Intelligent Medical Decision Support System for Predicting Patients at Risk in Intensive Care Units

Completed Research Paper

Araek Tashkandi
Institute of Computer Science
Georg-August-University Goettingen
Goettingen, Germany
araek.tashkandi@cs.uni-goettingen.de
astashkandi@uj.edu.sa

Lena Wiese
Fraunhofer Institute for Toxicology and Experimental Medicine
Hannover, Germany
lena.wiese@item.fraunhofer.de

Abstract

Patients' lives can be rescued by a prediction made by an Intelligent Medical Decision Support System (IMDSS). Such a system can harness the information wealth of patient Electronic Medical Records and leverage up-to-date Machine Learning technology. The accuracy of prediction is one of the most critical characteristics of this intelligent system. Moreover, the technical issues of the medical data as the curse of dimensionality and imbalance are significant challenges. In this paper, we implement the main building block of an IMDSS which is the predictive model. In addition, a comprehensive study of different accuracy factors of this system is given. We tested different approaches and methods for these factors to reach an optimal setting for system development. A big real-world medical dataset is used to test the model for predicting the in-hospital risk of mortality from only the first 24 hours of stays in the Intensive Care Unit.

Keywords

Machine Learning, Decision Support System, Healthcare, Imbalanced data, Risk of Mortality.

Introduction

With advanced health information technology for electronically collecting data through different sources, a vast amount of medical data has been available. Analyzing these data can produce useful knowledge for the patients and the medical staff. For this aim, an Intelligent Medical Decision Support System (IMDSS) is implemented. This system harnesses the information wealth of this vast amount of data to support medical decision making. The intelligence of this system comes from its main building block, which is a Machine Learning (ML) model. The ML model is trained on the medical data to learn how to predict the patient's case accurately. Thus, it is unlike the most clinical decision support systems that require a knowledge base for decision support. In this paper, the intelligent healthcare system is used for predicting patients at risk in Intensive Care Units (ICUs). Sudden death is a serious problem the ICU patients suffer. Implementing an IMDSS for predicting the patients that are at risk of death can minimize the number of sudden deaths in ICUs.

The technical issues of the medical data as the curse of dimensionality, missing values (sparsity), and class imbalance problems are significant challenges when implementing the system. For more details, see (Lee and Yoon 2017; Johnson et al. 2016). Curse of dimensionality is when high dimensional data causes many issues such as data sparsity, which makes the classifier decision boundaries difficult. Feature selection is one approach to handle high dimensional data. As stated by (Li et al. 2010): "In medical data sets, data are

predominately composed of “normal” samples with only a small percentage of “abnormal” ones, leading to the so-called class imbalance problems.”. An imbalanced real-world ICU dataset contains a majority percentage of the survived patients and a minority percentage of the died patients. The issue of imbalanced class distribution causes the classifier to be extremely biased towards the majority class and discounting the minority class. Nevertheless, the minority class is the class of interest. These issues are accuracy factors to this IMDSS. The accuracy of prediction is one of the most critical characteristics of this intelligent system. In this paper, we test some methods for handling these problems of imbalanced data (by data sampling) and high dimensionality (by feature selection).

Different Machine Learning (ML) models can be leveraged to implement this intelligent system. The selection of the predictive model influences the system accuracy. Researches have been using the advances of ML to develop such an IMDSS for predicting the risk of mortality for ICU patients (Ghassemi et al. 2015; Luo et al. 2016). However, less attention has been giving to studying different accuracy factors for this IMDSS system.

Implementing this system requires considerable effort and many steps and decisions. In this study, we aim to give a first level overview of model deployment steps for this IMDSS system. Furthermore, we provide a comprehensive study of different accuracy factors that affect the prediction performance of this system. This was done partially or ambiguously in the previous literature. For implementing this intelligent system, we develop different ML models and compare their performances. We aim to reach the optimal setting for accurate prediction. We compare the prediction performance of seven ML models such as Logistic Regression, Gradient Boosted Decision Tree, and K-Nearest Neighbors. The models are applied on data of the first 24 hours in the ICU stay to predict the in-hospital risk of death for ICU patients. A big real-world medical dataset is used to test the system.

The imbalanced data affect the performance of the ML models. However, our implemented model GBDT had a significantly higher performance than other tested models, even on the imbalanced data without any optimization (area under the curve (AUC)=0.859). Moreover, it outperforms the prediction performance of some of the previous studies on similar versions of our used dataset (Ghassemi et al. 2015; Morid et al. 2017; Lee et al. 2015; Luo et al. 2016).

This paper is organized as follows. Section "Related Work" provides a survey of related approaches. Section "Steps for Developing the Predictive Model of the Health Predictive System" presents the process to develop the model for the predictive system. It includes the extraction and pre-processing of the dataset, implementing the ML models, and tuning the accuracy factors. Next, Section "Results" provides a comparative analysis of different ML models for the task at hand and choose the best-performing candidate. Moreover, it presents the optimal settings of model parameters and other accuracy factors. Section "Discussion" discusses the results we find through the proposed process. Finally, Section "Conclusion" concludes the paper.

Related Work

One approach for predicting the risk of mortality are predictive scoring systems as Simplified Acute Physiology Score (SAPS; Le Gall 1993), and Sequential Organ Failure Assessment score (SOFA; Vincent 1996). However, many studies approve that they outperform their performance by using Machine Learning (ML) models which is another approach for predicting the risk of mortality such as by Lee et al. (2015) and Morid et al. (2017). Machine learning models are utilized to enhance the accuracy of risk prediction.

Several studies used ML models for predicting patients at risk of death, and even some compare the predictive performance between different ML models. However, little attention has been paid to report on the handling of varying accuracy factors simultaneously: model parameters, feature selection techniques, and treatment of imbalanced data. Lee et al. (2015) deploy Logistic Regression, Decision Tree, and death counting of similar patients to predict mortality for ICU patients. They find that using the data of similar patients as a training dataset improves prediction accuracy. No discussion or experiment take place regarding different data pre-processing or feature selection techniques. For instance, they only use normalized data to train and test the models. We compare the usage of both normalized and un-normalized data. Morid et al. (2017) implement a framework for ICU mortality prediction based on similarities amongst ICU patients’ data. Their similarity-based predictive model uses k-nearest neighbor learning. They also utilize feature weight adjustment (by only the wrapper approach (Gradient Descent)).

Stylianou et al. (2015) compare Logistic Regression against different ML models (artificial neural network, support vector machine, random forests, and naïve Bayes) for predicting mortality risk from a burn injury. They find that all the predictive models have comparable performance. The simple Logistic Regression model performs well in comparison to the other complex models. Allyn et al. (2017) is another work using ML in medical prediction, and it is outperforming the score model in mortality risk prediction. They compare different ML model's performance (Logistic Regression, Gradient Boosting Machine, Random Forests, Support Vector Machine, Naïve Bayes) with the EuroSCORE-II to predict mortality after cardiac surgery. They approve that the ML models provide significantly higher accuracy than EuroSCORE-II. Regarding feature selection, they only use the filter approach (Chi-Squared). Hoogendoorn et al. (2016) also exploit the ML models for building a mortality prediction model. They compare the performance of LR and KNN. For feature selection, they use the Pearson correlation coefficient.

