An Ensemble Model for Early Sepsis Prediction using Clinical Records from the Intensive Care Unit (ICU)

Zeina Rayan, Marco Alfonse, Abdel-Badeeh M. Salem

Computer Science Dept, Faculty of Computer & Information Sciences, Ain Shams University Cairo, Egypt.
zeinarayan@cis.asu.edu.eg, marco_alfonse@cis.asu.edu.eg, absalem@cis.asu.edu.eg

Abstract

One of the diseases which is life-threatening is Sepsis. The unbalanced body reaction to some chemicals which is released by the body into its blood stream in response to fighting an infection is the main cause of Sepsis. Early sepsis prediction is a necessity in order to decrease the mortality rates of ICU patients. The accuracy of early prediction of sepsis can be enhanced using the machine learning techniques. This paper presents an Ensemble model that can early predict sepsis. This model is applied to dataset provided by PhysioNet/Computing in Cardiology Challenge 2019. This model achieved an accuracy of 98%.

Keywords: *Predicting Sepsis, Machine Learning, Artificial Intelligence, Intensive Care Unit, Medical Informatics*

1. Introduction

One of the main causes of mortality of the ICU patients is Sepsis which places a huge burden on the health care system [1]. The body's response to an infection causes the Sepsis disease which potentially leads to death [2]. Early prediction of sepsis is a challenging task. It could allow an aggressive and targeted therapy while maintaining antimicrobial stewardship. The methodologies used to detect and predict Sepsis require time-consuming laboratory test results and often suffer from low performance [3].

The tools of clinical decision support, which use clinical values of various tests in ICU, can help with sepsis prediction. Having many clinical values, collected from critically ill patients in the ICU, make a good resource for a lot of research topics [3] [4]. The machine learning techniques can use this huge amount of data to accurately predict Sepsis [4].

The organizers of PhysioNet/Computing in Cardiology (CinC) Challenge 2019 [5] made a challenge to develop methodology for early sepsis detection and prediction using physiological data collected from ICU patients. The aim of this work is to present an effective model that can early predict sepsis in an accurate way.

This paper is constructed as follows: section 2 contains a brief description of the dataset and a comparative study between different machine learning approaches that are applied for sepsis prediction, section 3 presents the proposed sepsis prediction model, section 4 includes the results and discussion, and section 5 contains conclusion and future work.

2. Machine Learning Approaches used in Sepsis Prediction

This section presents a comparative study between recent different machine learning models that are applied to data gathered from the ICU used for predicting sepsis onset. All of the following models uses the datasets of the PhysioNet/CinC challenge 2019 [5]. This section focuses only on the researches that were evaluated using the accuracy metric in order to be able to compare our proposed model performance for sepsis prediction with other models' performance, since our proposed model is evaluated using the accuracy metric.

The data provided by the PhysioNet/CinC challenge 2019 contains three different datasets (A, B, and C). These datasets are collected from three different hospitals (from ICU patients). The datasets (A and B) are publicly available while the dataset (C) is not publically available. Dataset A contains 20,336 patients while dataset B contains 20,000 patients. Each patient has a file in the datasets containing his/her records during the ICU stay where the clinical variables were collected each hour. Each dataset has a total of forty features that are divided into three groups; the first 8 features represent the vital signs, the following 26 features represent the laboratory values, and the last 6 features represent the demographics [5].

Table 1 shows a comparison between different machine learning models that are applied to the PhysioNet/CinC Challenge 2019's datasets for predicting sepsis.

Table 1. Machine Learning prediction models in ICU Sepsis Prediction

Prediction Model	Evaluation
Ensemble of bagged Decision Trees [6]	Accuracies of set A, set B, and set C are 0.871, 0.912, and 0.754
Ensemble of AEC-Net, Random Forest, and Gradient Boosted Decision Tree (GBDT) [7]	Accuracies of set A, set B, and set C are 0.836, 0.894, and 0.709
Ensemble of 4 KNN classifiers [8]	Accuracy = 0.97
Random Forest ensemble [9]	Accuracy = 87.7%
Weight Assignment for each feature then calculating the total weight. if Total Weight > 2 then the record is assigned to class "have sepsis" [10]	Accuracies of set A, set B, and set C are 0.968, 0.978, and 0.984
Ensemble of CNN, and LSTM [11]	Accuracy = 0.927
Logistic Regression [12]	Accuracies of set A, set B, and set C are 0.795, 0.889, and 0.815
EasyEnsemble: an ensemble of Light Gradient Boosted Machine (LightGBMs) [13]	Accuracies of set A, set B, and set C are 0.835, 0.912, and 0.765
Recurrent Neural Network (RNN) [14]	Accuracy of All Sets = 0.854±0.009

