and test sets as this may enable the used algorithms to memorize the records and perform better in the testing phase. Each patient was represented with a feature vector that encodes summary information about the patient's health status over the chosen period. For example, the heart rate measurements are encoded into multiple variables that describe the maximum, minimum, and mean values. These summarization values allow us to consider the differences in the feature time series during the selected period.

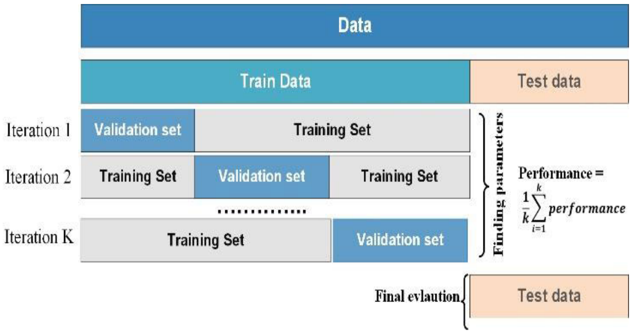


**FIGURE 5.** The K-Fold cross-validation.

### B. EVALUATION METRICS

We used the standard list of evaluation metrics for classification problems, including precision, recall, F1-score,

**TABLE 6. Evaluation metrics.**

| Metric | Abbreviation | Equation | No. | Definition |
|---|---|---|---|---|
| Precision | P | $\frac{tp}{tp+fp}$ | (2) | Percentage of positive records that are correctly classified from the total number of positive records |
| Recall | R | $\frac{tp}{tp+tn}$ | (3) | Percentage of positive records that are correctly classified from the total number of records |
| Accuracy | ACC | $\frac{tp+tn}{tp+fp+tn+fn}$ | (4) | The ratio between the correct predictions and the total number of instances |
| Specificity | SP | $\frac{tn}{tn+fp}$ | (5) | Percentage of the actual negative records that are correctly classified |
| F1 score | F1 | $2*\frac{P*R}{P+R}$ | (6) | The mean of both precision and recall. Used for checking the balanced performance of the model. |
| Area under the ROC curve | AUC | $\frac{s_p - n_p + (n_{n+1)/2}}{n_p n_n}$ | (7) | Measures how much the model can discriminate between models, where sp is the number of positive records, and np, nn are the number of positive and negative records, respectively. |

specificity, and AUC. Both CV and testing results were obtained. The CV results were used to select the best models, and the generalization performance was measured using the testing datasets. Table 6 lists the used metrics.

### *C. RESULTS OF BASE CLASSIFIERS*

#### 1) RESULTS WITHOUT FEATURE SELECTION

In this section, we evaluate the five single classifiers KNN, MLP, LDA, LR, and DT using the three prepared feature sets A, B, and C with three sliding windows of 6, 12, and 24 hours, as illustrated in Figure 3. In this section, 45 different experiments were performed for the three feature sets, with each feature set classified with the five ML classifiers and three different time frames. We carefully selected the most common ML algorithms in the medical domain. Table 7 presents the CV accuracy of each model and also details the testing performance of each model represented by F1, P, R, ACC, SP, and AUC.

For feature set A, we observed that MLP generates the best testing results using the 6- and 12-hour datasets. It achieves values of F1 = 0.775 and AUC = 0.766 using the 6-hour dataset and F1 = 0.781 and AUC = 0.772 using the 12-hour dataset. For the 24-hour dataset, LDA offers the best performance with F1 = 0.776 and AUC = 0.760. For feature set B, the overall testing results were improved by 3–4% in terms of F1, AUC, and ACC. As detailed in Table 7, for all three-time windows—6, 12, and 24 hours—MLP outperforms all other algorithms by achieving testing results of F1 = 0.829 and AUC = 0.804 using the 6-hour dataset. These testing results improved to F1 = 0.855 and AUC = 0.824 using the 24-hour dataset. For feature set C, the overall testing performance was improved by 6–15%, compared to those of feature sets A and B. The LDA and MLP achieved the best testing results using the 6- and 12-hour datasets, respectively. MLP test results are F1 = 0.890 and AUC = 0.874. Using the 12-hour dataset, both MLP and LDA improved the test results by approximately 2–3%, compared to those when using the 6-hour dataset. By using the 24-hour dataset, the LR achieved the best testing performance with F1 = 0.905 and AUC = 0.922.

From these previous experiments, we observed the following: (i) as we extended the extraction window, the performance of each model improved; (ii) feature set C achieved the best results, and all the models performed better when using it. Therefore, feature set C is critical for further analysis to determine how we can improve the ML models using these data. Figures 6(a), (b), and (c) compare the performance of all three feature sets on the time frames of 6, 12, and 24 hours, respectively.

#### 2) RESULTS AFTER FEATURE SELECTION

Based on the results presented in Table 7, our experiment enhanced the performance of the best-performing feature set. As discussed in the previous section, feature set C achieved the best results. In this experiment, we explored the role of the feature selection step to enhance the performance of the ML models based on feature set C. There are three main feature-selection techniques: filter methods, wrapper methods, and embedding methods [44]. In this study, we used the wrapper approach that works depending on the learning classifier results: the best features were chosen based on the classifier results [77]. Wrapper methods require more computations than filter methods do, but they provide higher accuracy. We used the recursive feature elimination (RFE) wrapper approach to select the best features from feature set C. As presented in Table 8, after using the feature-selection step, the overall performance improved the accuracy by 1–2% for some algorithms such as LDA and LR. For the 6-hour dataset, we observe that LDA yields the best testing results of F1 = 0.873 and AUC = 0.894. MLP delivers the best testing results using the 12-hour dataset (F1 = 0.874 and AUC = 0.892). When using the 24-hour dataset, MLP outperforms all the other models (F1 = 0.881 and AUC = 0.901). All these evaluations are based 10-fold cross-validation.

Figure 6(d) compares all the algorithms. Feature selection removes the noisy and less-informative features from the dataset, which improves the results and stability of the ML models; it also improves the computational complexities of the resulting models. However, using single ML models can be improved by using an ensemble of these diverse models. In the next section, we extend our research to explore the

**TABLE 7.** Base classifier model results.

| Time Frame | Model | CV Accuracy | F1 | P | R | ACC | SP | AUC |
|---|---|---|---|---|---|---|---|---|
| Subset (A) | | | | | | | | |
| First 6 hours | KNN | 0.759±0.026 | 0.744 | 0.744 | 0.734 | 0.744 | 0.776 | 0.745 |
| | DT | 0.743±0.017 | 0.730 | 0.700 | 0.740 | 0.730 | 0.810 | 0.737 |
| | LR | 0.748±0.098 | 0.745 | 0.741 | 0.741 | 0.735 | 0.795 | 0.726 |
| | MLP | 0.789±0.012 | 0.775 | 0.775 | 0.765 | 0.775 | 0.772 | 0.766 |
| | LDA | 0.715±0.023 | 0.673 | 0.673 | 0.663 | 0.673 | 0.645 | 0.718 |
| First 12 Hours | KNN | 0.786±0.090 | 0.754 | 0.755 | 0.754 | 0.754 | 0.731 | 0.750 |
| | DT | 0.749±0.023 | 0.734 | 0.734 | 0.734 | 0.734 | 0.750 | 0.739 |
| | LR | 0.785±0.127 | 0.753 | 0.723 | 0.753 | 0.753 | 0.777 | 0.753 |
| | MLP | 0.821±0.077 | 0.781 | 0.781 | 0.781 | 0.781 | 0.788 | 0.772 |
| | LDA | 0.758±0.012 | 0.754 | 0.764 | 0.754 | 0.754 | 0.761 | 0.740 |
| First 24 hours | KNN | 0.785±0.026 | 0.728 | 0.718 | 0.728 | 0.768 | 0.820 | 0.738 |
| | LDA | 0.802±0.060 | 0.776 | 0.774 | 0.770 | 0.776 | 0.732 | 0.760 |
| | DT | 0.780±0.098 | 0.740 | 0.740 | 0.742 | 0.740 | 0.776 | 0.722 |
| | LR | 0.76±0.0321 | 0.728 | 0.728 | 0.748 | 0.745 | 0.717 | 0.756 |
| | MLP | 0.808±0.020 | 0.748 | 0.746 | 0.749 | 0.787 | 0.713 | 0.757 |
| Subset (B) | | | | | | | | |
| First 6 hours | KNN | 0.763±0.012 | 0.750 | 0.741 | 0.759 | 0.750 | 0.705 | 0.752 |
| | DT | 0.763±0.098 | 0.746 | 0.726 | 0.767 | 0.779 | 0.737 | 0.727 |
| | LR | 0.821±0.023 | 0.799 | 0.821 | 0.777 | 0.826 | 0.765 | 0.796 |
| | MLP | 0.831±0.033 | 0.829 | 0.862 | 0.799 | 0.860 | 0.806 | 0.804 |
| | LDA | 0.828±0.019 | 0.812 | 0.824 | 0.800 | 0.843 | 0.795 | 0.804 |
| First 12 hours | KNN | 0.761±0.048 | 0.753 | 0.750 | 0.757 | 0.792 | 0.737 | 0.787 |
| | DT | 0.779±0.923 | 0.734 | 0.759 | 0.770 | 0.784 | 0.797 | 0.734 |
| | LR | 0.842±0.039 | 0.825 | 0.834 | 0.816 | 0.855 | 0.793 | 0.790 |
| | MLP | 0.872±0.032 | 0.855 | 0.877 | 0.833 | 0.881 | 0.831 | 0.824 |
| | LDA | 0.842±0.082 | 0.831 | 0.830 | 0.831 | 0.858 | 0.801 | 0.804 |
| First 24 hours | KNN | 0.802±0.023 | 0.771 | 0.787 | 0.756 | 0.816 | 0.783 | 0.800 |
| | DT | 0.762±0.046 | 0.731 | 0.781 | 0.686 | 0.792 | 0.735 | 0.736 |
| | LR | 0.861±0.021 | 0.84z | 0.852 | 0.831 | 0.871 | 0.830 | 0.855 |
| | MLP | 0.882±0.012 | 0.859 | 0.867 | 0.852 | 0.885 | 0.820 | 0.880 |
| | LDA | 0.873±0.033 | 0.850 | 0.856 | 0.844 | 0.878 | 0.817 | 0.831 |
| Subset (C) | | | | | | | | |
| First 6 hours | KNN | 0.861±0.391 | 0.837 | 0.822 | 0.852 | 0.847 | 0.795 | 0.824 |
| | DT | 0.792±0.021 | 0.787 | 0.758 | 0.766 | 0.728 | 0.745 | 0.788 |
| | LR | 0.870±0.011 | 0.889 | 0.893 | 0.889 | 0.889 | 0.919 | 0.863 |
| | MLP | 0.901±0.018 | 0.890 | 0.896 | 0.890 | 0.890 | 0.917 | 0.874 |
| | LDA | 0.891±0.032 | 0.901 | 0.915 | 0899 | 0.901 | 0.919 | 0.823 |
| First 12 hours | KNN | 0.912±0.029 | 0.877 | 0.877 | 0.868 | 0.877 | 0.867 | 0.877 |
| | DT | 0.802±0.019 | 0.825 | 0.833 | 0.806 | 0.855 | 0.958 | 0.810 |
| | LR | 0.892±0.022 | 0.874 | 0.876 | 0.878 | 0.878 | 0.913 | 0.863 |
| | MLP | 0.916±0.033 | 0.885 | 0.884 | 0.887 | 0.889 | 0.925 | 0.874 |
| | LDA | 0.901±0.073 | 0.891 | 0.910 | 08611 | 0.890 | 0.891 | 0.865 |
| First 24 hours | KNN | 0.917±0.021 | 0.904 | 0.916 | 0.891 | 0.889 | 0.906 | 0.892 |
| | DT | 0.811±0.012 | 0.876 | 0.866 | 0.868 | 0.828 | 0.837 | 0.832 |
| | LR | 0.880±0.038 | 0.890 | 0.905 | 0.861 | 0.880 | 0.902 | 0.898 |
| | MLP | 0.929±0.072 | 0.912 | 0.922 | 0.893 | 0.902 | 0.917 | 0.901 |
| | LDA | 0.901±0.098 | 0.90.2 | 0.894 | 0.872 | 0.882 | 0.903 | 0.891 |

