Julen Berrueta Llona

Early Detection and Prevention of Pneumonia in ICU Patients: Developing a Decision Support Tool Using Clinical Information System Data and Real-Time Infection Mapping

Master's Thesis

Supervised by Dr María Bodí Saera and Dr Josep Gómez Alvarez

Master's Degree in Biomedical Data Science

Tarragona

2024

Abstract

This thesis explores the development of a decision support tool for early detection and prevention of Ventilator-Associated Pneumonia (VAP) in ICU patients, utilizing patient-specific data including demographic details, laboratory test outcomes, drug histories, comorbidities, and ventilation-related metrics. Variables related to comorbidities, demographics, and those associated with the patient's status 24, 48, and 72 hours before the occurrence of the event were added to the study. The event day was defined as the day when the patient developed pneumonia. For patients who did not develop pneumonia, the last date of ventilation was considered as the reference point. Through comparative analysis of Random Forest, Logistic Regression, and SVM models, the Random Forest model demonstrated superior performance. The model achieved notable accuracies of 0.85, 0.79, and 0.73 for predictions made within the 24-hour, 48-hour, and 72-hour time windows before the onset of pneumonia, respectively. Alongside, it demonstrated high recall rates of 0.86, 0.79, and 0.79 for these intervals, further underlining its robust predictive capability in early pneumonia detection. A pipeline has been integrated into a real-time infection dashboard for ICU environments. This dashboard visualizes patient trends and infection-related data, acting as a decision support tool. The probabilities predicted by the model will remain blind for physicians until the model is validated prospectively.

Keywords: ventilator-associated pneumonia (VAP); machine learning (ML); intensive care unit (ICU); dashboard.

Acknowledgments

To Dr. María Bodí Saera, Dr. Josep Gómez Alvarez and Dr. Alejandro Rodriguez Oviedo, for their help in the planning, information and organization of this Master's Final Thesis. For their constant support, guidance and trust in accompanying me throughout the course of the project. Thank you for the opportunity you have given me to continue my research, and the road ahead.

I would also like to thank my family, especially my parents and my sister, for the unconditional support they have always given me, for their encouragement to continue despite the difficulties and for their unquestionable affection. Also, thanks to the friends, specially to Agnès, who have been concerned throughout the whole process.

Dr. María Bodí Saera, certifies that the student Julen Berrueta Llona has elaborated the work under her direction and she authorizes the presentation of this Master's Thesis for its evaluation.

Advisor(s) signature:

Dr. Josep Gómez Alvarez, certifies that the student Julen Berrueta Llona has elaborated the work under his direction and he authorizes the presentation of this Master's Thesis for its evaluation.

Advisor(s) signature:

Contents

1 Introduction

Ventilator-associated pneumonia (VAP) refers to inflammation of the lung parenchyma caused by an infectious agent acquired specifically as a result of invasive mechanical ventilation [1]. The primary risk factor for VAP is the use of an endotracheal tube, which disrupts normal airway defences and facilitates microaspiration of contaminated secretions. Hospital stays, illness, and antibiotic treatments can lead to rapid oropharyngeal colonization by aerobic Gram-negative bacteria, enhancing the risk. Secretions accumulate above the tracheal cuff, entering the airway and forming a bacterial biofilm on the endotracheal tube, resistant to antibiotics and aiding infection spread. The development of VAP depends on both the virulence of the bacteria and the host's immune response [2].

Despite extensive research into the pathophysiology, epidemiology, treatment, and prevention of VAP, a definitive prevention strategy has not yet been established, making VAP a crucial indicator of the quality of care in clinical and epidemiological assessments [3][4]. A high clinical suspicion of pneumonia should lead to the immediate administration of appropriate antibiotics and delays in antimicrobial treatment increase mortality [2].

VAP not only contributes to significant prevalence (9-27%) and mortality (30–70%) but also increases healthcare costs by requiring additional resource consumption, thereby prolonging intensive care stays and accounting for a substantial proportion of all antibiotic prescriptions. [5][6]. The treatment of patients diagnosed with VAP results in an additional expense of approximately £9000 in the UK [7]. Therefore, an early detection of VAP is essential for mitigating the health repercussions for the patient and reducing the financial implications associated with the condition.

1.1 Problem Statement

It should be emphasized that the challenge of accurately diagnosing VAP is due to the absence of objective and universally accepted diagnostic criteria. This issue is further complicated by the need to assess lung injury in a highly varied patient population. Currently, the diagnosis largely depends on identifying clinical symptoms and signs suggestive of a "lung infection," a method fraught with uncertainty as these indicators are not unique to VAP and can be seen in numerous other medical conditions [8]. This highlights a critical gap in our diagnostic capability, emphasizing the need for more precise tools to differentiate VAP from other potential causes of respiratory distress.