From table 1, we can conclude that different ML approaches are used in the proposed models that help in predicting sepsis for ICU patients. These models are applied to many patients' data records with different time intervals. The ensemble of different machine

learning models is the most commonly used approach where it proved to achieve high accuracies in most of the mentioned proposed models (0.877 to 0.968).

3. The Proposed Model of Sepsis Prediction

The goal of our study is to develop a machine learning model that can predict sepsis efficiently. The developed model is applied to dataset A, and dataset B of the PhysioNet/Computing in Cardiology Challenge 2019.This model is represented in figure 1. The first step in this model is data normalization. The second step is handling data unbalancing. The third step is handling missing values. The fourth step is selection of clinical measurements (feature selection). The fifth step is applying supervised machine learning approach, evaluating the model using test set A, and validating the model using dataset B. Each step is explained in the next sub-sections.

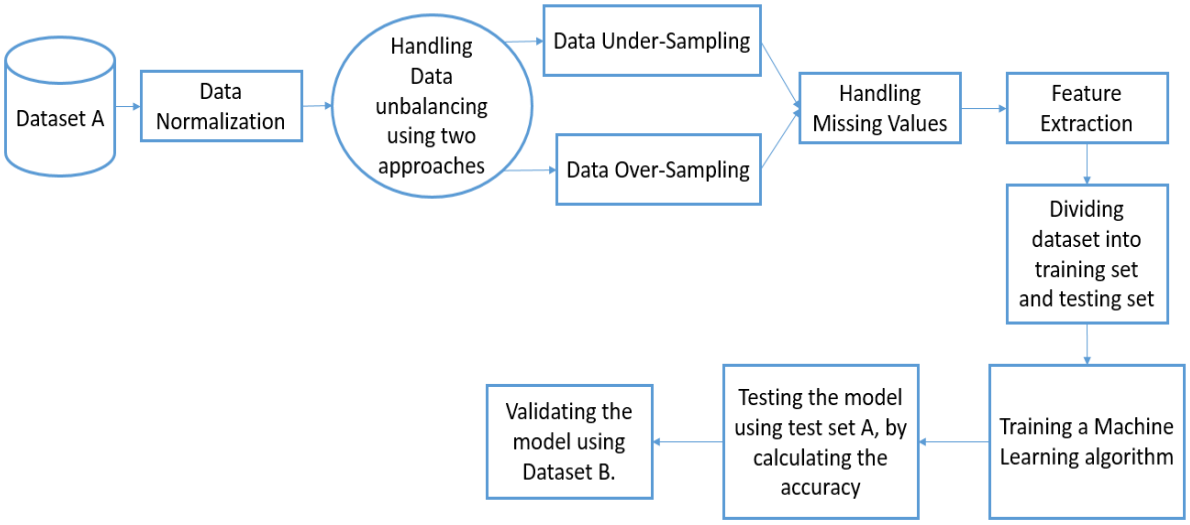

Figure (1): The Proposed Sepsis Prediction Methodology

3.1. Data Normalization

For each input feature, data values were standardized using the z-score method, which subtracts the mean of all values from each feature value, and then divides by the standard deviation of each feature.

3.2. Handling Data Unbalancing

The data was highly unbalanced (only 1.8% of the patient records showed sepsis) and high number of missing values (up to 99.8% in some features) calculated for both of the public datasets [6]. For dataset A, figure 2 shows the data unbalancing where 773079 records do not have sepsis (Value is 0.0), while 17136 records have sepsis (Value is 1.0), so the proportion is 45.11: 1.