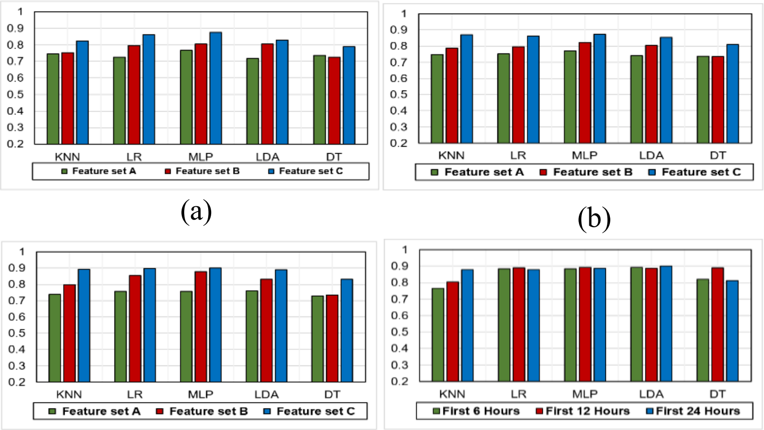
possible role of ensemble models to improve the mortality prediction performance.

### D. RESULTS OF ENSEMBLE MODELS

Ensemble classifiers are expected to enhance the performance of our classification. Several experiments were performed to compare and evaluate ensemble methods. We explore the capabilities of the ensemble models with and without a feature-selection step. We examine the most popular ensemble techniques, including RF, voting, bagging, and boosting.

#### 1) RESULTS BEFORE FEATURE SELECTION

Table 9 presents the results of using an ensemble classifier based on feature set C as this achieved the best results with single models. Several ensemble classifiers were tested, including homogenous classifiers (bagging, RF, and boosting) and heterogeneous classifiers (voting). Not all ensemble classifiers improve mortality-prediction performance. As indicated in Table 9, RF offers the worst testing performance among the ensemble classifiers. It achieves $F1 = 0.765$ and $AUC = 0.812$ using the 6-hour dataset, $F1 = 0.773$ and $AUC = 0.832$ using the 12-hour dataset, and $F1 = 0.776$ and $AUC = 0.842$ using the 24-hour dataset. The overall performance with RF was no better than that of a single classifier because RF chooses a random subset of features to be considered at each branch, and these random subsets may not be able to influence the predictive powers of the models. Other ensemble classifiers achieved some improvements, compared to the single classifiers. Bagging



(a) (b) (c) (d)

**FIGURE 6.** Base classifier AUC testing results for mortality prediction using (a) 6 h, (b) 12 h, (c) 24 h, and (d) feature selection.

**TABLE 8.** Single classifier model results with feature selection.

| Time Frame | Model | CV accuracy | F1 | P | R | ACC | SP | AUC |
|---|---|---|---|---|---|---|---|---|
|  | KNN | 0.848±0.032 | 0.727 | 0.697 | 0.760 | 0.768 | 0.774 | 0.767 |
| First 6 Hours | LR | 0.890±0.002 | 0.862 | 0.861 | 0.863 | 0.887 | 0.904 | 0.884 |
|  | DT | 0.845±0.021 | 0.782 | 0.911 | 0.732 | 0.854 | 0.912 | 0.821 |
|  | MLP | 0.926±0.042 | 0.862 | 0.855 | 0.870 | 0.887 | 0.899 | 0.884 |
|  | LDA | 0.914±0.022 | 0.873 | 0.857 | 0.889 | 0.895 | 0.899 | 0.894 |
|  | KNN | 0.866±0.018 | 0.773 | 0.779 | 0.767 | 0.812 | 0.843 | 0.805 |
| First 12 Hours | LR | 0.903±0.005 | 0.8736 | 0.880 | 0.860 | 0.895 | 0.915 | 0.891 |
|  | DT | 0.889±0.011 | 0.882 | 0.839 | 0.929 | 0.896 | 0.872 | 0.900 |
|  | MLP | 0.927±0.062 | 0.874 | 0.880 | 0.869 | 0.896 | .0915 | 0.892 |
|  | LDA | 0.911±0.032 | 0.870 | 0.873 | 0.868 | 0.892 | 0.909 | 0.888 |
|  | KNN | 0.893±0.033 | 0.855 | 0.871 | 0.840 | 0.880 | 0.886 | 0.879 |
| First 24 Hours | LR | 0.893±0.027 | 0.845 | 0.872 | 0.840 | 0.880 | 0.886 | 0.879 |
|  | DT | 0.846±0.003 | 0.77 | 0.69 | 0.92 | 0.840 | 0.910 | 0.814 |
|  | MLP | 0.921±0.000 | 0.865 | 0.878 | 0.852 | 0.888 | 0.895 | 0.887 |
|  | LDA | 0.912±0.003 | 0.881 | 0.901 | 0.862 | 0.901 | 0.901 | 0.901 |

**TABLE 9.** Results of ensemble classifiers based on feature set C.

| Model | Time Frame | CV accuracy | F1 | P | R | ACC | SP | AUC |
|---|---|---|---|---|---|---|---|---|
| RF | 6 Hours | 0.827±0.112 | 0.765 | 0.902 | 0.679 | 0.813 | 0.945 | 0.812 |
|  | 12 Hours | 0.855±0.003 | 0.773 | 0.901 | 0.690 | 0.843 | 0.941 | 0.832 |
|  | 24 Hours | 0.867± 0.015 | 0.776 | 0.899 | 0.752 | 0.851 | 0.939 | 0.842 |
| Bagging | 6 Hours | 0.901±0.022 | 0.891 | 0.911 | 0.873 | 0.900 | 0.917 | 0.900 |
|  | 12 Hours | 0.920±0.007 | 0.899 | 0.921 | 0.860 | 0.919 | 0.930 | 0.902 |
|  | 24 Hours | 0.922±0.000 | 0.901 | 0.940 | 0.903 | 0.911 | 0.924 | 0.916 |
| AdaBoost | 6 Hours | 0.907±0.022 | 0.886 | 0.923 | 0.862 | 0.899 | 0.922 | 0.883 |
|  | 12 Hours | 0.927±0.332 | 0.889 | 0.926 | 0.867 | 0.921 | 0.928 | 0.879 |
|  | 24 Hours | 0.935±1.105 | 0.911 | 0.933 | 0.900 | 0.929 | 0.933 | 0.904 |
| Voting | 6 Hours | 0.890±0.087 | 0.863 | 0.892 | 0.823 | 0.888 | 0.912 | 0.852 |
|  | 12 Hours | 0.891±0.113 | 0.872 | 0.902 | 0.813 | 0.879 | 0.922 | 0.863 |
|  | 24 Hours | 0.899±0.009 | 0.903 | 0.959 | 0.919 | 0.891 | 0.941 | 0.906 |

and boosting ensemble classifiers achieved better results than RF in all the time frames. The best performance for both the classifiers was achieved using the 24-hour dataset (F1 = 0.901 and AUC = 0.911 for bagging and F1 = 0.916 and AUC = 0.904 for boosting). The F1 and AUC scores for bagging are better than those for all the other models.

Regarding the heterogeneous voting classifiers [33], we used LR, KNN, DT, LDA, and MLP as the base classifiers. We used the soft aggregation method, which predicts the class label of the record based on the argmax of the sums of the predicted probabilities. Voting classifiers achieved better results than the other ensemble classifiers did. For the 6-hour dataset, they provided F1 = 0.863 and AUC = 0.825. The results were improved when using the 12-hour dataset (F1 = 0.872 and AUC = 0.863) and further improved when using the 24-hour dataset (F1 = 0.903 and AUC = 0.891). Figure 7 (a) compares the ROC curves for these ensemble classifiers. All these evaluations are based on 10-fold cross-validation.

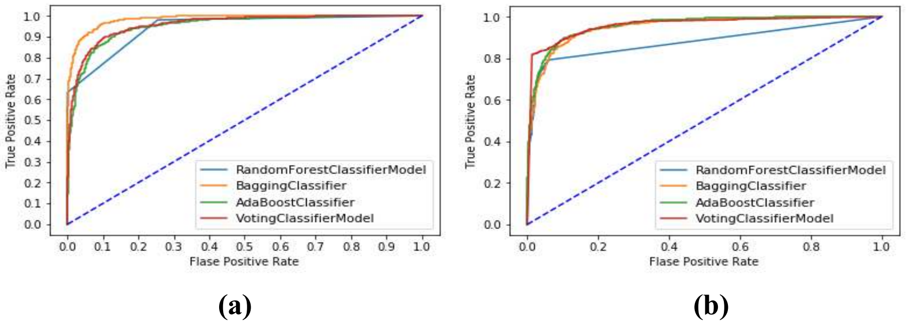


**FIGURE 7.** AUC score for all ensemble classifiers after 24 hours: (a) using all features and (b) with feature selection.