This is where the role of Artificial Intelligence (AI) and Machine Learning (ML) becomes crucial, as they enable the detection of nuanced distinctions between individuals with and without pneumonia that are indiscernible through current diagnostic approaches.

1.2 Aim and Objectives

The aim of this thesis is to develop a novel approach for predicting VAP with the potential to forecast occurrences up to 24, 48, and 72 hours in advance. This initiative seeks to address the critical gap in early detection practices by leveraging a model that is not only generalizable and reproducible but also integrates seamlessly with real-time predictive pipelines. The

objectives to achieve are as follows:

1. **To develop a Predictive Model**: A robust, generalizable model will be created, utilizing a minimal set of variables for predicting VAP. This facilitates the model's wide applicability across various clinical settings and patient populations.
2. **To create Real-time Prediction Pipeline:** A pipeline will be implemented that processes real-time data, thus enabling the immediate use of the predictive model in clinical environments. The probabilities won't be shown until model prospective validation.
3. **To design an Infection-related Frontend Dashboard Application:** A dashboard will be designed as a practical tool for clinicians, providing an intuitive interface for tracking patient conditions and assessing VAP risk. It will display critical infection-related data and trends.

1.3 Significance of the Study

VAP is known to extend hospital stays, necessitate the increased use of antibiotics, and require further medical interventions, all factors that escalate healthcare costs significantly. By employing a decision support tool that enhances the early detection and streamlined management of VAP, can help mitigate these expenses.

ICU units often operate at or near full capacity, with a finite number of beds and a limited workforce to care for critically ill patients. A decision support tool that enables the early detection of VAP can help optimize the use of these valuable resources. By preventing the progression of pneumonia in some patients, ICU staff can allocate their time and resources more effectively, potentially improving care for all patients in the unit.

The overuse of antibiotics is a significant concern in hospitals, contributing to the rise of antibiotic-resistant bacteria. A decision support tool that aids in the accurate diagnosis of VAP can support more precise use of antibiotics, aligning with antibiotic stewardship principles. By ensuring that only patients with a high likelihood of VAP receive antibiotics, the tool can help preserve the effectiveness of these critical medications.

The development and implementation of a decision support tool for VAP detection represents a step forward in the integration of technology and healthcare. It sets a precedent for future innovations in the field, encouraging further research and development of tools that harness the power of data analytics to tackle other healthcare challenges.

2 Literature Review

2.1 Current Practices and Limitations

In the challenging environment of Intensive Care Units (ICUs), the detection and prevention of pneumonia VAP remains a critical concern for healthcare providers. The literature identifies several promising biomarkers for the diagnosis of VAP, including C-reactive protein, procalcitonin, and the soluble triggering receptor expressed on myeloid cells. These biomarkers offer potential for early and accurate diagnosis, addressing a significant gap in current clinical practices [9][2].

Clinical criteria encompass a range of symptoms and laboratory findings. A fever greater than 38°C, without another recognized cause, along with either a white blood cell counts below 4,000/µL or exceeding 12,000 µL, signals potential pneumonia. Diagnosis is further supported by at least two of the following symptoms: new or increased purulent sputum, changes in respiratory secretions, cough, dyspnoea, tachypnoea, abnormal breath sounds, or deteriorating gas exchange [10][2].

The microbiological criteria, although optional, add an important dimension to diagnosis, with positive culture results from blood, pleural fluid, or respiratory samples (obtained through methods like bronchoalveolar lavage or protected specimen brush) serving as definitive evidence of infection [10].

A principal risk factor for the development of VAP is the use of endotracheal tubes. The literature highlights that reintubation after unsuccessful extubation, prolonged mechanical ventilation, and certain patient care practices, such as supine positioning and oversedation, significantly increase the risk of pneumonia. These factors, coupled with the absence of a universally accepted definition of VAP, contribute to challenges in diagnosis, leading to potential underdiagnosis or overdiagnosis [2].

Diagnosing pneumonia in the ICU is further complicated by the presence of conditions like pulmonary edema, pulmonary haemorrhage, and acute respiratory distress syndrome, which can mimic the signs and symptoms of pneumonia. This overlap underscores the importance of accurate and timely diagnostic methods to distinguish between these conditions. Clinical criteria alone have been shown to produce a substantial rate of false negatives (30-35%) and false positives (20-25%), thus underscoring the necessity of integrating microbiological samples for diagnosis [2].

Moreover, studies reveal that conventional signs, such as purulent secretions and infiltrates on chest radiography, though they exhibit high sensitivity (87.9% and 85.2%), demonstrate poor specificity for VAP diagnosis [11].