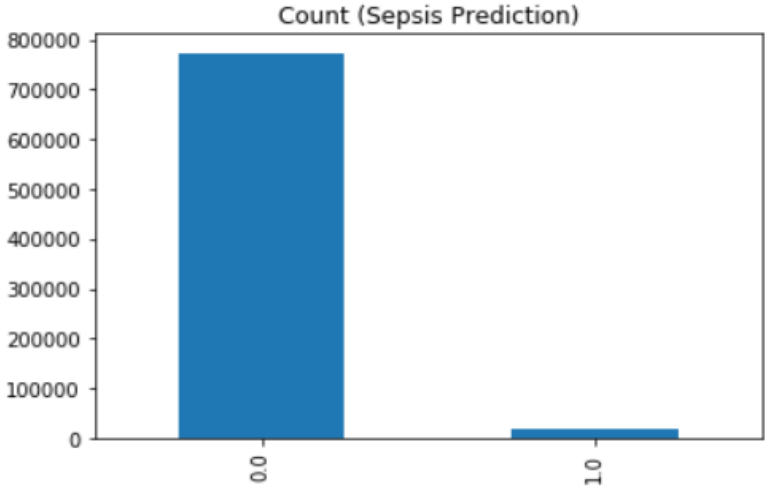

Figure (2): Dataset A, unbalanced sepsis prediction

Dataset unbalancing was handled by two ways. The first way was under-sampling the dataset, which made the 2 classes have same number of records which was the number of the minimum class; this was achieved by removing records from the maximum class. Each class now contains 17136 records. Figure 3 shows dataset A after under-sampling.

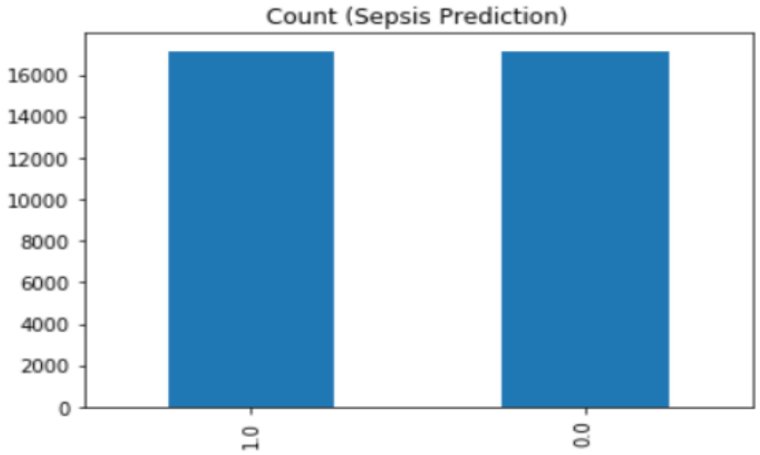

Figure (3): Dataset A, after under-sampling

The second way was over-sampling the dataset, which made the 2 classes have same number of records which was the maximum by repeating same records from the minimum class. Each class now contains 773079 records. Figure 4 shows dataset A after over-sampling.

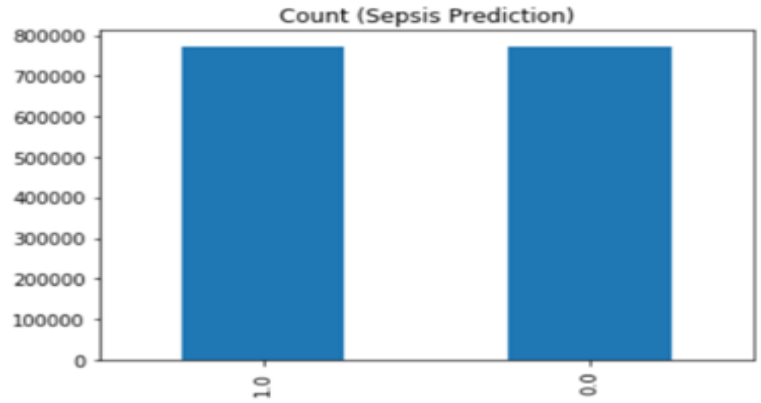

Figure (4): Dataset A, after over-sampling

3.3. Handling Missing Values

The patient records were preprocessed before being used in development of the model, through imputing the missing values by forward filling if a value was available in past. The remaining missing values with no previous values were replaced by the population mean, calculated from the public datasets A and B. The statistical values for each feature after preprocessing is shown in table 2.