**TABLE 10.** Results of the ensemble classifier with feature selection.

| Model | Time Frame | CV Accuracy | F1 | P | R | ACC | Sp | AUC |
|---|---|---|---|---|---|---|---|---|
| RF | 6 Hours | 0.861±0.005 | 0.798 | 0.891 | 0.724 | 0.847 | 0.936 | 0.830 |
| | 12 Hours | 0.861±0.083 | 0.799 | 0.891 | 0.724 | 0.847 | 0.936 | 0.830 |
| | 24 Hours | 0.831± 0.024 | 0.799 | 0.891 | 0.724 | 0.847 | 0.936 | 0.832 |
| Bagging | 6 Hours | 0.893±0.019 | 0.863 | 0.878 | 0.848 | 0.887 | 0.915 | 0.881 |
| | 12 Hours | 0.911±0.006 | 0.880 | 0.885 | 0.874 | 0.899 | 0.918 | 0.896 |
| | 24 Hours | 0.911±0.009 | 0.862 | 0870 | 0.855 | 0.886 | 0.908 | 0.882 |
| Boosting | 6 Hours | 0.911±0.018 | 0.880 | 0.885 | 0.874 | 0.899 | 0.918 | 0.896 |
| | 12 Hours | 0.893±0.003 | 0.863 | 0.878 | 0.848 | 0.887 | 0.915 | 0.881 |
| | 24 Hours | 0.901±0.843 | 0.867 | 0.872 | 0.862 | 0.889 | 0.909 | 0.885 |
| Voting | 6 Hours | 0.916±0.043 | 0.889 | 0.912 | 0.866 | 0.909 | 0.94 | 0.903 |
| | 12 Hours | 0.919±0.032 | 0.836 | 0.949 | 0.747 | 0.877 | 0.971 | 0.859 |
| | 24 Hours | 0.920±0.004 | 0.923 | 0.951 | 0.898 | 0.938 | 0.966 | 0.909 |

### 2) RESULTS AFTER FEATURE SELECTION

In this section, we reevaluate the same ensemble models but after performing the wrapper feature-selection step. Table 10 indicates that using RFE feature selection improves the performance of some ensemble classifiers. RF achieves values of F1 = 0.799 and AUC = 0.830 using the 12-hour dataset. The bagging classifier achieves the best performance (F1 = 0.885). The best performance of the boosting classifier was obtained when using the 6-hour dataset (F1 = 0.880 and AUC = 0.896). The performance (F1) of the voting classifier improved by 2%, and we observe that the best accuracy was obtained when using the 24-hour dataset (F1 = 0.923 and AUC = 0.909). All these evaluations are based on 10-fold cross-validation. Figure 7(b) compares the ensemble algorithms after using the feature selection step. From the previous results, we made the following observations: (i) using feature selection decreases the computational time but does not guarantee an improvement in the classifier performance, and (ii) no model performs the best in all time frames. To summarize the effect of using wrapper feature selection on single and ensemble classifiers, Figure 8 compares the best performance of all the classifiers.

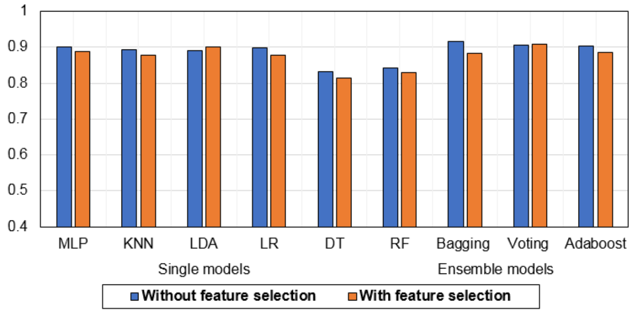


**FIGURE 8.** Comparison of all classifiers (with and without) the feature selection phase.

## *E. RESULTS OF THE STACKING MODEL*

Stacking ensemble learning uses the concept of meta-level based learning [36]. The set of base-learners is generated by applying various ML algorithms. The ensemble model achieved the best results when all the base classifiers were accurate and diverse. Accuracy can be achieved by optimizing the hyperparameter of each base classifier, and diversity can be achieved using many methodologies, including different training examples, feature sets, and parameters. Several studies [23], [78] analyzed the relationship between classifier diversity and the quality of stacking and concluded that diversity may be considered as the selection criteria for building the ensemble classifier. In the previous section, we tested the most popular ensemble techniques such as bagging, boosting, and voting, and they did not enhance performance. In this experiment, we adopt our novel methodology to build a customized and medically oriented ensemble model. The overall development process is formulated in Algorithm 1. This model is also described in the following specific steps.

**TABLE 11. Feature subset used in an ensemble classifier.**

| Subset # | Features |
|---|---|
| Subset (1) | Arterial blood pressure diastolic, arterial blood pressure mean, arterial blood pressure systolic, glucose finger stick, heart rate, HR alarm [high], HR alarm [low], respiratory rate, high resp-rate, NBP alarm [low], NBP alarm [high], SpO2 alarm [high], SpO2 Alarm [low], temperature C, heart rhythm, mean airway pressure, skin color, skin condition, skin integrity, skin temperature. |
| Subset (2) | Arterial CO2 (Calc), arterial O2 pressure, arterial paCO2, HCO3 (serum), PAO2, calculated total CO2, arterial pH. |
| Subset (3) | Anion gap, BUN, CVP, hematocrit, hemoglobin, cortisol, creatine kinase, bicarbonate, chloride, platelet count, prothrombin time, RCW, O2 Flow (lpm), Protein, glucose, ketone, lipase, oxygen, cardiac index, carbon dioxide, chloride (serum). |
| Subset (4) | GCS Total, dorsal pedpulse [right], dorsal ped pulse [left], eye-opening, pupil response left, pupil response right, verbal response, level of consciousness, motor response, arterial line site appears, INV2 wave form appear. |
| Subset (5) | calcium, magnesium, phosphorous, sodium, phosphate, sodium urine, sodium whole blood, potassium, creatinine, RBC, alkaline phosphatase, urea nitrogen. |
| Subset (6) | Tidal volume, dextrose, fentanyl, fresh frozen plasma, gastric meds, insulin, LR, norepinephrine, propofol, PO intake, solution, amiodar, epinephrine-k, KCL, neosynephrine-k, midazolam, Foley, urine out void, hemoglobin, albumin , alveolar-arterial gradient, urobilinogen, urine color, creatinine urine, urine [appearance], urine color. |

*First*, we divide feature set C, which achieved the best results, into six subsets according to medical expert knowledge. Table 11 details these six feature sets. These feature subsets are independent. *Second*, each subset of these heterogeneous sets is used to optimize a list of different base classifiers. We investigate the combination of heterogenous learning algorithms to create a more accurate ensemble model. In other words, we used diverse types of learning algorithms, including MLP (non-linear classifier), DT (tree-based classifier), LR and LDA (linear classifier), and KNN (instance-based classifier). The resulting five models are diverse, and each produces a different error. As a result, the staking model is expected to outperform all models. The two main evaluation techniques, namely, classification performance test and statistical significance tests (Wilcoxon sign rank test, Friedman test, and Nemenyi test), are used to estimate the base classifier efficiency. The following subsections discuss the selection process. *Third*, the decisions of the level-0 classifiers are combined using the meta classifier (level-1 classifier). In our case, we selected the LR as the meta classifier. Figure 9 shows the cross-validation process for all base classifiers.

### 1) CLASSIFICATION PERFORMANCE TESTIN

The purpose of this step is to calculate the differences among the testing accuracies of the KNN, LDA, MLP, LR, and DT models using every subset. This comparison is used to specify the most suitable algorithm for each subset. For example, subset 1 is trained using all the classifiers, and the differences of ACC measures between all algorithms are recorded. The first part of Table 12 shows the differences between the test accuracy for subset 1. The performance of MLP was superior to that of other techniques for this subset. Table 12 details the differences between the testing accuracies of all the algorithms for all subsets.

### 2) STATISTICAL SIGNIFICANCE TESTIN

The main challenge encountered when selecting the most suitable model for every subset is how much we can trust the estimated skill for the selected model. Because our dataset is balanced, we depend on both accurate results and statistical tests to measure the statistical significance of the differences among the tested models. In this study, we depend on the Wilcoxon signed-rank test [79]. It is a nonparametric test recommended by Demsar for comparing algorithm performance. [79]. It works upon the number of losses, ties, and wins obtained over the algorithm. An algorithm is considered statistically better if the number of wins is plus half the number of ties.

All the algorithms were compared for each feature subset using the Friedman test [80]. The Friedman test is a non-parametric test of the repeated measures ANOVA. The Friedman test determines where there is a significant difference among classifiers, but it does not show which algorithm is the best.

To rank the classifiers and select the best one, the Nemenyi post hoc test is used to calculate the average rank for each classifier on each feature subset. Table 13 shows the results of the Freidman test for all subsets and the average rank obtained from the Nemenyi test for all classifiers. When multiple classifiers are compared against each other, the results of the Nemenyi test can be visually represented using critical difference diagrams. Figure 10 shows the critical difference among all classifiers for each subset according to the average rank from the Nemenyi test.

The optimization hyperparameters for each algorithm were tuned using the grid search technique. Table 14 specifies the most suitable algorithm for each subset and presents its list of hyperparameters.

### 3) RESULTS OF THE STACKING CLASSIFIER

Our novel proposed algorithm is based on the generalization stacking ensemble model (also called the stacking ensemble model). It combines the decisions of different classification algorithms. All the sub-models contribute equally to the final combined prediction. In comparing the results of our customized ensemble classifier with the performance of all the base classifiers, we found that the proposed model