Ghassemi et al. (2015) use Lasso logistic regression and L2 linear kernel Support Vector Machine. They also consider data sampling to handle imbalanced class distribution, the best AUC=0.812. Luo et al. (2016) implement a Logistic Regression model for mortality risk prediction and use non-negative matrix factorization for feature extraction. The authors aim to improve model interpretability and accuracy; their model had an AUC=0.848.

Our contribution to this field is as follows. In this paper, we give a first level overview of model deployment steps for this IMDSS system. This study highlights the significant effect of different accuracy factors that have been ignored or partially searched by the previous studies. The former works implement their approach of accurate prediction without comparing different methods, such as for features selection or data pre-processing. We provide a detailed discussion on selecting the model parameters, whereas, in the previous studies, they made an ambiguous selection.

Furthermore, we compare seven ML models and the two approaches (filter and wrapper) for feature selection. We test the different combinations between the best-selected ML models and these techniques of feature selection. Moreover, we shed light on the common problem of imbalanced data. Therefore, we provide a comprehensive study and comparisons of different approaches and factors that lead to accurate predictions. Consequently, we aim to deliver the optimal setting for developing a predictive model for IMDSS predicting patients at risk. As a result, our selected and implemented ML model GBDT provides substantially higher performance than models evaluated on similar versions of the publicly available dataset MIMIC reported in the literature (Ghassemi et al. 2015; Morid et al. 2017; Lee et al. 2015; Luo et al. 2016). In our analysis, GBDT achieved an AUC of 0.859 (with the imbalanced data). Lee et al. (2015) have the same selected features as us (except we excluded two features), and the best produced AUC is 0.830. Morid et al. (2017) have 0.66 F-measure, while GBDT made a higher F-measure 0.78. Furthermore, with another dataset Allyn et al. (2017) reached an AUC of 0.795.

Steps for Developing the Predictive Model of the Health Predictive System

To develop an accurate medical DSS, significant and critical works have to be done in developing the predictive model. Developing the predictive model for such an intelligent system has many steps starting from patient data extraction and ending when an optimal model with high accuracy is defined. Figure 1 gives an overview of this process, which inspired by the typical ML pipeline or workflow.

The process in Figure 1 is as follows: The patient data are extracted from the Electronic Medical Record (EMR) database in the form of vectors and features. Data of each patient are collected in a vector. The features consist for example of vital signs or laboratory measurements. These medical data are often messy and contain many problems that have to be solved to make the data ready to be used by the models (Malley et al. 2016). Messiness in this sense can occur in the form of noisy (e.g. outlier values), incomplete (e.g. missing values), and inconsistent (e.g. errors made at data entry) data. An important step is hence to apply data pre-processing. Then, different ML models are implemented and tested to find the best one. Different factors affect the accuracy of this system: for instance, the model parameters, the selected features, and the data format. These factors are tuned, and simultaneously the model training and testing processes are repeated. Once an optimal setting of a different combination of the model and parameters values of the different factors is found the process is accomplished. This ideal model and parameters are used to develop the DSS. In the following subsections, we will go through the main steps with the different approaches.

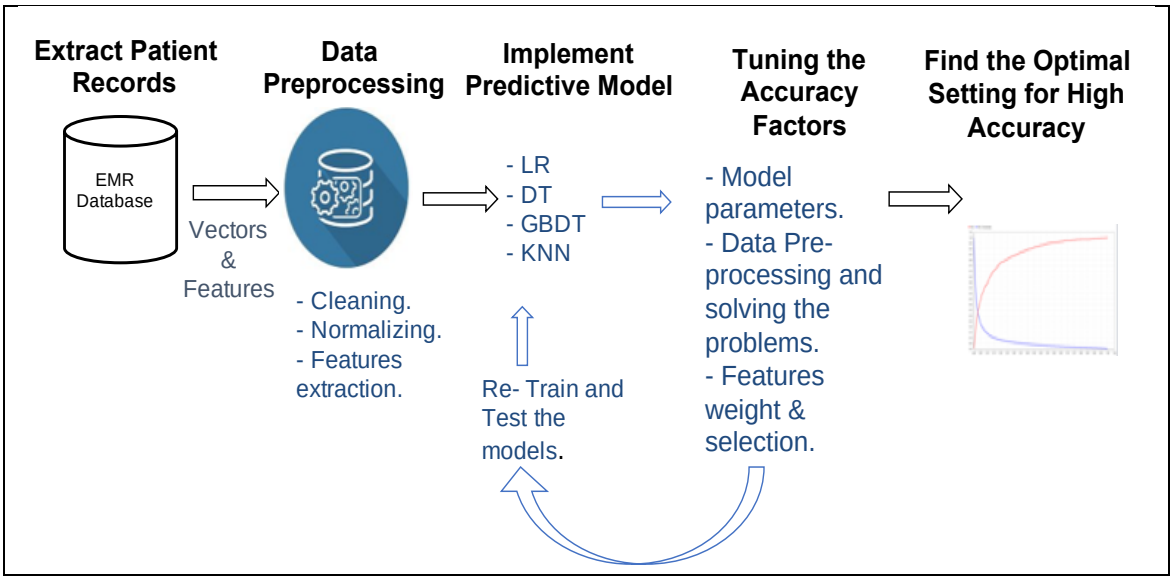

Figure 1. Developing the Predictive Model of the Health Decision Support System

Data Extraction

For health prediction purposes various medical data has to be extracted and analyzed. The selection of the medical measurements is based on the intention of the prediction. For instance, for diagnoses prediction, the feature selection depends on the disease we are looking to diagnose. In this paper, we use the real-world critical care database Medical Information Mart for Intensive Care (MIMIC) (Johnson 2016). The data is collected from patients admitted to critical care units at the Beth Israel Deaconess Medical Center in Boston, Massachusetts in June 2001 to October 2012. It is a publicly available, widely used, and de-identified dataset. We use the latest version of MIMIC which is MIMIC-III. MIMIC-III comprises over 61,000 hospital admissions to critical care units of 53,423 adult admissions and 7870 neonate admissions with thousands of medical data.

MIMIC-III was collected from different sources: archives from critical care information systems, hospital electronic health record databases, and Social Security Administration Death Master File. It includes the clinical data of critical care. These data include the time-stamped hourly collected physiological measurements as heart rate and other notes and medication data. Moreover, it includes demographic data and in-hospital mortality, laboratory results, and discharge report.