Table 2. Machine Learning prediction models in ICU Sepsis Prediction

Feature	Stands For	Unit of measurement	Missing values (%)
HR	Heart Rate	beats per minute	9.9
Temp	Temperature	Degree Celsius	66.2
MAP	Mean Arterial Pressure	mm Hg	12.5
Resp	Respiration rate	breaths per minute	15.4
BaseExcess	Measure of Excess Bicarbonate	mmol/L	94.6
FiO2	Fraction of inspired Oxygen	%	91.7
BUN	Blood Urea Nitrogen	mg/dL	93.1
Calcium		mg/dL	94.1
Creatinine		mg/dL	93.9
Hct	Hematocrit	%	91.1
WBC	Leukocyte count	count*10^3/µL	93.6
Platelets		count*10^3/µL	94.1
HospAdmTime	Hospital Admit Time	Hours between hospital admit and ICU admit	0.0
ICULOS	ICU Length of Stay	hours since ICU admit	0.0

3.4. Feature Selection

A total of 15 features were used for sepsis prediction in [6]. These features are: HR, Temp, MAP, Resp, BaseExcess, FiO2, BUN, Calcium, Creatinine, Hct, WBC, Platelets, HospAdmTime, ICULOS, and a combination between Unit1, and Unit2. In our experiments we utilized the same features excluding Unit1, and Unit2 features, as those features represent an identifier which represent the patient entered the ICU through Medical Intensive Care Unit (MICU) or Surgical Intensive Care Unit (SICU).

3.5. Machine Learning

In this phase, four steps were applied; dividing dataset A into training set and testing set, training different machine learning algorithms using the training set A, evaluating each learning algorithm by calculating the accuracy using the testing subset of set A, and finally validating the models using dataset B.

For the first step, dataset A was divided into training set and testing set, where the testing subset is 20% of the original dataset and the training subset contains 80% of the original dataset. For the second step, the training subset of set A was used to train many supervised machine learning algorithms (either after under-sampling or over-sampling). These algorithms are Naïve Bayes (NB), Logistic Regression (LogR), Linear Discriminant Analysis (LDA), K-Nearest Neighbor (KNN), Decision Tree (DT) classifier, Support Vector Machine (SVM), Ada Boost classifier, Gradient Boosting (GB) classifier, Random Forest (RF) Classifier, Extremely Randomized Trees (Extra Trees) Classifier, Bagging classifier using

KNN and ET Classifier. For the third step, the testing subset of set A was used to evaluate each learning technique by calculating the accuracy. Finally, dataset B was used to validate each classifier.

3.5.1. Naïve Bayes

It is a method that is used as a supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable [15]. There are many types of NB such as Gaussian Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, …etc. The applied NB is the Gaussian NB.

3.5.2. Logistic Regression

LogR is another machine learning algorithm that is based on statistics. Despite its name, LogR is a classification algorithm usually limited to only binary (two-class) classification problems. Logistic regression is named after the function used at the core of its method, the logistic function (also called the sigmoid function) [16]. Three types of logistic Regression [17]; Binary LogR model, for two classes classification, multinomial LogR where the classification problem has more than two classes, and ordinal LogR where the multiple classes are ordered. The applied LogR algorithm is a binary LogR model where the L2 penalty is used [18].

3.5.3. Linear Discriminant Analysis

The Linear Discriminant Analysis is the preferred technique for linear classification. LDA is a classifier uses a linear decision boundary, which is formulated by fitting class conditional densities to the input data using Bayes' theorem. The model generates a Gaussian density for each label. LDA can also be used for reducing the dimensions of the input features [19]. The applied LDA gets the most discriminate features out of the fourteen input features.