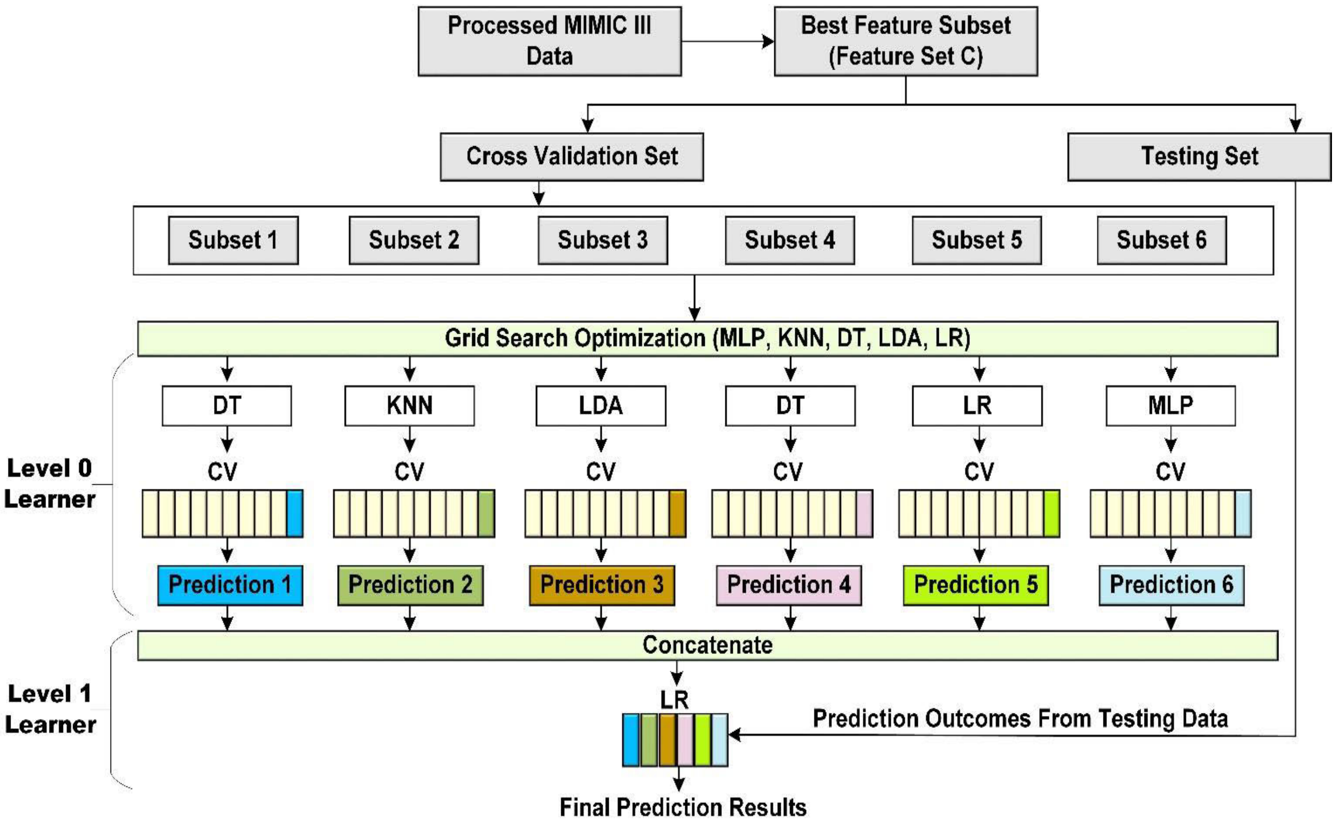


**FIGURE 9.** Stacking ensemble based on a cross-validation of all feature subsets.

**TABLE 12.** Average differences between the accuracies of base classifiers for all subsets.

| | LR | LDA | MLP | KNN | DT |
|---|---|---|---|---|---|
| | | Using subset 1 | | | |
| LR | 0.000 | 0.022 | -0.090 | 0.010 | -0.014 |
| LDA | | 0.000 | -0.028 | -0.017 | -0.031 |
| MLP | | | 0.000 | 0.010 | -0.037 |
| KNN | | | | 0.000 | -0.033 |
| DT | | | | | 0.000 |
| | | Using subset 2 | | | |
| LR | 0.00 | 0.008 | -0.008 | -0.015 | -0.010 |
| LDA | | 0.000 | 0.080 | -0.020 | -0.027 |
| MLP | | | 0.000 | -0.012 | -0.010 |
| KNN | | | | 0.000 | -0.071 |
| DT | | | | | 0.000 |
| | | Using subset 3 | | | |
| LR | 0.000 | -0.027 | 0.012 | 0.020 | -0.060 |
| LDA | | 0.000 | 0.038 | 0.030 | 0.020 |
| MLP | | | 0.000 | -0.080 | -0.014 |
| KNN | | | | 0.000 | -0.080 |
| DT | | | | | 0.000 |

| | LR | LDA | MLP | KNN | DT |
|---|---|---|---|---|---|
| | | Using subset 4 | | | |
| LR | 0.000 | 0.015 | 0.012 | -0.017 | 0.018 |
| LDA | | 0.000 | 0.107 | -0.220 | 0.010 |
| MLP | | | 0.000 | -0.138 | -0.011 |
| KNN | | | | 0.000 | 0.018 |
| DT | | | | | 0.000 |
| | | Using subset 5 | | | |
| LR | 0.000 | 0.013 | 0.040 | 0.018 | 0.010 |
| LDA | | 0.000 | -0.011 | -0.012 | -0.010 |
| MLP | | | | - 0.027 | -0.028 |
| | | | 0.000 | | |
| KNN | | | | 0.000 | -0.002 |
| DT | | | | | |
| | | | | | 0.000 |
| | | Using subset 6 | | | |
| LR | 0.000 | -0.020 | -0.053 | 0.010 | -0.020 |
| LDA | | 0.000 | -0.030 | 0.011 | -0.003 |
| MLP | | | 0.000 | 0.016 | 0.032 |
| KNN | | | | 0.000 | 0.013 |
| DT | | | | | 0.000 |

achieved the best results. All the stacking classifiers used the LR as the metaclassifier. To ensure the superiority of LR as the best meta classifier in our proposed stacking model, 15 different experiments were conducted with the same base classifiers but using four other meta classifiers: KNN, MLP, DT, and LDA. Table S6 in the Supplementary File presents the CV accuracy of each of the resulting stacking models. In addition, Table S6 in the Supplementary File details the testing performance of each model represented by F1, P, R, ACC, SP, and AUC.

In addition, we compared the proposed model (Figure S1 in the Supplementary File) with the traditional stacking model

**Algorithm 1** Construction of the Enhanced Stacking Ensemble Classifier

```
Algorithm-1: Stacking with cross validation and independent based classifiers
Input: Training data $D = \{X_i, y_i\}_{i=1}^{m}, X_i \in \mathbb{R}^n, y_i \in \{0, 1\}$
        $\{DS = D1, D2, D3, D4, D5, D6\}$
        $B$ base classifiers $\{DT, LR, MLP, LDA, KNN\}$, each classifier is optimized for a specific subset
Output: An ensemble classifier H
1. By using cross validation approach, randomly split $DS_i \in DS$ into $K$ equal-size subsets: $DS_i = \{DS_{i1}, DS_{i2}, \ldots, DS_{iK}\}$
   for k = 1 to K do
       //1.1 Learn level-0 classifiers
       for t = 1 to T do
           Learn a classifier $h_{kt}$ from $DS_i \backslash DS_{ik}$
       end for
       //1.2 Create level-1 a training set
       for $X_i \in DS_{ik}$ do
           $DB = []$
           $Dp+ =$ Create a new instance $\{x_i', y_i\}, x_i' = \{h_{k1}(X_i), h_{k2}(X_i), \ldots, h_{kT}(X_i)\}$
       end for
   end for
2. Repeat step 1 for each subset in $DS$
3. Concatenate the generated $DB$ from all classifiers $B$
4. // Learn a second-level classifier
   Learn a new classifier $h'$ from the collected $DB$ //LR in our case
5. Re-train all first level classifiers
   for t = 1 to 5 do
   Train classifier $h_t$ based on $Dt$
   end for
return H(x)=$h'(h_1(x), h_2(x), \ldots, h_5(x))$
```

**TABLE 13.** Freidman and Nemenyi tests result for all feature subsets.

| Subset Num | Friedman chi-square | | Average Rank | | | | |
|---|---|---|---|---|---|---|---|
| | Statistics | P-value | KNN | DT | LR | MLP | LDA |
| Subset 1 | 6.201 | 0.028 | 1.92 | 1.28 | 2.24 | 2.02 | 1.80 |
| Subset 2 | 2.776 | 0.008 | 1.12 | 1.62 | 1.71 | 2.8 | 2.20 |
| Subset 3 | 3.331 | 0.006 | 3.2 | 4.08 | 4.16 | 2.34 | 2.61 |
| Subset 4 | 5.598 | 0.008 | 4.06 | 2.08 | 3.09 | 2.87 | 2.66 |
| Subset 5 | 3.762 | 0.001 | 2.76 | 2.17 | 1.32 | 1.66 | 1.90 |
| Subset 6 | 3.130 | 0.012 | 2.59 | 1.65 | 1.64 | 1.13 | 2.01 |

**TABLE 14.** Optimized hyperparameters for selected algorithms.

| Subset No. | Algorithm | Hyperparameter |
|---|---|---|
| Subset 1 | DT | criterion=gini, max_depth=3, random_state=33 |
| Subset 2 | KNN | n_neighbors= 5, weights =uniform, algorithm=auto |
| Subset 3 | LDA | n_components=3, solver= svd, tol=0.0001 |
| Subset 4 | DT | criterion=gini, max_depth=2, random_state=33 |
| Subset 5 | LR | penalty=l2, solver= sag, C=1.0 |
| Subset 6 | MLP | Activation = tanh, solver – lbfgs, learning_rate= adaptive, hidden_layer_sizes= (100, 3) |

(Figure S2 in the Supplementary File) and the stacking model associated with a wrapper feature selection step (Figure S3 in the Supplementary File). Table 15 compares our model with two stacking models in terms of the CV accuracy and other testing metrics. In the case of the 6-hour dataset, the proposed stacking model achieved values of F1= 0.911 and AUC= 0.919. Compared to the traditional stacking model, the proposed model achieved increments of 0.038 and 0.026 in F1 and AUC, respectively. Compared to the stacking model with feature selection, it achieved increments of 0.022 and 0.027 in F1 and AUC, respectively. In the case of the 12-hour dataset, the proposed model achieved values of F1 = 0.923 and AUC = 0.920.

Compared to the traditional stacking model, the proposed model achieved increments of 0.022 and 0.031 in F1 and AUC, respectively; compared to the stacking model with feature selection, the proposed model achieved increments of 0.053 and 0.032 in F1 and AUC, respectively. In the case of

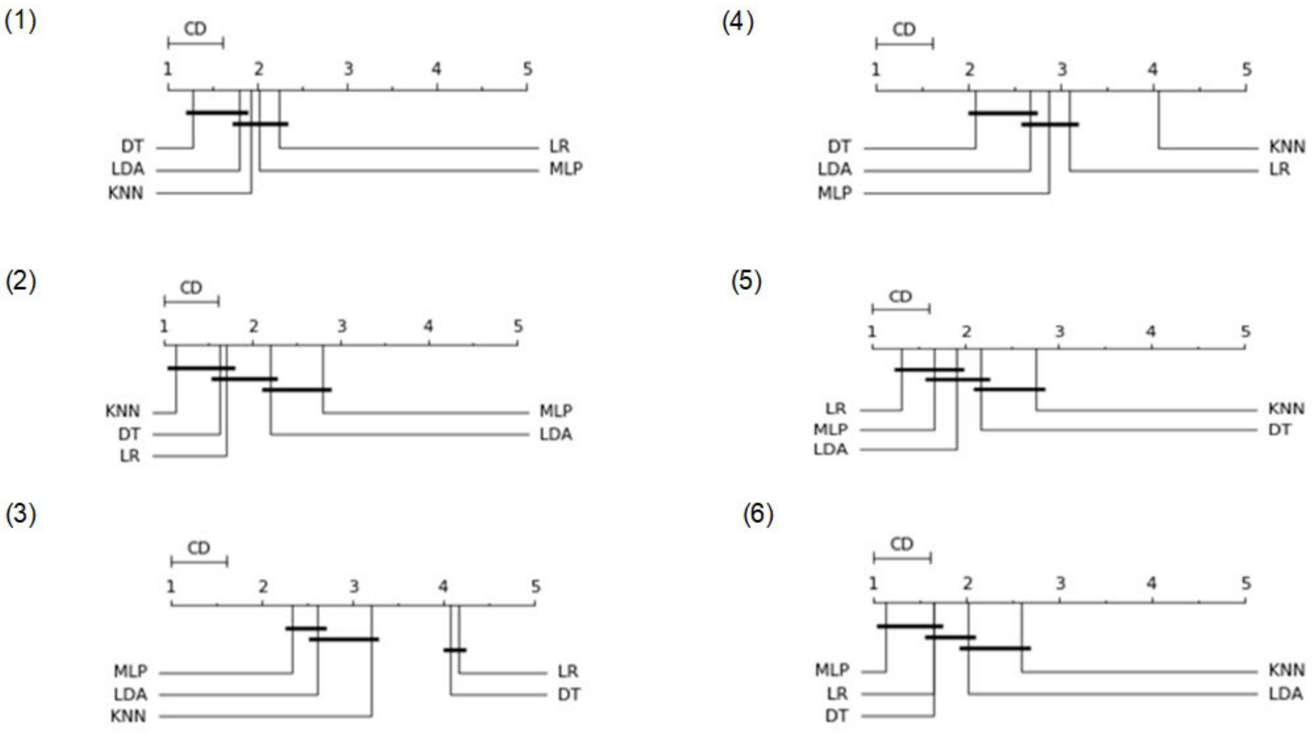


**FIGURE 10.** Comparison of all classifiers based on the Nemenyi testfor each subset. Numbered from 1 to 6 according to feature subset number, from subset 1 to 6. The groups of classifiers are significantly different ($p<0.005$).