In our study, we are interested in predicting the risk of mortality for the adult patient (aged 15 years or above). Thus, only the data of the adult patient admissions to the different critical care units are extracted. The data of the neonate admissions are not included. The medical measurements (the predictor variables) selection is inspired by Lee et al. (2015). There are 74 predictive variables from the first 24 hours in the ICU stay (see Table 1). Furthermore, age, gender, and ICD-9 code were also extracted.

Predictor variables	Feature extracted	Time window
Vital signs (heart rate, mean blood pressure, systolic blood pressure, Spo2, body temperature, and spontaneous respiratory rate)	Min and Max	From each non-overlapping 6-hour period during the first 24 hours
Lab variables (blood urea nitrogen, hematocrit, white blood cell count, serum glucose, serum HCO3, serum potassium, serum sodium, and serum creatinine.)	Min and Max	From the first 24 hours

Categorical variables (use of mechanical ventilation, receipt of vasopressor therapy)	Binary	From the first 24 hours
Glasgow Coma Scale	Min	From each non-overlapping 6-hour period during the first 24 hours
Urinary output	Sum	From each non-overlapping 6-hour period during the first 24 hours

Table 1. Feature Extracted from the Predictor Variables

The value that we want the model to predict is if the patient has a risk of in-hospital mortality. Thus, the value of the in-hospital mortality flag is also extracted from MIMIC for test purpose. As a result, these extracted data can help us build the model to predict the risk of in-hospital mortality after the first 24 hours of ICU stay.

Data Pre-processing

Medical measurements are collected by numerous sources. This produces big and messy data collections. From the previous dataset description and extraction, we can imagine how diverse the medical data is. It is collected from different sources and systems with different formats and problems such as missing values. A real-world dataset usually is not ready for directly applying Machine Learning models. It has to be pre-processed and cleaned. Therefore, the pre-processing step is one of the main steps in developing the predictive model. Consider the dataset used in training and testing the model as a critical accuracy factor; data pre-processing should be done with caution.

The patient data of n patients with m features are represented in an m-dimensional vector space: The patient vectors N = P1, P2, . . . , Pn each consist of features x1, x2, . . . , xm (see Table 2). Thus, m is the size of the feature set, while n is the size of the patient set.

PatientID	Feature_1	Feature_2	. . . . .	Feature_m
1	value_1	value_2	. . . . .	value_m
.	.	.	.	.
.	.	.	.	.
n	value_1	value_2	. . . . .	value_m

Table 2. The Patient Data Layout

The data should be clean. All the patient records with null values in any measures are excluded. The final dataset that met our criteria and without missing values is 32,548 patients. The 74 extracted medical measurements have multiple ranges and units. Some ML models require normalizing or scaling the data as for example KNN. Moreover, the predictor variables with different ranges will have different weight. Therefore, we normalize the data into a scale of smaller range. All the predictors are normalized into the range [0, 1] by computing this formula of min-max normalization to the features (Han et al. 2011):

$$x' = \frac{x - \min_x}{\max_x - \min_x} . (\max_{new} - \min_{new}) + \min_{new}$$

In this equation, the feature or the predictor variable x has a range with the minimum value ($\min_x$) and the maximum value ($\max_x$) is normalized to a new range with the minimum value ($\min_{new}$) and the maximum value ($\max_{new}$) to produce the new value x'.

Implement Machine Learning models for Prediction

Machine learning models are utilized to enhance the accuracy of risk prediction. Different models can be used in the predictive analysis for medical data. Predicting risk of mortality can be seen as a classification task. It is a binary classifier for two classes either a patient has a death risk (the positive class with the label "1"), or a patient has no risk of death (the negative class with the label "0"). Several supervised learning algorithms can be employed for this task. The developed medical DSS will need one predicative model as a

main building block. There is no such a superior model that works best for all the purposes. The models work differently with different situations. Therefore, the selection between the models happens after comparing their performances in the aimed prediction. Then, select the one with the highest accuracy. In this paper, we select some of the commonly used models (as mentioned in the Related Work section) and compare their performance in predicting the risk of mortality in our dataset. The models we compare are Logistic Regression (LR), Decision Tree (DT), K-nearest neighbor (K-NN), Naïve Bayes (NB), Gradient Boosting Decision Tree (GBDT), Support Vector Machine (SVM), and Random Forests. We test the models on the complete dataset of 32,548 patients by 10-fold cross validation. We set the models' parameters to the default values. We compare their performance by the AUC (see Figure 2) to select an initial group of models for further tests. The value of AUC=0.5 indicates a random guesses predictor, while the higher value of the AUC indicates better model discrimination.

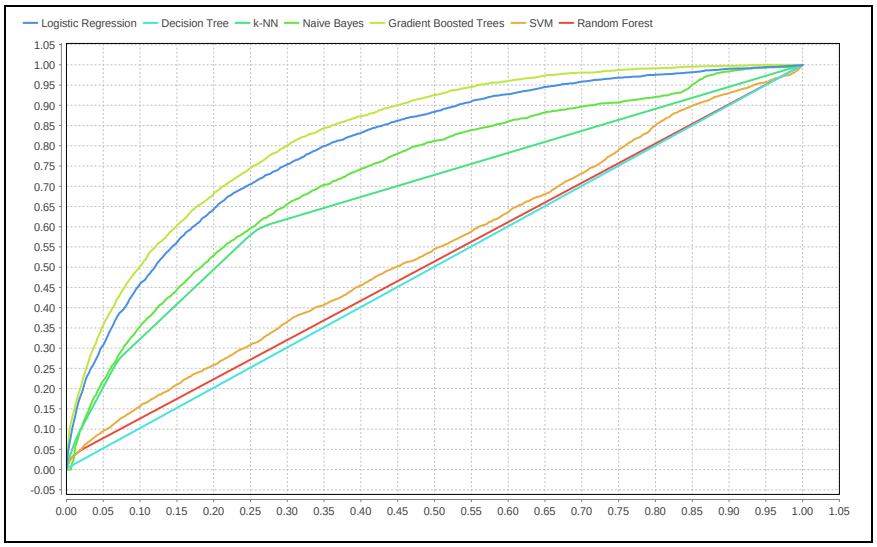

Figure 2. Compare the AUC Performance of different ML Models

The model performance can be measured by different accuracy metrics such as Accuracy, Recall, Precision, F-Measure, AUC. We will explain some of these metrics and what they mean in our study content. In our classifier we have two classes to predict: the positive class (the class of interest, i.e. the patient with risk of mortality) and the negative class (the survived patient). First, we need to know the output of our classifier which are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). A TP is a patient who is truly predicted to be in risk, FP is a patient who is incorrectly predicted to be in risk, TN is a patient who is correctly predicted as survived, and FN a patient that is incorrectly predicted as survived.