3.5.4. K-Nearest Neighbor

KNN is a supervised machine learning algorithm. The idea behind the nearest neighbor method is to find a predefined number of training samples closest in distance to the new testing sample, and predict the label from these nearest K-samples. K is a constant defined by the user [20]. The applied KNN is tested through multiple Ks; 5, 7, and 9. The used values for Ks are odd numbers in order to avoid ties.

3.5.5. Support Vector Machine

SVM is a supervised machine learning technique used for classification, regression and outliers' detection. Its aim is to produce a hyperplane; this hyperplane could distinctly classify the dataset samples in a N-dimensional space where N is the number of features. SVM uses kernel function to transform the input features to a different representation space. Common kernels are mostly used, such as linear function, polynomial function, and Radial Basis Function (RBF). However, it is also possible to specify custom kernels [21]. The applied kernel for the SVM model is RBF since the linear, and polynomial kernels do not fit well with the input dataset.

3.5.6. Decision Tree Classifier

DT classifier is a non-parametric predictive supervised machine learning approach used for classification and regression. The objective is to generate a model (tree) that can predict the class of a sample by learning simple decision rules generated from the dataset features.

The generated tree is produced through partitioning the dataset (binary recursive partitioning). Dividing the dataset into partitions is an iterative process which splits it up further on each of the branches [22]. The applied DT classifier uses an optimized version of the Classification and Regression Trees (CART) algorithm which builds binary trees using the variable and a threshold that gets the maximum information gain at each node [23].

3.5.7. Ada Boost Classifier

Ada Boost [24] classifier is an ensemble of supervised machine learning models that is a boosting ensemble method. It begins by training a base classifier on the input dataset and then trains more copies of that classifier on the same dataset, where the weights of misclassified records are modified such that following classifiers focus more on difficult samples. The applied Ada Boost classifier implements the algorithm known as AdaBoost-SAMME [25].

3.5.8. Gradient Boosting Classifier

GB or Gradient Boosted Decision Trees (GBDT) classifier is another ensemble of supervised machine learning models which is a boosting ensemble method. It can be used for regression and classification problems. It is a weighted sum of weak classifiers which transforms the problem to a gradient descent problem [26].The applied GB classifier uses 100 weak classifiers. It uses a special version of mean square error function to measure the split quality [27].

3.5.9. Random Forest

Random forest classifier, also called random decision forests are an ensemble learning method where many Decision Trees (a forest) are used for classification [28]. The applied model fits 100 Decision Tree classifiers on different subsets of the dataset where a random subset of input features is used. Each tree in the ensemble is built from a sample drawn with replacement (bootstrap) from the training set. The model aggregates the results from trained classifiers (Decision Trees) through averaging to improve the accuracy of prediction and control over-fitting.

3.5.10. Extremely Randomized Trees (Extra Trees)

Extra Trees Classifier is an ensemble supervised machine learning classifier that fits a number of randomized Decision Trees on different subsets of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. As in RF, a random subset of input features is used, but instead of looking for the most discriminative thresholds, thresholds are drawn at random for each feature and the best of these randomly-generated thresholds is chosen for the splitting rule [29]. The applied Extra Tree classifier utilized 100 Decision Trees. Each tree in the ensemble is built using bootstrapping. Two experiments are implemented. The first one, the minimum number of samples used to split an internal node was 3. The second trial, the minimum number of samples used to split an internal node was 2 and the maximum depth of the tree was 3.

3.5.11. Bagging Classifier

The bagging classifier is one of the ensemble methods [30]. It fits several base classifiers each on random subsets of the dataset drawn with replacement (bootstrap samples) and then aggregate their results either by voting or by averaging; to reach a prediction. Such a classifier can be used in order to get a model with a lower variance than a single classifier (such as Decision Tree), by using randomization in its core and then making an ensemble out

of it [31]. The applied Bagging classifier uses 100 base classifiers, where 2 different base classifiers are applied. The applied base classifiers are KNN, and Extra Trees. The subsets that are used to build the classifier are drawn with replacement.

4. Results and Discussion

In this work we tried different machine learning techniques for two ways of balancing the dataset. The first way was under-sampling the dataset. Table 3 shows comparison between machine learning techniques that were used to predict sepsis using training set A, after under-sampling the dataset.