**TABLE 15.** The effect of no feature selection, feature selection, and proposed domain expert-based data splitting on the stacking technique.

| Classifier | Time frame | CV Accuracy | F1 | P | R | ACC | SP | AUC |
|---|---|---|---|---|---|---|---|---|
| Traditional stacking | 06 Hours | 0.900±0.032 | 0.873 | 0.884 | 0.863 | 0.895 | 0.918 | 0.890 |
| | 12 Hours | 0.901±0.061 | 0.901 | 0.927 | 0.870 | 0.913 | 0.945 | 0.899 |
| | 24 Hours | 0.928±0.028 | 0.912 | 0.934 | 0.890 | 0.926 | 0.952 | 0.911 |
| Stacking with feature selection | 06 Hours | 0.903±0.013 | 0.873 | 0.880 | 0.860 | 0.895 | 0.915 | 0.891 |
| | 12 Hours | 0.911±0.078 | 0.870 | 0.873 | 0.868 | 0.892 | 0.909 | 0.888 |
| | 24 Hours | 0.931±0.011 | 0.881 | 0.862 | 0.901 | 0.901 | 0.923 | 0.919 |
| Stacking (proposed Model) | 06 Hours | 0.932±0.042 | 0.911 | 0.936 | 0.882 | 0.928 | 0.921 | 0.917 |
| | 12 Hours | 0.941±0.033 | 0.923 | 0.954 | 0.872 | 0.938 | 0.935 | 0.920 |
| | **24 Hours** | **0.957±0.089** | **0.937** | **0.964** | **0.911** | **0.944** | **0.940** | **0.933** |

the 24-hour dataset, the proposed model achieved values of F1 = 0.937 and AUC = 0.933. Compared to the traditional stacking model, the proposed model achieved increments of 0.025 and 0.022 in F1 and AUC, respectively. Finally, compared to the stacking model with feature selection, the proposed model achieved increments of 0.056 and 0.014 in F1 and AUC, respectively.

From the previous experiments, we observed that: (i) feature set C includes important critical features that lead to an improvement in mortality prediction for all the time frames; (ii) not all base classifiers are suitable for all subsets; therefore, choosing the most accurate classifier for each subset enhances the overall performance of the proposed ensemble classifier; (iii) the results of the first 6 h are comparable to those of the first 12 h owing to the large percentage of missing features in the first 12 h; (iv) the most accurate results are obtained by using the 24-hour dataset; (v) our proposed ensemble classifier outperforms all the base classifiers; and (vi) the results indicate that a well-built heterogeneous ensemble classifier can outperform any other classifier in terms of mortality prediction. Figure 11 compares the ROC performance of all the stacking algorithms with time frames of 24 hours.

To ensure the superiority of our proposed ensemble stacking model. We compare all models, including single classifiers and ensemble classifiers, with our proposed model using the Nemenyi test. The rank is based on the accuracy of the classifiers. Figure 12 shows the results of the average rank using the critical differences diagram. The critical differences were calculated using the Nemenyi, after comparing all models against each other based on accuracy and Freidman test. The test asserted a significant difference among classifiers (statistics = 8.89, $p < 0.005$). As shown in the figure, using KNN with wrapper feature selection gives

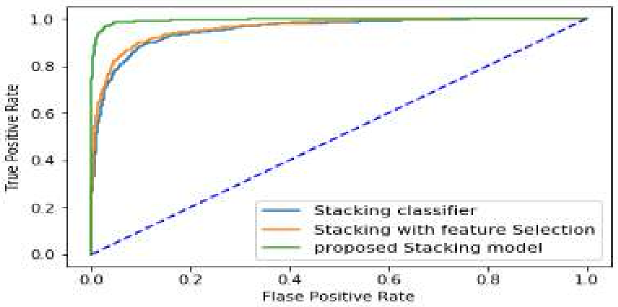


**FIGURE 11.** AUC for the stacking classifiers.

the worst performance (ACC = 0.840, $p < 0.005$) and MLP gives the best performance (ACC = 0.902, $p < 0.005$) among all base classifiers. Regarding the ensemble classifiers, the RF classifier presents the worst significance performance (ACC= 0.851, $p < 0.005$) followed by bagging (ACC = 0.911, $p < 0.005$). Boosting statistically outperformed all other ensemble classifiers (ACC = 0.929, $p < 0.005$). The proposed stacking ensemble classifier statistically significantly outperformed all base and other ensemble classifiers (ACC = 0.944, $p < 0.005$). Thus, we can conclude that our proposed stacking ensemble classifier shows a significant performance gain over other classifiers using the same feature set.

#### 4) THE ROLE OF THE MOST CRITICAL FEATURES

Studying the importance of the features is critical from the ML point of view [77]. Generally, feature importance provides a specific score for each feature, and these scores indicate the effect of each feature on the model performance. Accordingly, we calculated the feature importance using three different techniques: information gain (IG), correlation coefficient (CC), and RF. Each technique returns a specific rank for each feature. Table S4 in the supplementary file details the importance of all features according to these algorithms. Based on these scores, we can observe that age, heart rate, level of conscious, GSC score, alarm[high], respiratory rate, WBC, SpO2 alarm [high], motor response, heart rhythm, non-Invasive Blood Pressure, and temperature are the top contributing features to the mortality prediction. To study the impact of the most important features on the model performance, we conducted 12 different experiments, where in each experiment, we excluded one of these features and checked the model performance. Table S7 in the Supplementary File shows the details of all experiments.

## VI. DISCUSSION

Our proposed ensemble classifier achieved this performance because (1) all the feature subsets are preprocessed in terms of missing values, outliers, and normalizations; (2) the three feature sets are medically divided into six feature subsets; (3) all the subsets are tested on all the classifiers, and then, the most efficient models are selected for each set; and (4) the stacking model are used to fuse the decisions of these diverse and accurate models. In this section, we closely examine the performance of the proposed stacking model, in comparison with that of the traditional scoring systems, the base classifiers, and the literature studies in the ICU mortality prediction domain.

### A. PROPOSED MODEL VS. THE TRADITIONAL SCORING SYSTEM

In this section, we compare the performance of the proposed model against that of the traditional scoring systems, single classifier models, and ensemble techniques. First, we implement several scoring systems that are commonly used in the ICU (SAPS II, APACHE II, and SOFA). We calculate each score on our dataset after doing all the preprocessing steps. For example, the SAPS II scoring system formula calculates the patient's score according to the degrees of measurements. The score ranges from 0 to 160 points, where a higher number indicates a higher risk. For additional clarity, Table S5 in the Supplementary File details the steps for performing these calculations. As presented in Table 15, SAPS II delivered the best performance (F1 = 0.772 and AUC = 0.812). We observe that most of the scoring systems

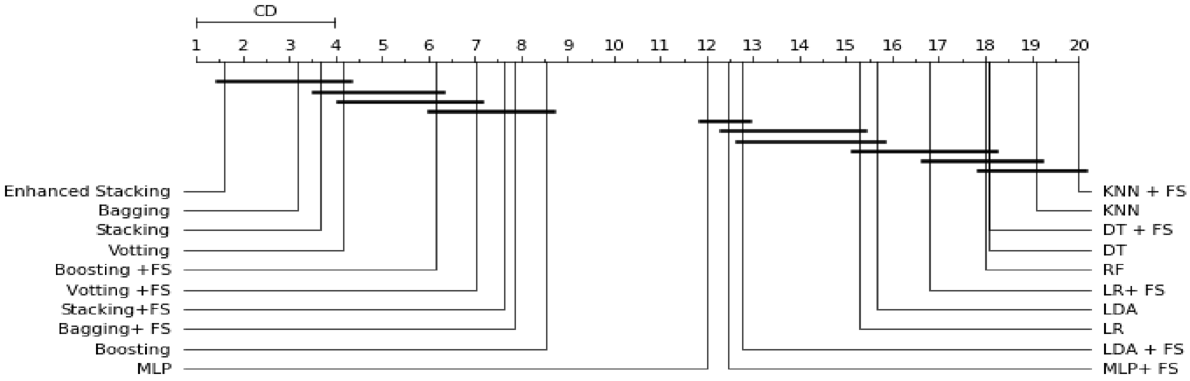


**FIGURE 12.** Comparison of all classifiers based on Nemenyi test applied on the whole feature set. Groups of classifiers that are significantly different (*p<0.005).*

**TABLE 16.** Score performance results for all the models.

| Method | Algorithm | First 24 Hours | | | | |
|---|---|---|---|---|---|---|
| | | CV accuracy | F1 | P | R | AUC |
| Score method | Apache II | - | 0.721 | 0.8985 | 0.602 | 0.734 |
| | SAPS- II | - | 0.772 | 0.7720 | 0.767 | 0.812 |
| | SOFA | - | 0.733 | 0.752 | 0.708 | 0.782 |
| Single Model | KNN | 0.917±0.021 | 0.904 | 0.916 | 0.891 | 0.892 |
| | DT | 0.811±0.012 | 0.876 | 0.866 | 0.868 | 0.832 |
| | LR | 0.880±0.038 | 0.890 | 0.905 | 0.861 | 0.898 |
| | LDA | 0.901±0.098 | 0.902 | 0.894 | 0.872 | 0.891 |
| | MLP | 0.929±0.072 | 0.912 | 0.922 | 0.893 | 0.901 |
| Ensemble Model | RF | 0.867±0.015 | 0.776 | 0.899 | 0.752 | 0.842 |
| | Bagging | 0.922±0.007 | 0.901 | 0.940 | 0.903 | 0.916 |
| | AdaBoost | 0.935±1.105 | 0.911 | 0.933 | 0.900 | 0.904 |
| | Voting | 0.899±0.009 | 0.923 | 0.959 | 0.919 | 0.906 |
| Proposed Model | Stacking | 0.957±0.089 | 0.937 | 0.964 | 0.911 | 0.933 |