In summary, current practices in detecting and preventing pneumonia in ICU patients are underpinned by a combination of clinical, microbiological, and biomarker-based methods. However, the limitations, including the high risk of false diagnoses and the challenges posed by conditions with similar presentations, highlight the need for further research and development of more accurate diagnostic tools and protocols.

2.2 Clinical Information Systems in Healthcare

Clinical Information Systems (CIS) have increasingly become integral to developing decision support tools and predictive models within healthcare. These systems leverage vast amounts of data collected from patient records, including diagnostics, treatment outcomes, and patient histories, to inform and enhance clinical decision-making processes.

2.3 Decision Support Tools

CIS serve as the backbone for Clinical Decision Support (CDS) systems by providing the necessary data infrastructure for analyzing patient information. CDS tools utilize algorithms and data analytics to offer tailored recommendations to healthcare providers, aiding in diagnosis, treatment options, and patient care plans. These tools are designed to integrate seamlessly into clinical workflows, presenting alerts and reminders to physicians at critical decision points. The effectiveness of CDS tools is contingent on the accuracy and comprehensiveness of the data within CIS, highlighting the importance of robust information systems in healthcare settings [12][13][14].

2.4 Predictive Models

Predictive modelling in healthcare utilizes data from CIS to forecast individual patient outcomes or the likelihood of certain conditions. These models can predict disease progression, response to treatments, and even the risk of readmission. By analyzing patterns within large datasets, predictive models identify risk factors and outcomes associated with different patient cohorts. This predictive capability is pivotal for preventive medicine, allowing healthcare providers to intervene early and tailor treatments to individual patients' needs [15].

The integration of Artificial Intelligence (AI) and Machine Learning (ML) technologies has significantly advanced the development of predictive models and decision support tools. AI algorithms can process and analyze data at a scale beyond human capability, uncovering insights and patterns that may not be immediately apparent. ML techniques, through the analysis of historical and real-time data, improve the accuracy of predictions over time, adapting to new information as it becomes available [16].

2.5 Challenges and Future Directions

Despite the potential of CIS in enhancing patient care through decision support tools and predictive models, challenges remain, particularly in terms of data quality, interoperability, bias, and the ethical use of AI. Ensuring data accuracy and consistency across different systems is critical for the reliability of these tools. Additionally, the need for standardized data protocols to enable seamless data exchange between disparate systems is a significant hurdle. Ethical considerations, including patient privacy and the potential for algorithmic bias, also necessitate careful consideration and oversight [17][18].

2.5.1 Pneumonia predictive models

The quest for accurate predictive models for VAP in ICU patients has led researchers to leverage various data sources and machine learning (ML) techniques, aiming to improve early detection and intervention strategies. This exploration spans the utilization of rich datasets like MIMIC and the Philips eRI dataset, which encompass a wide array of patient information, from demographics to detailed clinical parameters.

Among the machine learning models employed, Random Forest (RF), Decision Trees (DTs), Logistic Regression (LR), Support Vector Machine (SVM), and Multilayer Perceptron (ANN), a type of Artificial Neural Network, have been notably utilized. These models have been applied to predict the likelihood of VAP occurrence within specific time windows after the initiation of mechanical ventilation, focusing on the critical 48-hour mark that significantly raises the risk for VAP. However, RF models have been the most frequently used in the literature review articles.

The predictive power of these models, as measured by the Area Under the Curve (AUC), highlights their potential effectiveness. For instance, one approach aimed to predict VAP incidence at any point after 48 hours of mechanical ventilation initiation, achieving a mean AUC of 79%. Another model employed decision trees, focusing on the imminent risk of VAP within the next 24 hours based on a comprehensive range of patient data, achieving an AUC of 76%. Additionally, a notable effort to predict early VAP detection within the first 24 hours post-intubation reported an AUC of 84%, with sensitivity and specificity of 74% and 71%, respectively. These metrics suggest a promising direction towards refining the accuracy and utility of predictive models in clinical settings [19][20][21].

The diversity in model types and their respective prediction windows underscores the multifaceted approach to tackling VAP prediction. While the pursuit of high AUC values is indicative of model performance, the real-world applicability of these models hinges on their integration into clinical workflows, the interpretability of their predictions, and, critically, their ability to facilitate timely and effective preventive interventions. The ongoing refinement of these models, coupled with advances in data collection and analysis technologies, points towards a future where predictive modelling can significantly impact the management and outcomes of ICU patients at risk of VAP.