Table 3. Results of running each Machine Learning Technique and evaluated using test set A after under-sampling dataset A

Technique	Accuracy of set A	Accuracy of set B
Naïve Bayes	84%	88%
Logistic Regression	76%	82%
Linear Discriminant Analysis	78%	84%
K-Nearest Neighbor		
• **K=5**	77%	73%
• **K=10**	78%	81%
• **K=20**	80%	83%
Decision Tree Classifier	70%	76%
Support Vector Machine using RBF kernel	83%	86%
Ada Boost Classifier	81%	89%
Gradient Boosting Classifier	81%	88%
Random Forest Classifier	77%	84%
Ensemble bagging		
• **using K-Nearest Neighbor**	77%	80%
• **using Extra Trees Classifier**	80%	85%
• **using 100 (Extra Trees using 100 tree, and minimum sample split = 3)**	85%	88%
Extra Trees Classifier		
• **Using 100 tree, and minimum sample split = 3**	81%	86%
• **Using 100 tree , and depth of the tree= 3**	88%	89%

From table 3, we get that the best machine learning technique used with the under-sampling of dataset A is the extra trees classifier having 100 sub-tree and the max depth is three where we got an accuracy of 88%. Validating using dataset B, that same trained technique got an accuracy of 89%.

The second way was over-sampling the dataset. Table 4 shows comparison between machine learning techniques that were used to predict sepsis using training set A, after over-sampling the dataset.

Table 4. Results of running each Machine Learning Technique and evaluated it after over-sampling dataset A

Technique	Set A Accuracy	Validation using Set B (Accuracy)
Naïve Bayes	63%	88%
Logistic Regression	67%	82%
Linear Discriminant Analysis	67%	83%
K-Nearest Neighbor		
• **K=5**	96%	93%
• **K=7**	95%	91%
• **K=9**	94%	90%
Support Vector Machine Using RBF kernel	74%	80%
Decision Tree Classifier	98%	96%
Ada Boost Classifier	71%	90%
Gradient Boosting Classifier	72%	89%
Random Forest Classifier	99%	98%
Extra Trees Classifier		
• **Using 100 trees, and minimum sample split = 3**	93%	94%
• **Using 100 trees, and depth of the tree= 3**	66%	86%
Bagging Classifier		
• **Using 100 KNN classifiers**	90%	94%
• **Using 100 Extra Trees (each extra tree has 100 trees, and minimum sample split = 3)**	80%	85%

From table 4, we get that the best machine learning technique used with the over-sampling of dataset A is the Random Forest classifier having 100 sub-trees where we got an accuracy of 99% on the testing subset of set A. Validating using dataset B, that same trained technique got an accuracy of 98%.

From tables 3, and 4; the best final model is represented in figure 5. The first step in this model is data normalization using the Z-score method. The second step is handling data unbalancing through over-sampling. The third step is handling missing values using the forward filling method. The fourth step is feature selection where 14 features are selected. The fifth step is applying the Ensemble Random Forest algorithm. The final step is evaluating the resulted model using test set A, and then validating this model using dataset B.