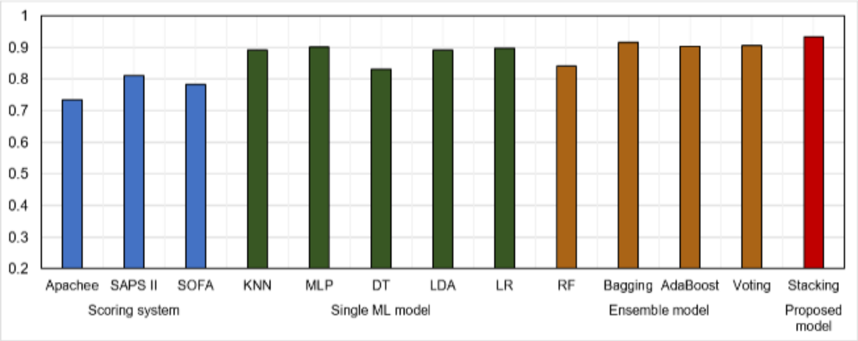


**FIGURE 13.** AUC scores of all the models evaluated in the study for a period of 24 hours.

are similar in terms of the performance, which confirms the insights of other researchers [60]. Our proposed model relies on an advanced ensemble classifier that produces a more sophisticated model. This model is capable of handling various types of data and has a greater classification power than that of the traditional scoring systems. Our proposed ensemble model outperforms all the scoring systems by 7–19% in terms of accuracy. As the previous discussion asserted, feature set C achieved the best results for all the base classifiers and ensembles. Table 16 compares the best models from each category. Table 16 indicates that MLP delivers the best performance ($F1 = 0.912$ and $AUC = 0.901$) when using single classifiers. Using ensemble classifiers provides some improvements, and the best performance is achieved using both bagging ($F1 = 0.901$ and $AUC = 0.916$) and boosting ($F1 = 0.911$ and $AUC = 0.904$). Our proposed heterogeneous ensemble classifier produces the best results. It outperforms single classifiers by 4–16% ($F1 = 0.937$ and $AUC = 0.933$). Figure 13 presents an AUC-based comparison between the results of the scoring system, single classifier's model, ensemble models, and our proposed model.

### B. PROPOSED MODEL VS. LITERATURE STUDIES

As discussed in Section 2, numerous techniques have been proposed to solve the mortality-prediction problem. However, most suffer from limitations that can be summarized as follows:

1. Use of less-informative feature sets. Some studies only depend on the vital signs to predict mortality, as these signs are the most frequently recorded features [75], [81]. Others use only specific measurements such as laboratory-test results [55]. These features may provide acceptable results for specific patient categories but cannot be used in general for the ICU. In this study, based on the decision of our domain expert, we considered selecting features related to risks in most critical diseases. For example, for renal failure patients, we added Blood Urea Nitrogen (BUN) and Urea Nitrogen. For patients with heart problems, we added creatinine kinase, heart rims, and heart cardiac) enzymes. We used albumin for liver-failure patients, partial thromboplastin time (PTT) for hemophilia diseases, etc.
2. Limitation of determining the time window to extract features. Many studies work only on the 6-hour dataset to provide a model that can offer an early prediction. In this study, we choose to work with three-time windows (6, 12, and 24 hours). The most accurate result was then determined (i.e. features extracted after 24 hours). This is because many features may not be recorded in the first 12 h.
3. Most literature studies depend on single ML algorithms to analyze the patient's patients' data and predict future events [17], [76]. This may not be suitable for ICU

**TABLE 17. Comparison with the literature.**

| Study | Data set | Sample | Year | Used algorithms | Features | Model | # hours | Accuracy |
|---|---|---|---|---|---|---|---|---|
| [4] | MIMIC III | 3000 | 2018 | SVM, K-NN | 12 | single | 6 | 0.91 |
| [38] | MIMIC II | 13164 | 2019 | NN | 17 | Single | 6-8 | 0.89 |
| [51] | MIMIC II | 4000 | 2018 | LR, ELM | 6 | Single | 24 | 0.90 |
| [59] | MIMIC II | 4000 | 2012 | Bayesian model | 10 | Ensemble | 12 | 0.86 |
| [82] | Private | 50 | 2018 | SVM, RBF | 22 | Single | 6 | 0.81 |
| [54] | Private | 400 | 2019 | JIT Learning | 24 | Single | 12 | 0.78 |
| [83] | MIMIC II | 9600 | 2014 | Developed algorithm | 8 | Single | 12 | 0.92 |
| [58] | MIMIC-III | 22.413 | 2019 | ConvNet | 27 | ConvNet | 3-6-12 | 0.87 |
| [62] | MIMIC-II | 6400 | 2018 | DNN | | DL | 24.48 | 0.93 |
| Proposed model | MIMIC III | 10664 | - | Stacking | 80 | Ensemble | 6-12-24 | 0.944 |

data as they vary in types and ranges. Therefore, we compare the results obtained when using common base classifiers and ensemble classifiers. To improve the efficiency of the ensemble techniques, we improved the accuracy and diversity of a customized stacking technique. We vertically divided the best-performing dataset into six subsets according to medical expert opinion. Then, we selected the most suitable algorithm for each subset, which was used to develop our proposed stacking model. In our study, to guarantee the accuracy of the developed model, we completely separated the training and testing data in advance. In terms of the classification accuracy, the accuracy of the proposed algorithm was considerably improved, compared to the state-of-the-art techniques.

Table 17 compares the proposed stacking model and the literature models in terms of sample size, used algorithm, number of features, number of hours, and accuracy. As can be observed, our model considers all features related to critical cases, not only the vital signs. We used 80 features from vital signs, patient demographics, and lab tests. Regarding time frames, we extracted data within three different time frames (6, 12, and 24 hours). The best performance was obtained when we used a period of 24 hours. Regarding accuracy, our proposed ensemble model achieved the best performance; it outperformed the state-of-the-art models. Most of the related literature is based on single classifiers that achieve results inferior to those of our model. The proposed model is acceptable because it was designed with the guidance received from a medical expert. The effectiveness of the heterogeneous stacking ensemble classifier in overcoming challenges related to the ICU heterogeneous data was confirmed.

## VII. CONCLUSION AND FUTURE WORK

This paper has presented a heterogeneous and medically intuitive ensemble classifier for ICU mortality prediction. This is a binary classification task. The model was based on a set of five well-known algorithms that are commonly used in the medical domain: KNN, LR, LDA, MLP, and DT. The most critical features related to ICU mortality prediction were collected with the help of a medical expert. Our study was based on the MIMIC III benchmark data. We utilized the time-series data of 80 features from 10,664 patients. These data were collected for the first 6, 12, and 24 h. To be considered reliable and more intuitive, mortality prediction was tested after one hour from the end of the training time window. Extensive experiments were performed using these features, the previous baseline ML algorithms, and the standard ensemble techniques of bagging, boosting, and voting. With the guidance of a medical expert, the best feature list was divided into six different subsets. A comprehensive analysis was conducted to select the best model for each subset of features. Finally, we developed our proposed stacking ensemble classifier by using the LR as a meta learner. The performance of the proposed ensemble model was evaluated and compared with that of traditional scoring systems for mortality prediction, single classifier models, and traditional ensemble models. The evaluation process was completed using the K-fold CV. Our proposed ensemble classifier achieved an encouraging performance ($F1 = 0.937$, $ACC = 0.944$, and $AUC = 0.933$) that improved the accuracy of the state-of-the-art studies by 2–3%. The proposed model is medically more intuitive because it is based on a comprehensive list of patient features. In addition, the model was designed under the guidance of a medical expert. In the future, we will extend our model to deal with other types of data such as EEG and ECG. We will explore the role of deep learning models for dealing with the time series data. Deep learning models such as LSTM and CNN are popular to hand the longitudinal data. Next, we will investigate the role of multitask modeling to enhance model stability and the prediction of multiple related tasks.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF-2016R1D1A1A03934816).

## REFERENCES

[1] Z. Rayan, M. Alfonse, and A. M. Salem, *Intensive Care Unit (ICU) Data Analytics Using Machine Learning Techniques*, vol. 26, no. 1. Sofia, Bulgaria: ITHEA, 2019, pp. 69–82.

[2] G. Rouleau, M.-P. Gagnon, and J. Côté, "Impacts of information and communication technologies on nursing care: An overview of systematic reviews (protocol)," *Systematic Rev.*, vol. 4, no. 1, pp. 1–8, Dec. 2015.

[3] K. M. D. M. Karunarathna, "Predicting ICU death with summarized patient data," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2018, pp. 238–247.

[4] R. Sadeghi, T. Banerjee, and W. Romine, "Early hospital mortality prediction using vital signals," *Smart Health*, vols. 9–10, pp. 265–274, Dec. 2018.

[5] C. Lee and T. Rainer, ''Application of APACHE II in the assessment, classification of severity and predictive ability of chinese patients presenting to an emergency department resuscitation room,'' *Hong Kong J. Emergency Med.*, vol. 9, no. 4, pp. 188–194, Oct. 2002.

[6] S. Jeong, ''Scoring systems for the patients of intensive care unit,'' *Acute Crit. Care*, vol. 33, no. 2, pp. 102–104, 2018.

[7] D. Sun, H. Ding, C. Zhao, Y. Li, J. Wang, J. Yan, and D. W. Wang, ''Value of SOFA, APACHE IV and SAPS II scoring systems in predicting short-term mortality in patients with acute myocarditis,'' *Oncotarget*, vol. 8, no. 38, pp. 63073–63083, Sep. 2017.

[8] M. K. Moridani, S. K. Setarehdan, A. M. Nasrabadi, and E. Hajinasrollah, ''Analysis of heart rate variability as a predictor of mortality in cardiovascular patients of intensive care unit,'' *Biocybern. Biomed. Eng.*, vol. 35, no. 4, pp. 217–226, 2015.

[9] R. Pirracchio, M. J. Cohen, I. Malenica, J. Cohen, A. Chambaz, M. Cannesson, C. Lee, M. Resche-Rigon, and A. Hubbard, ''Big data and targeted machine learning in action to assist medical decision in the ICU,'' *Anaesthesia Crit. Care Pain Med.*, vol. 38, no. 4, pp. 377–384, Aug. 2019.

[10] B. Graham, R. Bond, M. Quinn, and M. Mulvenna, ''Using data mining to predict hospital admissions from the emergency department,'' *IEEE Access*, vol. 6, pp. 10458–10469, 2018.