- Accuracy: is the ratio of the total true predictions (for both the patients with mortality risk (TP) and survived patients (TN)) to the all predictions made by the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Recall: is also called the True Positive Rate (TPR) and the Sensitivity. It is a fraction of the patients that are correctly predicted with risk over all the patients who have the risk. It measures the efficiency of the model of predicting the entire group of patients with risk of mortality. This formula calculates it:

$$Recall = \frac{TP}{TP + FN}$$

- Precision: for our model, it is the fraction of the patients who are truly predicted with mortality risk over all the patients that are predicted with mortality risk. The higher the precision is, the lower the number of incorrectly predicted patient with risk. This formula calculates it:

$$Precision = \frac{TP}{TP + FP}$$

- F-Measure: is the f1 score that combines the Precision and the Recall to give an average of them. As a result, this metric measures the efficiency of the predictive model for predicting both the patient with mortality risk and without risk.

$$F1\ score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

- AUC: Area Under the ROC Curve (AUC). First, the ROC graph shows the model performance at all classification thresholds by plotting the relation between TPR and FPR. The AUC aggregates the performance among all the classification thresholds. The FPR is the ratio FPR= FP/FP+TN.

A tradeoff occurs between accuracy metrics such as between Recall and Precision. Thus, we should consider which of the evaluation metric is the most worth for our predictive model. We mostly care to correctly predict the patients with mortality risk (i.e., TP). Moreover, a low number of FNs is more crucial than a low number of FPs. A high Recall has a high priority for our system.

We select the models that have an excellent performance in Figure 2 for further analysis. The models that had the highest AUC values are GBDT, LR, Naïve Bayes, and KNN. However, even though Naïve Bayes gives a good AUC of 0.729 (from Figure 2), its accuracy was low 29.77%. Thus, we do not include it for further tests. We include DT instead. In the following section of "Results," we will compare the models' performance in detail with other metrics. As a result, we will find the best model for our intended prediction.

Tuning the Accuracy Factors

Brink et al. (2017) mentioned three techniques for achieving better model accuracy: tuning the model parameters, selecting a subset of features, and pre-processing the data. We applied these three steps as follows.

Tuning the Model Parameters

Each of the used Machine Learning models is configured by specific parameters. These tuning parameters control how the algorithm uses training data to build a model. There are no standard best values of these model parameters. In general, the optimal value of these parameters is entirely dependent on the type, and the structure of the used dataset and on the problem that needs to be solved. The parameter values impact the predictive performance of the model. Thus, we should do cautiously selecting.

We study the critical parameters of each of the selected ML models. We test the effect of those parameters on the model performance. Furthermore, we implement Grid Search to find the optimal values of the parameters for our used case. Here are the crucial parameters we test for the models:

For LR the main parameter is the regularization parameter. Assigning a small value to the regularization parameter develops a simple model. LR can be prone to over-fitting with high dimensional feature spaces. Regularization gives a more reasonable decision boundary (for separating the positive and negative samples) that prevent over-fitting. It is penalizing the parameters from being too large.

DT's parameters that affect the performance are splitting criterion, max depth of the tree, and minimum samples needed to do a split. Splitting criterion selects the criterion on which attributes will be selected for splitting. The split value is optimized with regards to the chosen criterion: Gain Ratio, Information Gain, or Gini Index.

GBDT parameters: the more complex the algorithm is, the more numerous the tuning parameters are. In our case, the GBDT model is the one that has the largest number of tuning parameters. The critical parameters of GBDT are the number of trees, max depth of the tree, learning rate, splitting criterion, and minimum samples needed to make a split. Hence the GBDT is built out of ensemble DTs; the parameters are mainly similar. However, it has two other parameters which are the main parameters – number of trees and learning rate. The learning rate is the degree of mistakes correction that each tree is allowed to do of the previous trees. These two main parameters have an inverse relationship — the lower the learning rate, the larger number of learning trees that are needed to build a model. The maximum depth is usually set to very low to reduce the complexity of each tree, often not deeper than five splits.

KNN parameters: The critical key choices that affect the KNN performance are the value of k nearest neighbor, the approach to combine the class labels and the choice of distance metric. The approach to combine the class labels is the way to decide the predicted class label.

Selecting a Subset of Features

In the big data age, it is common that the used dataset for prediction is high dimensional. A large number of features might include a noise that causes difficult knowledge discovery and makes it hard to find such relevant indicators from the data. A high dimensional dataset not only slows down the training process but also makes finding the optimal solution harder. Thus, dimensionality reduction has to come into play. It is not obvious to know the effect of the features on the model. Therefore, we should carefully search for the features that build the most general and accurate model.

Dimensionality Reduction reduces the noise and removes the unnecessary details in the data which produce higher performance. However, it is not the general case, since it commonly only speeds up the training (Géron 2017). Therefore, by reducing the dimensionality, we should make sure not to lose much information. We test two approaches for feature selection: the wrapper approach and the filter approach; see Guyon et al. (2003) and Kaushik (2016). The wrapper method wraps a machine learning model inside it as a black box to evaluate a subset of the features. The ML model is trained on the feature subset, and it scores them according to their predictor power. On the other hand, the filter method selects the features independently of the learning machine. It selects the features based on a statistical score about their correlation to the predicted value. From the filter approach we test the Chi-squared. From the wrapper approach we test these two methods: Forward Selection, and Backward Elimination; see Kohavi et al. (1997) and Panthong et al. (2015):

- The backward elimination is searching for the features by beginning with the full set of the features and removing one feature at a time. With each deletion of the feature, the model performance is evaluated. The feature that gives the lowest performance decrease it will be deleted. This process continues until a decrease in model performance occurs.
- The forward selection begins with an empty set of features and then adds one feature at a time. With each addition of a feature, the performance is estimated by cross validation. Only the feature that by its inclusion gives a high performance improvement is added to the selected list. It adds the feature that gives the highest increase of the model performance. The iteration is stopped when no increase in the model performance occurs.

Pre-processing the Data

The problems that a dataset contains affect the performance of the ML models. Thus, we explore our dataset and visualize it and do some statistical overviews to find the data issues. The main issue of our dataset is imbalanced classes. Our used dataset of MIMIC-III includes 28,887 alive patients and only 3,661 dead patients. The ratio of Class-1 (survived patient) instances to Class-2 (died patient) instances is 89:11. We test our models with a balanced dataset to test their performance without any data issue effects.

Different feature value range will have different weight. Therefore, normalizing features to a specific range make sure all features are considered equally. We normalized all of our predictive variables to the range from 0 and 1. However, to find the best for optimizing the predictive performance of our system, we also test the un-normalized data.

Results

The complete 26 tables of the MIMIC-III dataset were installed and were queried by a Structured Query Language (SQL) statement to extract the predictor variables. All the data pre-processing and model's implementation, training, and testing were done by RapidMiner studio version 9.2 Educational edition. RapidMiner is called a leader by Gartner in Gartner's 2019 Magic Quadrant for Data Science and Machine Learning Platforms for Sixth Consecutive Year (Idoine et al. 2019). The computation takes place in Windows 10, Intel i5-7300U, CPU 2.70 GHz, RAM 32 GB, x64-based processor.