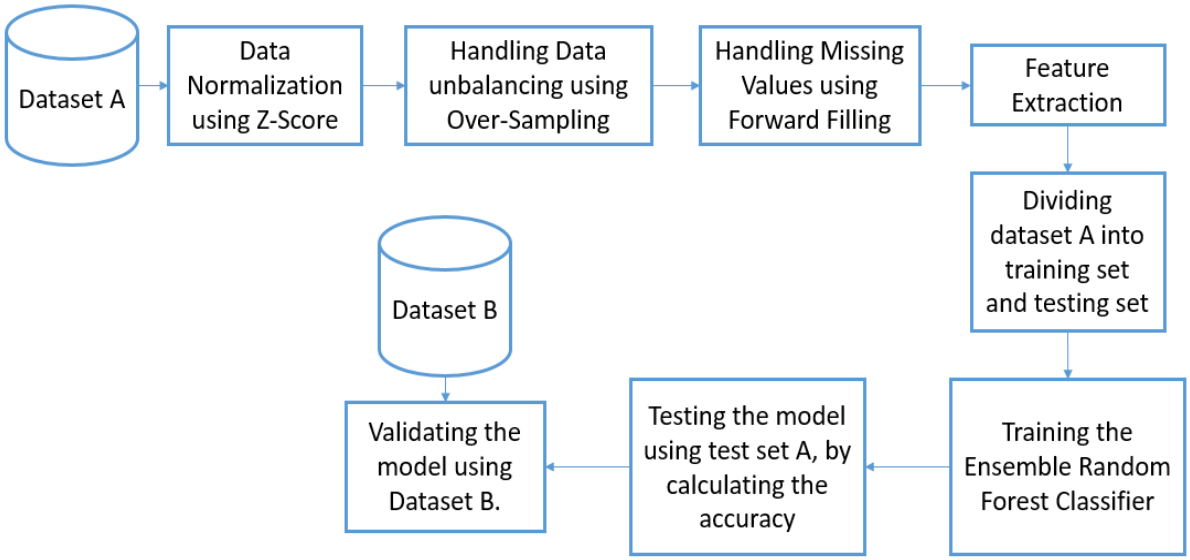

Figure (5): Proposed Final Model for Sepsis Prediction

In order to compare the performance of our proposed model with the teams that participated in the PhysioNet/CinC challenge 2019, we calculated the average results for all ranked teams which is presented in table 5.

Table 5. Challenge teams' average results

	Challenge teams' average Accuracy	Our final Model Accuracy
Set A	84.5%	99%
Set B	88.55%	98%

From tables 1, 3, 4, and 5; it is obvious that our proposed model outperforms the other models in both datasets A and B. The proposed model accuracies that were achieved for sets A and B (99% and 98%) were obtained from the Random Forest classifier considering the dataset imbalance was handled using the over-sampling method.

5. Conclusion and Future Work

Sepsis prediction is a challenging problem and remains so despite many years of research because its manifestation is often unclear until later stages. The model proposed in this paper for real-time sepsis prediction in intensive care unit for the critically ill people demonstrated a prediction performance with 98% as accuracy. Considering the fact that all features came from clinical variables that are regularly collected in ICUs, the proposed model can easily be applied to ICU patients monitoring to improve clinical decision making. In future work, the proposed methodology can be applied to different ICU datasets in order to find a generic model that can be used for real-time monitoring of the ICU patients.

References

[1]. D. C. Angus, W. T. Linde-Zwirble, J. Lidicker, G. Clermont, J. Carcillo and M. R. Pinsky, "Epidemiology of severe sepsis in the United States: Analysis of incidence, outcome, and associated costs of care," Critical Care Medicine, vol. 29, no. 7, pp. 1303-1310, July 2001.

[2]. M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith, . R. S. Hotchkiss, M. M. Levy, J. C. Marshall, G. S. Martin, S. M. Opal, G. D. Rubenfeld, T. van der Poll, J.-L. Vincent and D. C. Angus, "The third international consensus definitions for sepsis and septic shock (Sepsis-3)," Jama, vol. 315, no. 8, pp. 801-810, 2016.

[3]. T. Desautels, J. Calvert, J. Hoffman, M. Jay, Y. Kerem, L. Shieh, D. Shimabukuro, U. Chettipally, M. D. Feldman, C. Barton, D. J. Wales and R. Das, "Prediction of Sepsis in the Intensive Care Unit with Minimal Electronic Health Record Data: A Machine Learning Approach," JMIR medical informatics, vol. 4, no. 3, p. e28, 2016.

[4]. Z. Rayan, M. Alfonse and A.-B. M. Salem, "Intensive Care Unit (ICU) Data Analytics using Machine Learning Techniques," International Journal “Information Theories and Applications”, vol. 26, no. 1, pp. 69-82, 2019.