[11] T. Gentimis, A. J. Alnaser, A. Durante, K. Cook, and R. Steele, ''Predicting hospital length of stay using neural networks on MIMIC III data,'' in *Proc. IEEE 15th Intl Conf Dependable, Autonomic Secure Comput., 15th Intl Conf Pervas. Intell. Comput., 3rd Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr.(DASC/PiCom/DataCom/CyberSciTech)*, Nov. 2017, pp. 1194–1201.

[12] Q. Mao, M. Jay, J. L. Hoffman, J. Calvert, C. Barton, D. Shimabukuro, L. Shieh, U. Chettipally, G. Fletcher, Y. Kerem, and Y. Zhou, ''Multicentre validation of a sepsis prediction algorithm using only vital sign data in the emergency department, general ward and ICU,'' *BMJ Open*, vol. 8, no. 1, pp. 1–11, 2018.

[13] M. Carrara, G. Baselli, and M. Ferrario, ''Mortality prediction model of septic shock patients based on routinely recorded data,'' *Comput. Math. Methods Med.*, vol. 2015, pp. 1–7, Jan. 2015.

[14] J. Liu, X. X. Chen, L. Fang, J. X. Li, T. Yang, Q. Zhan, K. Tong, and Z. Fang, ''Mortality prediction based on imbalanced high-dimensional ICU big data,'' *Comput. Ind.*, vol. 98, pp. 218–225, Jun. 2018.

[15] N. M. Arzeno, K. A. Lawson, S. V. Duzinski, and H. Vikalo, ''Designing optimal mortality risk prediction scores that preserve clinical knowledge,'' *J. Biomed. Informat.*, vol. 56, pp. 145–156, Aug. 2015.

[16] R. S. Anand, P. Stey, S. Jain, D. R. Biron, H. Bhatt, K. Monteiro, E. Feller, M. L. Ranney, I. N. Sarkar, and E. S. Chen, ''Predicting mortality in diabetic ICU patients using machine learning and severity indices,'' *AMIA Summits Transl. Sci. Proc.*, vol. 2017, pp. 310–319, May 2018.

[17] I. M. Ball, S. M. Bagshaw, K. E. A. Burns, D. J. Cook, A. G. Day, P. M. Dodek, D. J. Kutsogiannis, S. Mehta, J. G. Muscedere, H. T. Stelfox, A. F. Turgeon, G. A. Wells, and I. G. Stiell, ''A clinical prediction tool for hospital mortality in critically ill elderly patients,'' *J. Crit. Care*, vol. 35, pp. 206–212, Oct. 2016.

[18] S. Ghose, J. Mitra, S. Khanna, and J. Dowling, ''An improved patient-specific mortality risk prediction in ICU in a random forest classification framework,'' *Stud. Health Technol. Inform.*, vol. 214, pp. 56–61, Aug. 2015.

[19] R. A. Taylor, J. R. Pare, A. K. Venkatesh, H. Mowafi, E. R. Melnick, W. Fleischman, and M. K. Hall, ''Prediction of in-hospital mortality in emergency department patients with sepsis: A local big data–driven, machine learning approach,'' *Academic Emergency Med.*, vol. 23, no. 3, pp. 269–278, 2015.

[20] A. B. Nielsen, A. P. Nielsen, B. S. Kaas-Hansen, P. Toft, J. Schierbeck, and T. Strøm, ''Articles Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: A retrospective study of high-frequency data in electronic patient records,'' *Lancet Digit. Health*, vol. 2, pp. 1–4, Apr. 2020.

[21] T. Alves, A. Laender, A. Veloso, and N. Ziviani, ''Dynamic prediction of ICU mortality risk using domain adaptation,'' *Proc. IEEE Int. Conf. Big Data, Big Data*, Dec. 2018, pp. 1328–1336.

[22] L. I. Kuncheva and E. Alpaydin, *Combining Pattern Classifiers: Methods and Algorithms*, vol. 18, no. 3, 2nd ed. Hoboken, NJ, USA: Wiley, 2014.

[23] B. Sluban and N. Lavrać, ''Relating ensemble diversity and performance: A study in class noise detection,'' *Neurocomputing*, vol. 160, pp. 120–131, Jul. 2015.

[24] L. Breiman, ''Pasting small votes for classification in large databases and on-line,'' *Mach. Learn.*, vol. 36, nos. 1–2, pp. 85–103, 1999.

[25] R. Matteo and V. Giorgio, *Ensemble Methods: A Review*. London, U.K.: Chapman & Hall, 2012.

[26] L. Rokach, ''Ensemble-based classifiers,'' *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 1–39, 2010.

[27] P. Kayal and S. Kannan, ''An ensemble classifier adopting random subspace method based on fuzzy partial mining,'' *Indian J. Sci. Technol.*, vol. 10, no. 12, pp. 1–8, May 2017.

[28] Y. Tounsi, L. Hassouni, and H. Anoun, ''An enhanced comparative assessment of ensemble learning for credit scoring,'' *J. Intell. Comput.*, vol. 10, no. 1, p. 15, Mar. 2019.

[29] M. A. King, ''Ensemble learning techniques for structured and unstructured data,'' Ph.D. dissertation, Dept. Bus. Inf. Technol., Virginia Polytech. Inst. State Univ., Blacksburg, VA, USA, 2015.

[30] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, ''Recurrent neural networks for multivariate time series with missing values,'' *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, Dec. 2018.

[31] P. D. Gopalan and S. Pershad, ''Decision-making in ICU—A systematic review of factors considered important by ICU clinician decision makers with regard to ICU triage decisions,'' *J. Crit. Care*, vol. 50, pp. 99–110, Apr. 2019.

[32] A. P. James and B. V. Dasarathy, ''A review of feature and data fusion with medical images,'' in *Multisensor Data Fusion, From Algorithms and Architectural Design to Applications*. Leiden, The Netherlands: CRC Press, 2017, pp. 491–507.

[33] R. Polikar, ''Ensemble based systems in decision making,'' *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–44, 3rd Quart., 2006.

[34] S. El-Sappagh, T. Abuhmed, S. M. Riazul Islam, and K. S. Kwak, ''Multimodal multitask deep learning model for Alzheimer's disease progression detection based on time series data,'' *Neurocomputing*, vol. 412, pp. 197–215, Oct. 2020.

[35] S. El-Sappagh, M. Elmogy, F. Ali, T. Abuhmed, S. M. R. Islam, and K.-S. Kwak, ''A comprehensive medical decision–support framework based on a heterogeneous ensemble classifier for diabetes prediction,'' *Electronics*, vol. 8, no. 6, p. 635, Jun. 2019.

[36] W.-H. Hsieh, D.-H. Shih, P.-Y. Shih, and S.-B. Lin, ''An ensemble classifier with case-based reasoning system for identifying Internet addiction,'' *Int. J. Environ. Res. Public Health*, vol. 16, no. 7, p. 1233, Apr. 2019.

[37] L. S. Alistair, E. W. Johnson, T. J. Pollard, ''Data descriptor: MIMIC-III, a freely accessible critical care database,'' *Thromb. Haemost.*, vol. 76, no. 2, pp. 258–262, 2016.

[38] Á. Silva, P. Cortez, M. F. Santos, L. Gomes, and J. Neves, ''Mortality assessment in intensive care units via adverse events using artificial neural networks,'' *Artif. Intell. Med.*, vol. 36, no. 3, pp. 223–234, Mar. 2006.

[39] B. Balkan, P. Essay, and V. Subbian, ''Evaluating ICU clinical severity scoring systems and machine learning applications: APACHE IV/IVa case study,'' in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 4073–4076.

[40] J. L. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. K. Reinhart, P. Suter, and L. G. Thijs, ''The SOFA (sepsis-related organ failure assessment) score to describe organ dysfunction/failure. On behalf of the working group on sepsis-related problems of the European society of intensive care medicine,'' *Intensive Care Med.*, vol. 22, no. 7, pp. 707–710, 1996.

[41] X. Jiang, Z. Su, Y. Wang, Y. Deng, W. Zhao, K. Jiang, and C. Sun, ''Prognostic nomogram for acute pancreatitis patients: An analysis of publicly electronic healthcare records in intensive care unit,'' *J. Crit. Care*, vol. 50, pp. 213–220, Apr. 2019.

[42] Y. Arabi, N. Al Shirawi, Z. Memish, S. Venkatesh, and A. Al-Shimemeri, ''Assessment of six mortality prediction models in patients admitted with severe sepsis and septic shock to the intensive care unit: A prospective cohort study,'' *Crit. Care*, vol. 7, no. 5, pp. R116–R122, 2003.

[43] S. L. Javan, M. M. Sepehri, M. Layeghian Javan, and T. Khatibi, ''An intelligent warning model for early prediction of cardiac arrest in sepsis patients,'' *Comput. Methods Programs Biomed.*, vol. 178, pp. 47–58, Sep. 2019.

[44] M. M. Masud and A. R. Al Harahsheh, ''Mortality prediction of ICU patients using lab test data by feature vector compaction & classification,'' in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 3404–3411.

[45] C.-W. Cheng, N. Chanani, J. Venugopalan, K. Maher, and M. D. Wang, ''IcuARM–An ICU clinical decision support system using association rule mining,'' *IEEE J. Transl. Eng. Health Med.*, vol. 1, 2013, Art. no. 4400110.

[46] S. Kim, W. Kim, and R. Woong Park, "A comparison of intensive care unit mortality prediction models through the use of data mining techniques," *Healthc. Inform. Res.*, vol. 17, no. 4, pp. 232–243, 2011.

[47] G. Thabut, I. Vinatier, J. B. Stern, G. Lesèche, P. Loirat, M. Fournier, and H. Mal, "Primary graft failure following lung transplantation: Predictive factors of mortality," *Chest*, vol. 121, no. 6, pp. 1876–1882, 2002.

[48] O. Luaces, F. Taboada, G. M. Albaiceta, L. A. Domínguez, P. Enríquez, and A. Bahamonde, "Predicting the probability of survival in intensive care unit patients from a small number of variables and training examples," *Artif. Intell. Med.*, vol. 45, no. 1, pp. 63–76, Jan. 2009.

[49] R. Dybowski, V. Gant, P. Weller, and R. Chang, "Prediction of outcome in critically ill patients using artificial neural network synthesised by genetic algorithm," *Lancet*, vol. 347, no. 9009, pp. 1146–1150, Apr. 1996.

[50] K. Lin, Y. Hu, and G. Kong, "Predicting in-hospital mortality of patients with acute kidney injury in the ICU using random forest model," *Int. J. Med. Informat.*, vol. 125, pp. 55–61, May 2019.

[51] S. Bayrak, Y. Doğan, R. Yılmaz, and A. Kut, "Intensive care unit—Clinical decision support system," in *Proc. 10th Int. Conf. Adv. Semantic Process. (SEMAPRO)*, 2016, pp. 41–44.