2.5.2 Real-time Infection Mapping

Recent advancements in healthcare technology have seen the emergence of dashboards and real-time infection mapping tools specifically designed to combat the prevalence of VAP within Intensive Care Units (ICUs). These innovations mark a significant progression in the domain of infectious disease surveillance and management, employing sophisticated data analytics and visualization techniques to furnish medical personnel with actionable insights crucial for patient care.

These technological advancements harness the power of real-time data visualization, predictive analytics, and clinical decision support to enhance the management of VAP. By merging and visually presenting various real-time data, including patient vitals and ventilator settings, healthcare professionals can continuously monitor for signs of VAP risk. Predictive

analytics, through machine learning algorithms, further refine this process by identifying complex patterns within the data, allowing for the early identification of patients at increased risk of VAP, sometimes even before symptoms appear. Consequently, these tools equip healthcare providers with evidence-based recommendations, enabling timely interventions aimed at reducing the incidence of VAP.

Three promising outcomes have been observed because of applying these maps.

- **Reduction in VAP Incidence:**: The deployment of dashboards and real-time infection mapping tools has been credibly linked with a significant reduction in VAP occurrences within ICUs. The essence of these technologies lies in their ability to enable prompt intervention strategies, an aspect fundamental to the prevention of VAP [22][23].
- **Enhanced Patient Outcomes:** A notable repercussion of diminishing VAP rates has been the overall improvement in patient outcomes. This encompasses a decrease in mortality rates, diminished duration of ICU stays, and a lowered dependency on antibiotics, thereby indirectly confronting the issue of antibiotic resistance [24].
- **Improved Clinical Workflow:** The assimilation of these advanced mapping and predictive tools into everyday clinical practice has demonstrably augmented the operational efficiency and care delivery effectiveness. By granting medical staff a unified and comprehensive overview of patient risk profiles, a paradigm shift towards a more proactive and preventive healthcare approach is facilitated [23].

The incorporation of real-time data visualization, predictive analytics, and clinical decision support mechanisms, as facilitated by modern technological tools, presents a viable and effective strategy in the ongoing battle against VAP in ICU settings. The implications of these developments are profound, paving the way for a more informed, efficient, and outcome-oriented healthcare paradigm.

3 Methodology

3.1 Population Inclusion and Exclusion Criteria

Initially, we selected all ICU patients who had been discharged before January 1, 2024. We refined this pool by removing patients whose type was not categorized as medical, surgical, or traumatic, those whose type of admission was not urgent or scheduled, and any duplicates. We further excluded all patients who had not been ventilated for at least 48 hours, as VAP is considered when a patient has been ventilated for more than 48 hours.

3.2 Data Extraction (ETL)

The patient cohort was performed using a data extraction and transformation (ETL) process, ensuring that only cases that met all criteria were included in the population. This ETL process was pivotal in refining the initial dataset of ICU patients.

3.2.1 Docker environment

A Docker environment was developed to efficiently support data extraction processes. The Dockerfile, specifies a base image from jupyter/scipy-notebook with a specific tag that ensures consistency and reproducibility in the Python environment provided by Jupyter. This base image is enhanced by installing necessary drivers and libraries, such as those for Oracle and Java, which are crucial for connecting to and interacting with the Centricity Critical Care (CCC) database.

By using Docker's containerization technology, applications can be packaged and run in a loosely isolated environment known as a container, which is lightweight and contains everything needed to run the application. This ensures that the application can run consistently across different computing environments. Docker's tools and platform support the entire development lifecycle, from development and testing to deployment and scaling, allowing a consistent environment from a developer's local machine all the way to the cloud [25].

3.2.2 VAP

As previously mentioned, VAP is only considered if the patient has been mechanically ventilated for at least 48 hours. This means that if a case was recorded as VAP without the patient having been ventilated for 48 hours, it would not be considered valid.

For each patient who developed pneumonia, we designated the onset date of the infection as date_0. For those who did not acquire this pneumonia, date_0 was marked as the last date they had been on mechanical ventilation. We calculated the dates known as date_1, date_2, and date_3 to represent 24 hours before date_0, 48 hours before date_0, and 72 hours before date_0, respectively. The 24-hour window between date_0 and date_1 was considered, as was the 48-hour window between date_1 and date_2, and the 72-hour window between date_2 and date_3. For a comprehensive illustration of the temporal relationship between the onset of ventilator-associated pneumonia and mechanical ventilation, refer to Figure 1.