[5]. M. A. Reyna, C. Josef, S. Seyedi, R. Jeter, S. P. Shashikumar, M. B. Westover, A. Sharma, S. Nemati and G. D. Clifford, "Early Prediction of Sepsis from Clinical Data: the PhysioNet/Computing in Cardiology Challenge 2019," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[6]. R. Firoozabadi and S. Babaeizadeh, "An Ensemble of Bagged Decision Trees for Early Prediction of Sepsis," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[7]. L. Tran, M. Nguyen and C. Shahabi, "Representation Learning for Early Sepsis Prediction," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[8]. P. Biglarbeigi, D. McLaughlin, K. Rjoob, Abdullah, N. McCallan, A. Jasinska-Piadlo, R. Bond, D. Finlay, K. Ng, A. Kennedy and J. McLaughlin, "Early Prediction of Sepsis Considering EarlyWarning Scoring Systems," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[9]. B. Sweely, A. Park, L. Winter, L. Liu and X. Zhao, "Time-Padded Random Forest Ensemble to Capture Changes in Physiology Leading to Sepsis Development," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[10]. A. Deogire, "A Low Dimensional Algorithm for Detection of Sepsis From Electronic Medical Record Data," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[11]. X. Li, G. A. Ng and F. S. Schlindwein, "Convolutional and Recurrent Neural Networks for Early Detection of Sepsis Using Hourly Physiological Data from Patients in Intensive Care Unit," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[12]. R. Pawar, J. Bone, J. M. Ansermino and M. Görges, "An Algorithm for Early Detection of Sepsis Using Traditional Statistical Regression Modeling," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[13]. Q. Yu, X. Huang, W. Li, C. Wang, Y. Chen and Y. Ge, "Using Features Extracted from Vital Time Series for Early Prediction of Sepsis," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[14]. B. Roussel, J. Behar and J. Oster, "A Recurrent Neural Network for the Prediction of Vital Sign Evolution and Sepsis in ICU," in 2019 Computing in Cardiology (CinC), Singapore, 2019.

[15]. H. Zhang, "The Optimality of Naive Bayes," Proceedings of 17th International Florida Artif. Intell. Res. Soc. Conf. FLAIRS, vol. 1, no. 2, pp. 1-6, 2004.

[16]. H.-F. Yu, F.-L. Huang and C.-J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," Machine Learning, vol. 85, no. 1-2, pp. 41-75, 2011.

[17]. D. Jurafsky and J. H. Martin, "Logistic Regression [http://web.stanford.edu/~jurafsky/slp3/5.pdf]," in Speech and Language Processing, Draft of October 2, 2019, p. Chapter 5.

[18]. A. E. Hoerl, R. W. Kannard and K. F. Baldwin, "Ridge regression: some simulations," Communications in Statistics-Theory and Methods, vol. 4, no. 2, pp. 105-123, 1975.

[19]. S. Puntanen, "Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning by Alan Julian Izenman," International Statistical Review, vol. 76, no. 3, pp. 462-463, 2008.

[20]. J. L. Bentley, "Multidimensional binary search trees used for associative searching," Communications of the ACM, vol. 18, no. 9, pp. 509-517, 1975.

[21]. T.-F. Wu, C.-J. Lin and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," Journal of Machine Learning Research, vol. 5, no. Aug., pp. 975-1005, 2004.

[22]. L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, "Classification and regression trees. Belmont, CA: Wadsworth," International Group, vol. 432, pp. 151-166, 1984.

[23]. R. Timofeev, Classification and regression trees (CART) theory and applications, Berlin: Humboldt University, 2004.

[24]. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of computer and system sciences, vol. 55, no. 1, pp. 119-139, 1997.

[25]. T. Hastie, S. Rosset, J. Zhu and H. Zou, "Multi-class adaboost," Statistics and its Interface, vol. 2, no. 3, pp. 349-360, 2009.

[26]. T. Hastie, R. Tibshirani and J. Friedman, The elements of statistical learning: data mining, inference, and prediction, Springer Science & Business Media, 2009.

[27]. J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Annals of statistics, pp. 1189-1232, 2001.

[28]. L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5-32, 2001.

[29]. P. Geurts, D. Ernst and L. Wehenkel, "Extremely randomized trees," Machine learning, vol. 63, no. 1, pp. 3-42, 2006.

[30]. L. Breiman, "Bagging predictors," Machine learning, vol. 24, no. 2, pp. 123-140, 1996.

[31]. Z.-H. Zhou, Ensemble methods: foundations and algorithms, CRC press, 2012.