[52] G. Carioli, P. Bertuccio, P. Boffetta, F. Levi, C. La Vecchia, E. Negri, and M. Malvezzi, "European cancer mortality predictions for the year 2020 with a focus on prostate cancer," *Ann. Oncol.*, vol. 31, no. 5, pp. 650–658, May 2020.

[53] A. R. M. Forkan and I. Khalil, "A probabilistic model for early prediction of abnormal clinical events using vital sign correlations in home-based monitoring," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2016, pp. 1–9.

[54] Y. Ding, Y. Wang, and D. Zhou, "Mortality prediction for ICU patients combining just-in-time learning and extreme learning machine," *Neurocomputing*, vol. 281, pp. 12–19, Mar. 2018.

[55] G. S. Krishnan and S. K. S., "A novel GA-ELM model for patient-specific mortality prediction over large-scale lab event data," *Appl. Soft Comput.*, vol. 80, pp. 525–533, Jul. 2019.

[56] O. H. Babatunde, L. Armstrong, J. Leng, and D. Diepeveen, "A genetic algorithm-based feature selection," *Int. J. Electron. Commun. Comput. Eng.*, vol. 5, no. 4, pp. 899–905, 2014.

[57] F. Miao, Y.-P. Cai, Y.-X. Zhang, X.-M. Fan, and Y. Li, "Predictive modeling of hospital mortality for patients with heart failure by using an improved random survival forest," *IEEE Access*, vol. 6, pp. 7244–7253, 2018.

[58] W. Caicedo-Torres and J. Gutierrez, "ISeeU: Visually interpretable deep learning for mortality prediction inside the ICU," *J. Biomed. Informat.*, vol. 98, Oct. 2019, Art. no. 103269.

[59] A. E. W. Johnson, N. Dunkley, L. Mayaud, A. Tsanas, A. A. Kramer, and G. D. Clifford, "Patient specific predictions in the intensive care unit using a Bayesian ensemble," *Comput. Cardiol.*, vol. 39, pp. 249–252, Jan. 2012.

[60] A. Awad, M. Bader-El-Den, J. McNicholas, and J. Briggs, "Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach," *Int. J. Med. Informat.*, vol. 108, pp. 185–195, Dec. 2017.

[61] J. Xia, S. Pan, M. Zhu, G. Cai, M. Yan, Q. Su, J. Yan, and G. Ning, "A long short-term memory ensemble approach for improving the outcome prediction in intensive care unit," *Comput. Math. Methods Med.*, vol. 2019, pp. 1–10, Nov. 2019.

[62] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," *J. Biomed. Informat.*, vol. 83, pp. 112–134, Jul. 2018.

[63] N. Kalid, A. A. Zaidan, B. B. Zaidan, O. H. Salman, M. Hashim, and H. Muzammil, "Based real time remote health monitoring systems: A review on patients prioritization and related 'big data' using body sensors information and communication technology," *J. Med. Syst.*, vol. 42, no. 2, p. 30, Feb. 2018.

[64] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: What is it and howdoes it work?" *Int. J. Methods Psychiatric Res.*, vol. 20, no. 1, pp. 40–49, Mar. 2011, doi: 10.1002/mpr.329.

[65] A. Moreo, A. Esuli, and F. Sebastiani, "Distributional random oversampling for imbalanced text classification," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. SIGIR*, 2016, pp. 805–808.

[66] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique SMOTE?: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jan. 2002.

[67] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328.

[68] M. A. Tahir, J. Kittler, K. Mikolajczyk, and F. Yan, "A multiple expert approach to the class imbalance problem using inverse random under sampling a multiple expert approach to the class imbalance problem using inverse random," in *Proc. Int. Workshop Multiple Classifier Syst.*, Jun. 2009, pp. 82–91.

[69] M. Hauskrecht, I. Batal, M. Valko, S. Visweswaran, G. F. Cooper, and G. Clermont, "Outlier detection for patient monitoring and alerting," *J. Biomed. Inform.*, vol. 46, no. 1, pp. 47–55, Feb. 2013, doi: 10.1016/j.jbi.2012.08.004.

[70] C. Curley, R. M. Krause, R. Feiock, and C. V. Hawkins, "Dealing with missing data: A comparative exploration of approaches using the integrated city sustainability database," *Urban Affairs Rev.*, vol. 55, no. 2, pp. 591–615, 2019.

[71] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[72] G. Molenberghs and U. Hasselt, "Models for discrete longitudinal data," in *Models for Discrete Longitudinal Data*. New York, NY, USA: Springer-Verlag, Oct. 2005.

[73] P. Agarwal, M. B. Warner, C. Reichner, and D. G. Lazarous, "Marino's the ICU book, 4th edition," *Ann. Amer. Thorac. Soc.*, vol. 11, no. 6, p. 999, 2014.

[74] B. J. Weled, "Book review: The little ICU book of facts and formulas. Paul 1 Marino MD PhD, with contributions from Kenneth M Sutin MD. Philadelphia: Wolters Kluwer/Lippincott Williams & Wilkins. 2009. Soft cover, illustrated, 781 pages, 48.95," *Respir. Care*, vol. 55, no. 2, pp. 227–228, 2010.

[75] J. Todd, A. Gepp, B. Richards, and B. J. Vanstone, "Improving mortality models in the ICU with high-frequency data," *Int. J. Med. Informat.*, vol. 129, pp. 318–323, Sep. 2019.

[76] V. J. Ribas, J. C. Lopez, A. Ruiz-Sanmartin, J. C. Ruiz-Rodriguez, J. Rello, A. Wojdel, and A. Vellido, "Severe sepsis mortality prediction with relevance vector machines," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2011, pp. 100–103.

[77] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," *Cybern. Inf. Technol.*, vol. 19, no. 1, pp. 3–26, Mar. 2019.

[78] F. A. Faria, J. A. dos Santos, A. Rocha, and R. D. S. Torres, "A framework for selection and fusion of pattern classifiers in multimedia recognition," *Pattern Recognit. Lett.*, vol. 39, pp. 52–64, Apr. 2014.

[79] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[80] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937.

[81] F. Seccareccia, F. Pannozzo, F. Dima, A. Minoprio, A. Menditto, C. Lo Noce, and S. Giampaoli, "Heart rate as a predictor of mortality: The MATISS project," *Amer. J. Public Health*, vol. 91, no. 8, pp. 1258–1263, Aug. 2001.

[82] M. Karimi Moridani and Y. Haghighi Bardineh, "Presenting an efficient approach based on novel mapping for mortality prediction in intensive care unit cardiovascular patients," *MethodsX*, vol. 5, no. 29, pp. 1291–1298, 2018.

[83] I. T. Utami, B. Sartono, and K. Sadik, "Comparison of single and ensemble classifiers of support vector machine and classification tree," *J. Math. Sci. Appl.*, vol. 2, no. 2, pp. 17–20, 2014.

**NORA EL-RASHIDY** received the B.Sc. degree in information systems and the M.Sc. degree from the Faculty of Computers and Information, Mansoura, Egypt, in 2009 and 2016, respectively. She is currently pursuing the Ph.D. degree in machine learning with Mansoura University, Egypt. From 2016 to 2018, she was a Teacher Assistant with the Misr Academy, Mansoura, Egypt. She is also a Teacher Assistant with the Machine Learning and Information Retrieval Department, Faculty of Artificial Intelligence, Kafrelsheikh University, Egypt. Her research interests include artificial intelligence, data science, machine learning, and optimization. In these areas, she has published many articles in major international journals and refereed international conferences. She served as a Reviewer for many journals.

**SHAKER EL-SAPPAGH** received the bachelor's degree in computer science and the master's degree from the Information Systems Department, Faculty of Computers and Information, Cairo University, Egypt, in 1997 and 2007, respectively, and the Ph.D. degree in computer science from Information Systems Department, Faculty of Computers and Information, Mansura University, Mansura, Egypt, in 2015. In 2003, he joined the Department of Information Systems, Faculty of Computers and Information, Minia University, Egypt, as a Teaching Assistant. Since June 2016, he has been with the Department of Information Systems, Faculty of Computers and Information, Benha University, as an Assistant Professor. He is currently a Postdoctoral Fellow with the Centro Singular de Investigación en Tecnoloxías Intelixentes, Universidade de Santiago de Compostela, Santiago, Spain. He has publications in clinical decision support systems and semantic intelligence. His current research interests include machine learning, medical informatics, (fuzzy) ontology engineering, distributed and hybrid clinical decision support systems, semantic data modeling, fuzzy expert systems, and cloud computing. He is a Reviewer for many journals, and he is very interested in the diseases' diagnoses and treatment studies.

**TAMER ABUHMED** received the Ph.D. degree in information and telecommunication engineering from Inha University, in 2012. He is currently an Assistant Professor with the College of Computing, Sungkyunkwan University, South Korea. His research interests include applied cryptography and information security, network security, Internet security, and machine learning and its application to security and privacy problems.

**SAMIR ABDELRAZEK** received the B.Sc. and M.Sc. degrees from the Faculty of Computers and Information, Mansoura University, Mansoura, Egypt, and the Ph.D. degree, in 2013. He has been an Assistant Professor with the Information systems Department, Faculty of Computers and Information, Mansoura University. He is interested in intelligent information systems areas, such as big data, AI, system analytics, intelligent systems, and decision support systems. He has been working on some research projects in healthcare area, E-learning, and management information systems. He has authored/coauthored over 30 research publications in peer-reviewed reputed journals, book chapters, and conference proceedings. His current research interests include data analysis, information systems, machine learning, and pattern recognition. He has served as a Reviewer for various international journals.

**HAZEM M. EL-BAKRY** was born in Mansoura, Egypt, in 1970. He received the B.Sc. degree in electronics engineering and the M.Sc. degree in electrical communication engineering from the Faculty of Engineering, Mansoura University, Egypt, in 1992 and 1995, respectively, and the Ph.D. degree from the University of Aizu, Japan, in 2007. He is currently a Full Professor with the Faculty of Computer Science and Information Systems, Mansoura University, where he is also the Head of the Information Systems Department. His research interests include neural networks, pattern recognition, image processing, biometrics, cooperative intelligent systems, and electronic circuits. In these areas, he has published many articles in major international journals and refereed international conferences. According to academic measurements, now, the H-index of his publications is 30. He holds the U.S. patent (No. 20060098887), in 2006. He received the Japanese Computer and Communication Prize, in April 2006, and the Best Paper Prize in two conferences cited by ACM. He also received the Mansoura University Prize for scientific publication, in 2010 and 2011. He has been selected in WHO Asia 2006 and BIC 100 educators in Africa, in 2008. He is an Associate Editor of the *Journal of Computer Science and Network Security* (IJCSNS) and the *Journal of Convergence in Information Technology* (JCIT). In addition, he is a Referee for the IEEE Transactions on Signal Processing, the *Journal of Applied Soft Computing*, the *International Journal of Machine Graphics and Vision*, the *International Journal of Computer Science and Network Security*, Enformatika journals, WSEAS journals, and many different international conferences organized by the IEEE.