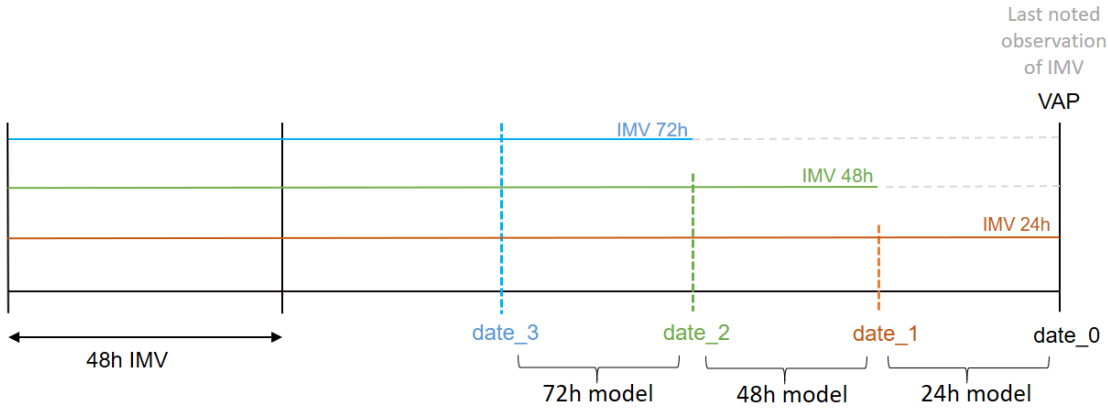

Figure 1. Design of time windows for extracting predictor variables for predictive models. Note that patients who did not have VAP, last noted observation of the IMV was set as date_0.

It is important to emphasize that "24h model", "48h model" and "72h model" are models trained with the data windows 24h, 48h and 72h prior to pneumonia respectively. In case the patient did not have pneumonia, we performed the same process but from the last time it was ventilated. The "24h model" does not imply that it can predict events 24 hours before they occur, but rather that it makes predictions based on data from the 24-hour window prior to the event. Conversely, the "48h model" is designed to predict outcomes 24 hours before the occurrence of VAP, using data from up to 48 hours prior.

3.2.3 Variables of Interest

The selection of independent variables was conducted in collaboration with medical experts to ensure clinical relevance and accuracy. This interdisciplinary approach was pivotal in defining variables that are essential for analyzing the impact of VAP. These variables were related to those discussed in the state of the art, and we also included some additional ones that seemed interesting and had not been used yet.

To illustrate the relevance of the independent variables used in this study, the table below provides a detailed overview. It lists each variable's name, description and the type, along with the statistical measures obtained for them, such as minimum, maximum, and median values. Variables marked with * have been calculated for three different time windows: 24 hours, 48 hours, and 72 hours.

Variable name	Description of the variable	Type	Measures
Sex	The biological sex of the patient.	Categorical	(male = M, female = F)
Admission type	Category of the patient's admission.	Categorical	(urgent, scheduled)
Patient type	Indicates the type of patient in the admission.	Categorical	(medical, surgical, traumatic)
Age	The patient's age, in years.	Numerical	last

Follows in the next page

Variable name	Description of the variable	Type	Measures
Tracheostomy*	Indicates whether the patient has had a tracheostomy.	Categorical	(1, 0)
VAP	Ventilator-associated Pneumonia.	Categorical	(1, 0)
Length of stay*	The number of days the patient has been admitted in the ICU.	Numerical	cumulative
Comorbidities (27)	A list or count of other existing diseases the patient has alongside the primary disease.	Categorical	(1, 0)
Leukocytes*	White blood cell count.	Numerical	median
Lymphocytes*	A type of white blood cell.	Numerical	median
PCR*	Polymerase Chain Reaction, a test to detect genetic material from a specific organism, such as a virus or bacteria.	Numerical	median
Creatinine*	A waste product measured in the blood to assess kidney function.	Numerical	median
Glucose*	Blood sugar level.	Numerical	max, min, median
SOFA*	Sequential Organ Failure Assessment score, to track a patient's status during ICU stay.	Numerical	median
APACHE II	Acute Physiology and Chronic Health Evaluation II, a severity-of-disease classification system.	Numerical	first
VMI Days*	Days on Ventilator Mechanical Intervention.	Numerical	cumulative
Heart Rate*	Beats per minute of the heart.	Numerical	max, min, median
Respiratory Rate*	Breaths taken per minute.	Numerical	max, min, median
Blood Pressure*	The force of blood pushing against the walls of the arteries	Numerical	max, min, median
Body temperature*	The internal temperature of the body.	Numerical	max, min, median
Number of aspirations*	Counts of attempts to remove accumulated secretions from airways or lungs.	Numerical	cumulative
Diuresis*	Volume of urine produced by the kidneys.	Numerical	cumulative
Intravenous antibiotics*	Indicates whether the patient is receiving antibiotics directly into a vein through a needle or catheter.	Categorical	(1, 0)
Positive End-Expiratory Pressure (PEEP)*	A mode in mechanically ventilated patients to maintain airway pressure above atmospheric pressure.	Numerical	max, min, median