The Optimal Model Parameters

We discover the critical parameters of each model. Then, we test different values of the models' parameters. For all of the models, we use Grid Search to find the optimal parameters values that give high accuracy. For instance, the KNN crucial parameters are the value of k (the number of the nearest neighbors), the distance metric, and the approach to combine the class labels. We test values for k from 1 to 50. We find that k=21 is the optimal value with high accuracy. Moreover, we test the distance metric parameter with 10-fold cross validation. We compare various distance metrics but all with the same k. In this example we test k=21. From Figure 3 we can see the different AUC performance the KNN model has with different distance metrics.

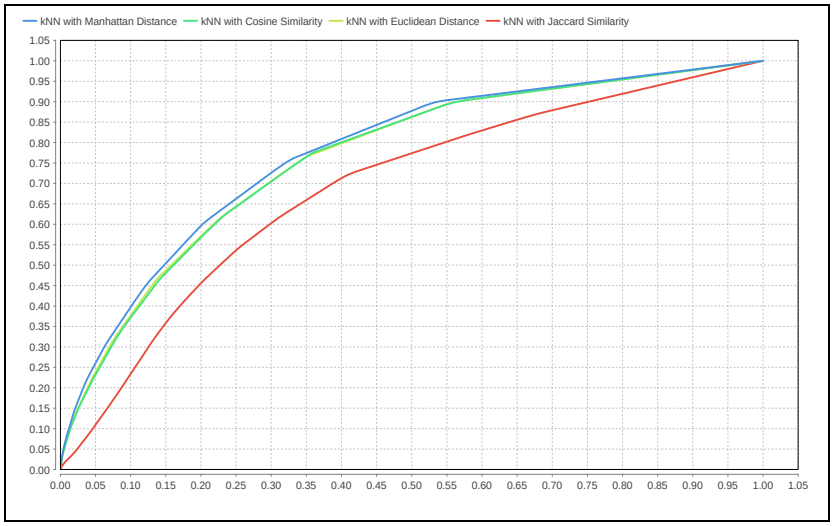

Figure 3. KNN Model with different Distance Metrics

The last parameter we test for KNN is the approach to combine the class labels. We compare two different approaches: the majority vote and the weighted vote — the accuracy of the two approaches was the same with 86.96%. However, looking more in detail into the performance result we find that differences were in AUC and the Precision and Recall of predicting the positive class (i.e., predicting mortality). The majority vote gives AUC of 0.770 and 72.13% precision and 1.03% recall. The weighted vote improves the performance where it gives 0.772 AUC and 1.28% recall but lowers the precision 68.75%. This is caused by the mentioned problem we have in our dataset (imbalanced class distribution). The class label that commonly occurs will affect the predicted value. In our dataset, the survived patient class is the common one. Thus, using the majority vote with larger k values is not a good choice in our case.

For LR the regularization parameter is called lambda which controls the amount of the applied regularization. We search for the best fit lambda value. Lambda of zero gives the highest AUC 0.801 +/- 0.011.

For the DT, we test the three splitting criterion approaches: gain ratio, information gain, and the Gini index. The max depth of the tree was fixed to be 20. We find that the gain ratio gives the highest accuracy of 88.76% among the other splitting criteria. Furthermore, we test the max depth of the tree from a range of 1 until 100 with ten steps linear scale. The splitting criterion was set to gain ratio. Max depth of 41 has the highest accuracy of 88.80%. Finally, we test the minimum samples needed to do a split. The range is from 1 to 100 with ten steps in a linear scale. The splitting criterion is set to gain ratio and the max depth to 41 which are the optimal values we found in the previous tests. The minimum samples size of 11 produces the highest accuracy of 88.84%.

For GBDT, we test both the number of learning trees and learning rate. All the combinations between the learning rate from 0.1 to 1.0 and the number of trees from 1 to 200 are tested. There are 11 variations for the learning rate and 11 variations for the learning trees. Each combination is tested by 10-fold cross-validation. The result is that 11*11*10=1210 models are trained and evaluated. The other parameters are set to be the same for all the 10-fold cross validation iterations. The maximum depth of the tree was 5. The optimal result is reached with the learning rate of 0.1 and with the number of trees equals to 200. This gives accuracy 88.94% and AUC 0.860 +/- 0.007. Figure 4 shows the accuracy of the different number of trees with the range of learning rate. However, the accuracy of the learning rate of 0.19 where the number of trees

equals 160 is almost similar to the best one. In general, low learning rates cause fewer corrections for each tree added to the model. Therefore, the smaller the learning rates is, the more trees are required to be added to the model. Moreover, the small max depth of the tree can affect this need for more learning trees.

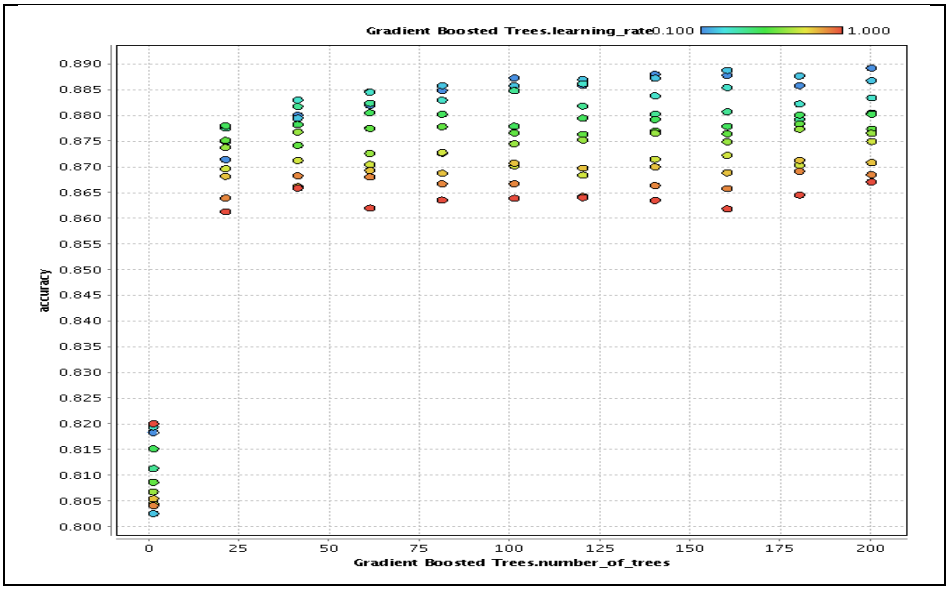

Figure 4. Testing Different Learning Rate with Different Number of Trees of the GBDT Model

In the following tests of performance optimization, we will use the optimal parameters values we found by Grid Search.