Follows in the next page

Variable name	Description of the variable	Type	Measures
Peripheral Oxygen Saturation (SpO2)*	The level of oxygen saturation in the blood.	Numerical	max, min, median
Inspired Fraction of Oxygen (FIO2)*	The concentration of oxygen being inhaled by the patient.	Numerical	max, min, median
SpO2/FiO2 ratio*	A calculated ratio used to assess oxygenation and lung function.	Numerical	median
Partial Oxygen Pressure (PaO2)*	The amount of oxygen gas in the blood, indicating how well oxygen is being dissolved and transported.	Numerical	median
PaO2/FiO2 ratio*	It evaluates the efficiency of oxygen transfer in the lungs.	Numerical	median
Consistency of secretions*	Refers to the thickness or viscosity of secretions.	Categorical	(thick, fluid, mucus plugs, other)
Appearance of secretions*	Describes the colour, clarity, or any other visual aspect of secretions.	Categorical	(purulent, non-purulent)

Table 1. Overview of cohort variables used in the study.

3.2.4 Data Extraction and Preprocessing

Before delving into the data processing methods used after cohort creation, it is crucial to outline the approach we took for outlier management. This step was implemented during the data extraction phase because, when determining the maximum or minimum values of certain variables, outliers must be removed prior to these calculations. Calculating the median for these variables would not need outlier removal, as the median remains unaffected by outliers. Additionally, addressing outliers prior to data aggregation was strategically beneficial. Performing this step later would have resulted in numerous null values and the loss of many data points that were essential for accurately computing the minimum and maximum medians. Three methods were considered for outlier removal: The Z-Score method, Extreme Percentiles method, and Median Absolute Deviation (MAD).

- The Z-Score method identifies outliers based on their distance from the mean in standard deviations. This method works best for normally distributed data but is less effective for non-normal data and extreme outliers.
- The Extreme Percentiles method sets thresholds at the data distribution tails to identify and remove outliers. This method does not assume normal distribution but can result in significant data loss if thresholds are too conservative.
- The Median Absolute Deviation (MAD) method is robust for non-normal distributions. It calculates the median of the absolute deviations from the data's median, making it less sensitive to extreme values. However, it can be computationally intensive and requires careful threshold selection.

We selected the Median Absolute Deviation (MAD) for outlier removal due to its robustness and applicability across various types of data distributions, which suited the complex clinical data involved in my study.

The processing of variables varied significantly among them. For instance, to calculate the maximum, minimum, and median of a particular variable within the 24-hour, 48-hour, and 72-hour windows, the following method was used: for the 24-hour window, all values between date_0 and date_1 were collected—date_0 being the outcome date (if the patient had pneumonia, it would be the date of the pneumonia onset; if not, it would be the last date the patient was ventilated as previously explained). The minimum, maximum, and median were then calculated. For the 48-hour window, we collected the values between date_1 and date_2 and processed them similarly, and for the 72-hour window, we gathered values between date_2 and date_3.

In situations where all values up to a certain time window need to be considered, such as the number of days a patient has been on invasive mechanical ventilation or the length of stay, we performed the calculation by aggregating the total days of ventilation up to specific dates: up to date_0 for the 24-hour period, up to date_1 for the 48-hour period, and up to date_2 for the 72-hour period. This variable was a more complex one to derive, as it was calculated based on observations noted by the nursing staff. Specifically, the nursing staff periodically recorded the ventilatory mode in which the patient was placed. Once we obtained the data, we filtered it to include all records prior to date_0, date_1, or date_2, depending on the specific time window we were extracting. We then calculated the time differences between a value noted as IMV and the subsequent recorded ventilatory setting, regardless of what that next setting was. This provided intervals representing the duration of mechanical support. Finally, we summed these intervals to determine the total days a patient had been mechanically ventilated.

Other variables such as intravenous antibiotics, involved another scenario. Values were collected and categorized by 24-hour windows; if a patient had any recorded use of antibiotics (regardless of dosage), they were assigned a value of 1, otherwise, they received a 0. Other variables, like the APACHE II score, were determined by taking the first recorded value during the stay. In the case of number of aspirations, we had to use regular expressions since this information was filled out in a text field. In this way, we extracted a numeric value from free text.