The Best ML Model

We test the initial model group further to select the best model for our system. We set the models' parameters values to the optimal values we found. Then, we test the models by 10-fold cross-validation. We compare the accuracy of the four models (LR, DT, GBDT, and KNN) on the normalized data. All the models have high accuracy and almost the same value. This suspicious result leads us to look at other performance metrics. Table 3 gives a comparison of the four models.

	Accuracy	AUC	Precision	Recall
LR	89.22%	0.801	58.63%	13.41%
DT	88.82%	0.500	70.00%	0.57%
GBDT	88.78%	0.859	50.08%	41.82%
KNN	88.90%	0.768	63.73%	2.43%

Table 3. Compare Models Performance

From Table 3, we find that the models were successful in predicting survival cases rather than the death cases. The high predictive accuracy was a sign for overall prediction of the majority class which is the survived case. Thus, we should consider other metrics in evaluating our model besides the accuracy. Even though the models have high accuracy, they have a very low Recall (that measures how often a positive class instance is truly predicted as a positive one). It is incredibly distinct in DT and KNN. Out of the box without any further optimization and with the imbalanced problem, the GBDT and LR give the highest performance (AUC and Recall). Therefore, we select them for the following tests.

Pre-processing the Data and Selecting a Subset of Features

One of the data pre-processing decisions to select is normalizing the data to a specific range or not. We compare the two approaches (The un-normalized dataset as the original one and the normalized dataset except the ICD-9 code) with the two models LR and GBDT. We find that the data with normalized features

(except the ICD-9 codes) gives a higher Recall than the data of unnormalized features. For instance, by GBDT the normalized data except the ICD-9 codes gives 45.77% Recall, but all normalized features (as shown in Table 4) give 41.82%. Thus, we use the data with all normalized features except the ICD-9. The results are shown in Table 4. The normalized data slightly improves the Recall and the F-measure. However, there are no significant differences. The reason might be that these models do not require feature scaling. Thus, scaling or not scaling makes no such difference. However, with the model that requires scaling such as KNN, the normalizing would affect the accuracy.

	Accuracy	Precision	Recall	F-Measure
LR+Udata	89.21%	58.53%	13.25%	21.59%
LR+Ndata	89.22%	58.52%	13.34%	21.70%
GBDT+Udata	88.53%	48.94%	43.87%	46.17%
GBDT+Ndata	88.41%	48.46%	45.77%	46.99%

Table 4. Compare Models Performance with Un-normalized Dataset and with Normalized Dataset

We apply feature selection approaches (wrapper and filter) on both LR and GBDT. First, regarding the wrapper approach we test the Forward Selection (FS) and the Backward Elimination (BE) on LR and GBDT. Then, from the filter approach, we test feature selection weight by Chi-Squared (Chi). Chi-Squared weights all the 74 features and then only the features with the top 20 weight are selected (see Table 5). All the tests were on the normalized data (except the ICD-9 code). We compare the three approaches on each model.

Feature	Weight
Blood urea nitrogen_min	1206.72
Blood urea nitrogen_max	1109.48
Serum HCO3_min	979.23
Serum HCO3_max	765.85
Spontaneous respiratory rate_18h_min	650.84
Spontaneous respiratory rate_12h_min	641.34
Sodium_max	609.99
ICD-9 code	582.77
Spontaneous respiratory rate_6h_min	571.23
Spontaneous respiratory rate_24h_min	561.54
Systolic blood pressure_24h_min	547.43
Heart rate_24h_max	506.01
Age	465.82
Heart rate_24h_min	430.77
Spo2_24h_min	418.36
Glasgow Coma Scale_min	392.76
Systolic blood pressure_18h_min	342.89
Heart rate_18h_max	336.49
Systolic blood pressure_6h_min	327.11
Use of mechanical ventilation	320.33

Table 5. Top-20 Features Weight by Chi-Squared

The result of comparing the three methods of feature selection on LR in Figure 5. LR with FS selects 17 attributes: total-urinary-output_18h, total-urinary-output_24h, mean-blood-pressure_12h_max, rr_24h_max, mean-blood-pressure_12h_min, mean-blood-pressure_24h_min, spontaneous-respiratory-rate_12h_min, systolic-blood-pressure_24h_min, spo2_12h_min, spo2_24h_min, body-temperature_6h_min, glucose_max, HCO3_min, blood-urea-nitrogen_min, age, use-of-mechanical-ventilation, and Glasgow-Coma-Scale_min. It has AUC of 0.778 +/- 0.013.

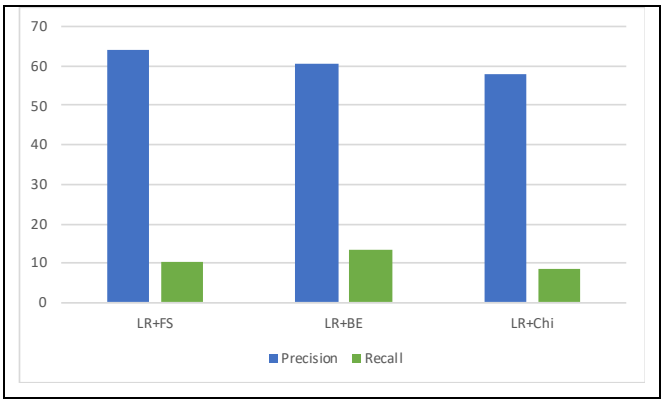

Figure 5. LR and Feature Selection Methods

The result of the three feature selection methods with GBDT is in Figure 6. The GBDT with FS selects 8 attributes: total-urinary-output_18h, total-urinary-output_24h, spontaneous-respiratory-rate_18h_max, hr_24h_min, mean-blood-pressure_18h_min, systolic-blood-pressure_6h_min, blood-urea-nitrogen_min, and serum-creatinine_min. The AUC is 0.752 +/- 0.015. LR takes much less time than GBDT.

Next, we tested the Backward elimination approach for LR and GBDT. The LR eliminates 6 attributes: total-urinary-output_24h, heart-rate_18h_max, spo2_12h_max, body-temperature_24h_min, sodium_max, and ICD-9_code. It produces AUC of 0.798 +/- 0.010. GBDT eliminates only one attribute: heart-rate_6h_max with AUC of 0.859 +/- 0.006. This test takes 5 hours and 33 minutes for GBDT and 1 hour and 37 minutes for LR.

By the Chi-squared we select the top 20 features; then we use the dataset with only these 20 features to test the models. GBDT gives AUC of 0.845 +/- 0.007 and LR gives AUC of 0.768 +/- 0.011. Chi-squared with LR gives the worst Recall and Precision. Chi-squared with GBDT gives Recall almost similar to the one of BE and Precision lower than BE, while in comparison to FS the chi-squared with GBDT is much better.