The PaO2/FiO2 ratio case needs to be highlighted and caution was required. The PaO2/FiO2 ratio, also known as the oxygenation index, is a calculation used in medicine to help assess how efficiently the lungs can transfer oxygen from the air to the blood. It is particularly useful for evaluating the severity of respiratory problems. This ratio is calculated by dividing the arterial partial pressure of oxygen (PaO2) by the fraction of inspired oxygen (FiO2), which is the concentration of oxygen a person is inhaling. For example, ambient air has an FiO2 of approximately 0.21 (or 21% oxygen), but this value can be higher if the patient is receiving supplemental oxygen. In our ICU, when PaO2 is measured, it should ideally be obtained from arterial blood. However, it is often unclear whether the sample is arterial or venous. As a result, we predominantly utilize the SpO2/FiO2 ratio instead, which is why it has been included in our measurements. Even so, we decided to estimate the value of PaO2 based on the SpO2 (Peripheral Oxygen Saturation). SpO2 is a monitored parameter that ranges from

96 to $\tilde{1}00$ in normal conditions. For this reason, patients who had a null value in the median oxygen saturation variable (this condition was not applied to the minimum and maximum variables), we chose to fill it with a random value between 96 and 100, inclusive of 96 and 100. Once SpO2 values were processed, we imputed the PaO2 based in the following formula:

$$PaO_2 = \left(\frac{28.603^3}{\left(\frac{1}{SpO_2} - 0.99 \right)} \right)^{\frac{1}{3}} \quad (1)$$

[26]

Once the estimation of the PaO2 variable was performed, we calculated the median PaO2/FiO2 ratio. Only the median was calculated since estimating the minimum and maximum values is challenging when the measurement is not directly taken from the patient.

The situation with FiO2 (Fraction of Inspired Oxygen) was like that of oxygen saturation, with the key difference being that a null value for FiO2 indicates ambient air. The FiO2 level of ambient air is 21%, so for the FiO2 variable, all values—whether median, minimum, or maximum—were filled with '21'.

All comorbidities were recorded as '1', meaning if a patient had any of the comorbidities extracted during the ETL process, they were assigned a '1'. Conversely, patients without recorded comorbidities were initially left as null. Therefore, those patients whose records remained null were subsequently filled with a '0' to indicate the absence of the noted comorbidities.

Finally, we transformed the categorical variables into dummy variables. This is necessary because models in Python cannot handle categorical variables directly. To make them usable, consider that a patient is of the 'medical' type. We know the patient type variable can be medical, surgical, or traumatic. What is done is to create a column for each category of the variable. Therefore, in this case, since the patient is medical, the table would appear as medical = 1, surgical = 0, and traumatic = 0, as follows:

Table 2. Example of PatType dummy variable.

PatType	PatType_Medical	PatType_Surgical	PatType_Traumatic
	1	0	0

The percentage of non-null values for each variable was calculated, indicating the percentage of patients who had data for that variable. Variables with less than 40% of non-null values were removed from the analysis.

3.3 Exploratory Data Analysis (EDA)

In this section of the thesis, we delve into the Exploratory Data Analysis (EDA) that we conducted to gain a deeper understanding of the dataset at hand. EDA is an essential analytical step that precedes more formal statistical modelling. It involves deploying a range

of techniques aimed at uncovering the underlying structure of the data, identifying key variables, detecting outliers and anomalies, and testing assumptions that could influence further analyses.

The primary purpose of the exploratory data analysis was to see what insights the data could reveal beyond the confines of formal hypothesis testing or predictive modelling. This involved utilizing various graphical and statistical methods to summarize the data. These visualizations were pivotal in offering a visual understanding of the data's distribution, trends, and relationships between variables.

In this part of the thesis, we realized which variables might be influential when selecting predictors. However, most of the graphs and comparisons were straightforward and simple, making it difficult to visualize multiple and non-linear relationships.

It should be noted that independent variables may be correlated with each other. For linear models, these correlations are undesirable because they introduce a phenomenon known as multicollinearity. Multicollinearity is a phenomenon where two or more independent variables in a statistical model are highly correlated with each other. In simple models, such as linear regression with a single independent variable, multicollinearity is not a direct problem, as there is only one independent variable to consider. However, in a multiple linear model, such as multiple LR, multicollinearity can be a significant issue. This occurs when two or more independent variables in the model are highly correlated. Multicollinearity can negatively affect the model's ability to estimate coefficients accurately and reliably, which in turn can impact the interpretation of the relationship between the independent variables and the dependent variable.

Although linear SVMs operate within the framework of linear models, they demonstrate a notable resistance to multicollinearity compared to LR. While LR directly models the probability of a binary outcome, SVMs adopt a different approach by seeking the hyperplane that optimally segregates classes within the feature space. This doesn't mean that multicollinearity is completely irrelevant for SVMs. Highly correlated features can still affect the performance of SVMs to some extent, but they typically have a lesser impact compared to LR. Nonetheless, it's generally a good practice to preprocess the data to reduce multicollinearity before training the model.