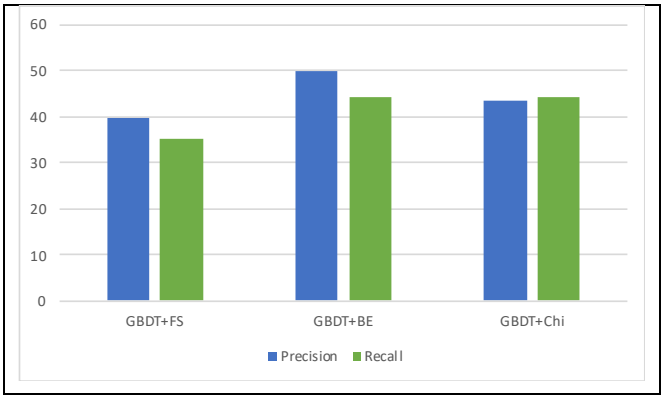

Figure 6. GBDT and Feature Selection Methods

At the end of implementing feature selection methods, we are still not able to achieve high predictive performance. The highest Recall we got is 44.43 of GBDT with Chi-squared. In comparison to the Recall GBDT has without feature selection 41.82 it is only improved by 5.83%. Moreover, for LR the improvement

in Recall is only by 4.45%. This low-performance improvement in predicting the critical cases are due to the imbalanced data we have.

Therefore, we test the models' accuracy without the effect of the imbalanced problem. Thus, we sampled the dataset to a balanced one. This balanced dataset has a 1:1 ratio of the two classes. We select all the instances of the minority class and we randomly select from the majority class the same instance number of the minority class. This sampling is considered to be a random under-sampling of the majority class. We test the effect of balancing the classes on the GBDT. Thus, we compare the GBDT performance with the original data (imbalanced data) and with the balanced data. In both cases, the data contains all normalized features. The result is summarized in Figure 7. We find that the balanced dataset significantly improves Precision, Recall, and F-Measure. We ignore the accuracy because we know that the accuracy of the imbalanced dataset is skewed (because of the problem of bias accuracy to the majority class which has higher occurrence).

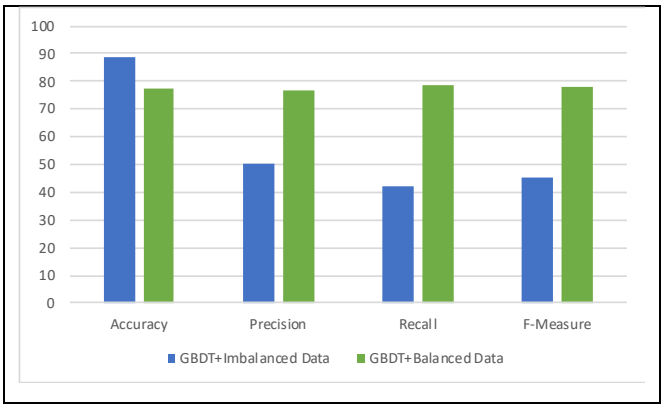

Figure 7. GBDT with The Balanced Dataset and with The Imbalanced Dataset

We test the effect of balancing the classes on the LR. Thus, we compare the LR performance with the original data (imbalanced data) and with the balanced data. The result in Figure 8. The Precision, Recall, and F-Measure is improved and again the accuracy of the imbalanced dataset is a skewed one. The LR performance is much better with the balanced dataset. However, the GBDT outperforms the LR. For instance, with the balanced dataset, the GBDT's Recall is higher than the LR's Recall by 10%, the Precision higher by 5% and the F-Measure higher by 7.46%.

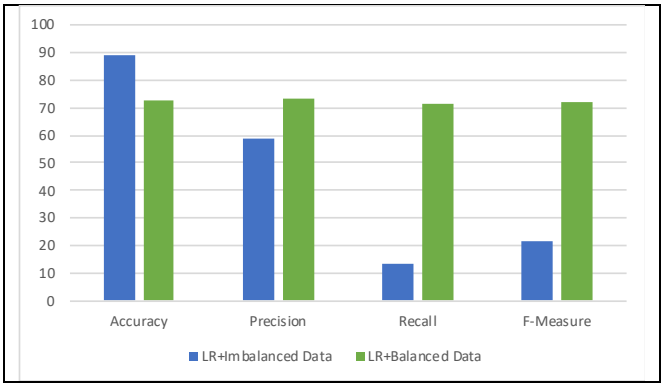

Figure 8. LR with The Balanced Dataset and with The Imbalanced Dataset

Discussion

Through the process of finding the optimal settings to develop the model of predicting patients at risk, we got to interesting findings. GBDT and LR outperform the other models we compare (DT, KNN, Naïve Bayes,

SVM, and Random Forests). Tuning different accuracy factors affect model prediction accuracy. However, the imbalanced class distribution problem has a significant impact on model performance.

This problem of imbalanced class distribution causes the classifier to be extremely biased towards the majority class (i.e., the survived patient). As a result, the models' high accuracy was obtained by predicting all instances as a majority class. Thus, it is the model accuracy in predicting most of the dominant class instances and discounting the accuracy in predicting the minority class ones. Nevertheless, the minority class (i.e., the passed-out patient) is a positive class, which is the class of interest (i.e., we focus on predicting this class). This situation where the higher accuracy metric is not an indicator of an excellent classifier performance is called Accuracy Paradox (Valverde-Albacete et al. 2014). It is paradoxical when accuracy is not a useful metric for the predictive model. With this problem, we should consider other metrics in evaluating our model besides the accuracy. Furthermore, imbalanced data affect accuracy factors, such as the selection of the optimal model parameters. For example, In the KNN model, the weighted vote gives higher performance than the majority vote because of the imbalanced problem. Table 6 summarizes the weaknesses and strengths of the different approaches we test with GBDT and LR.

	LR	GBDT
Un-normalized vs. Normalized data	Normalized data improves the Recall and the F-measure.	Normalized data improves the Recall and the F-measure.
Forward Selection	AUC=0.778 + less time.	AUC=0.752 - long time.
Backward Elimination	AUC=0.798 + less time. -lower accuracy.	AUC=0.859 - long time. +higher accuracy.
Chi-Squared (Top 20)	AUC=0.768	AUC=0.845
Balanced data	Improves all the accuracy measures comparing to imbalanced data.	Improves all the accuracy measures comparing to imbalanced data. +outperforms LR.

Table 6. Different Approaches with Strengths (+) and Weaknesses (-)

From the previous tests of feature selection methods on both models (GBDT and LR), we find these results: in general, GBDT has higher Recall and Precision in Backward Elimination in comparison to the Forward Selection. However, Forward Selection uses only seven attributes, while Backward elimination uses the original large number of features minus one attribute. Similarly, the LR has higher Recall and Precision in Backward Elimination than in Forward Selection. Therefore, the Backward Elimination gives high Recall and Precision and in reasonable computation time but needs a higher dimensional feature space while Forward Selection uses really few features but needs high computation time and gives good Recall and Precision. Here we see a tradeoff between the number of features and the prediction accuracy. Furthermore, we find that the models work differently with different features (Das 2001). It is not only the used features, or the data format or feature selection method that affects the prediction performance but also the model itself. The GBDT model achieves the best prediction performance among all models.

Even though feature selection can improve the prediction accuracy, the improvement is little because of the imbalanced data. We find that a balanced dataset has a remarkable accuracy improvement. Consequently, implementing feature selection on a balanced dataset will have higher accuracy than on an imbalanced dataset. Moreover, the balanced dataset will affect the selection of the features.

Conclusion

An Intelligent Medical Decision Support System (IMDSS) may save patients' lives by predicting the risk of death. Implementing this system requires much effort accompanied by cautious selection and configuration of technologies. The main building block of this system is the predictive model, which is a Machine Learning (ML) model. This paper gives an overview of the main steps of model deployment and data pre-processing that are required to build an IMDSS for predicting the risk of death. Prediction accuracy is a crucial

requirement for this system that determines the system's usefulness. Thus, we studied different accuracy factors that affect the system's prediction accuracy. We tested different ML models, feature selection methods, and pre-processing data approaches. We aim to reach the optimal setting for accurate prediction.

We find that the imbalanced class distribution problem has a significant impact on the performance of the ML models. Out of the seven ML models we tested, GBDT has an outstanding predictive performance even on the imbalanced data. Moreover, it resulted in higher AUC and F-measure than those reported by the related work. We however note that comparability of results with related work is limited, because they used an older (and smaller) version of the dataset. So far, we tested our model only on one dataset. Thus, using other datasets for verifying our model performance should be considered. We got notable accuracy improvement results from applying a balancing approach to solve the problem of imbalanced data. Therefore, testing different methods for solving the problem of imbalanced classes for this system is a topic of future work. In this paper, we implement random under-sampling. We plan to implement other under-sampling methods and a hybrid approach with over-sampling.

References

Allyn, J., Allou, N., Augustin, P., Philip, I., Martinet, O., Belghiti, M., Provenchere, S., Montravers, P. and Ferdynus, C. 2017. "A comparison of a machine learning model with EuroSCORE II in predicting mortality after elective cardiac surgery: a decision curve analysis," PLoS One (12:1), p.e0169772.

Brink, H., Richards, J.W., Fetherolf, M. and Cronin, B., 2017. Real-world machine learning (p. 330). Manning.

Das, S. 2001. "Filters, wrappers and a boosting-based hybrid for feature selection," Icml (1), pp. 74-81.

Géron, A. 2017. "Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems," O'Reilly Media, Inc.

Ghassemi, M., Pimentel, M. A., Naumann, T., Brennan, T., Clifton, D. A., Szolovits, P., & Feng, M. 2015. A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data. In Twenty-Ninth AAAI Conference on Artificial Intelligence. PP.446-453

Guyon, I. and Elisseeff, A. 2003. "An introduction to variable and feature selection," Journal of machine learning research, (3) pp.1157-1182.

Han, J., Pei, J. and Kamber, M. 2011. Data mining: concepts and techniques. Elsevier.

Hoogendoorn, M., El Hassouni, A., Mok, K., Ghassemi, M. and Szolovits, P. 2016. "Prediction using patient comparison vs. modeling: A case study for mortality prediction," In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2464-2467. IEEE.

Idoine, C., Krensky, P., Brethenoux, E., Linden, A. 2019. "Magic Quadrant for Data Science and Machine-Learning Platforms," Gartner, Inc.

James, G., Witten, D., Hastie, T. and Tibshirani, R. 2013. "An introduction to statistical learning,"Springer Texts in Statistics. (112) p. 18. New York: springer.

Johnson, A.E., Ghassemi, M.M., Nemati, S., Niehaus, K.E., Clifton, D.A. and Clifford, G.D., 2016. Machine learning and decision support in critical care. Proceedings of the IEEE. Institute of Electrical and Electronics Engineers, (104:2), p.444.

Johnson, A.E., Pollard, T.J., Shen, L., Li-wei, H.L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A. and Mark, R.G. 2016. "MIMIC-III, a freely accessible critical care database," Scientific data, 3, p.160035.

Kaushik, S. 2016. "Introduction to Feature Selection methods with an example (or how to select the right variables?)," Analytics Vidhya.

Kohavi, R. and John, G.H. 1997. "Wrappers for feature subset selection," Artificial intelligence, (97:1-2), pp.273-324.

Le Gall, J.R., Lemeshow, S. and Saulnier, F. 1993. "A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study," Jama (270:24), pp.2957-2963.

Lee, C. H., and Yoon, H. J. 2017. Medical big data: promise and challenges. Kidney research and clinical practice, (36:1), p.3.

Lee, J., Maslove, D.M. and Dubin, J.A. 2015. "Personalized mortality prediction driven by electronic medical data and a patient similarity metric," PloS one, (10:5), p.e0127428.

Li, D.C., Liu, C.W. and Hu, S.C., 2010. A learning method for the class imbalance problem with medical data sets. Computers in biology and medicine, (40:5), pp.509-518.

Luo, Y., Xin, Y., Joshi, R., Celi, L., & Szolovits, P. 2016. Predicting ICU mortality risk by grouping temporal trends from a multivariate panel of physiologic measurements. In Thirtieth AAAI Conference on Artificial Intelligence.pp.42-50.

Malley, B., Ramazzotti, D. and Wu, J.T.Y., 2016. Data pre-processing. In Secondary Analysis of Electronic Health Records, pp. 115-141. Springer, Cham.

Morid, M.A., Sheng, O.R.L. and Abdelrahman, S. 2017. "PPMF: A patient-based predictive modeling framework for early ICU mortality prediction," arXiv preprint arXiv:1704.07499.

Müller, A.C. and Guido, S. 2016. "Introduction to machine learning with Python: a guide for data scientists," O'Reilly Media, Inc.

Panthong, R. and Srivihok, A. 2015. "Wrapper feature subset selection for dimension reduction based on ensemble learning algorithm," Procedia Computer Science (72), pp.162-169.

Stylianou, N., Akbarov, A., Kontopantelis, E., Buchan, I. and Dunn, K.W. 2015. "Mortality risk prediction in burn injury: Comparison of logistic regression with machine learning approaches," Burns (41:5), pp.925-934.

Valverde-Albacete, F.J. and Peláez-Moreno, C. 2014. "100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox," PloS one, (9:1), p.e84217.

Vincent, J.L., Moreno, R., Takala, J., Willatts, S., De Mendonça, A., Bruining, H., Reinhart, C.K., Suter, P. and Thijs, L.G. 1996. "The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure," Intensive care medicine (22:7), pp.707-710.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y. and Zhou, Z.H. 2008. "Top 10 algorithms in data mining," Knowledge and information systems (14:1), pp.1-37.