Then, multicollinearity is a factor to consider when designing a model and choosing the desired features, as different treatments are required depending on the model to be used. This understanding shaped the approach to the subsequent phases of model development and feature selection in the thesis.

3.4 Model Development

In the model development section of the thesis, we worked on constructing and evaluating three distinct types of predictive models to investigate their effectiveness in predicting VAP. Each model represents a different approach to the problem, utilizing various strengths and capabilities within the realm of machine learning.

The first model we explored was the RF model. Known for its robustness and ability to handle unbalanced data, the RF model is an ensemble learning method that constructs multiple

decision trees during training and outputs the class that is the mode of the classes of the individual trees. This method is particularly beneficial for its performance and interpretability in handling complex datasets with potential interactions between variables.

We also tested a LR model. As a traditional statistical model used for binary classification, LR provides a probabilistic approach to model development. It estimates the probability of an event occurring by fitting data to a logistic function, offering a clear interpretation of the relationship between independent variables and the binary outcome. This model is valued for its simplicity and efficiency in providing probabilities that are easy to understand and apply in clinical decision-making.

Finally, we implemented both linear and non-linear SVM models. SVM is a powerful classifier that determines the best hyperplane to separate different classes in the feature space. It is particularly suitable at managing high-dimensional data and is known for its accuracy and effectiveness in classification tasks where the margin of separation between classes is crucial.

For the data split phase of this study, the data was systematically divided into three distinct sets: a training set (X_train, y_train), a validation set (X_val, y_val), and a test set (X_test, y_test). We allocated 80% of the data for training, 10% for validation, and 10% for testing, except for the LR model which divided the data into 80% to training and 20% to testing. This allocation was implemented in such a way that the distribution of the positive NAV class remained consistent across all three sets to ensure that each subset accurately reflects the overall dataset characteristics.

This structured approach to data division is critical for developing a robust model, as it ensures that the model is trained, validated, and tested on different subsets of data. Maintaining a consistent distribution of the VAP class across all sets is particularly important for preserving the integrity of the model's evaluation, thereby preventing the model's performance metrics from being influenced by disparities in data distribution.

3.4.1 Feature selection

Feature selection significantly varied depending on the data window utilized. For each model, different combinations of predictors were used as their importance shifts closer to the occurrence of VAP. To refine our feature selection, we relied on insights from our exploratory data analysis, such as correlation matrices, box plots, and trend analyses, as well as feature importance rankings derived from RF and XGBoost algorithms, and prior knowledge. This last aspect is crucial as it incorporates domain expertise; sometimes a variable might not appear analytically significant, yet its inclusion can reveal non-linear correlations that automated methods alone might miss.

For each type of model developed in this section, we applied different feature selection strategies tailored to the characteristics of the model. The RF model, being a non-linear model, allows for the inclusion of correlated independent variables without significant concerns about multicollinearity. This flexibility makes RF particularly useful for exploring complex interactions within the data.

In contrast, for the LR model, we needed to carefully manage the feature selection process due to the issues of multicollinearity discussed earlier. LR, being a linear model, is

sensitive to high correlations among predictors, which can distort the estimated coefficients and compromise the model's interpretability and predictive accuracy. For the LR we decided to keep only one of the correlated variables.

For the SVM model, the approach to feature selection depended on whether the model was implemented in a linear or non-linear form. When using SVM in its non-linear form, we adopted a feature selection approach similar to that used for RF, allowing a broader inclusion of variables. Conversely, when deploying SVM in its linear form, we followed a more restrictive feature selection like that used for LR to avoid the pitfalls of multicollinearity.

Despite the effectiveness of automated algorithms for feature selection, the nuanced understanding of variables through expert knowledge remains superior. We tested many variable combinations to identify the most predictive features. Ultimately, given the large number of variables—50 for each window—it was impractical to test all possible combinations. For instance, testing all combinations of these variables would result in 250 (approximately 1.125 quadrillion) different sets, illustrating the computational challenge as evaluating such combinations follows an exponential complexity, O(2^n), where n is the number of variables. This is known as exponential complexity, in stark contrast to linear complexity, O(n), which would involve 50 combinations, or quadratic complexity, O(n2), resulting in 2500 combinations. However, the exponential growth in complexity is vastly more demanding than the others and, in this case, executing such extensive evaluations could be computationally endless.

3.4.2 Missing Values

When handling missing data, we employed the Iterative Imputer from Scikit-Learn, tailoring it specifically to each subset of variables selected for our models. Instead of applying imputation across all variables, we strategically chose to impute only those variables included in each specific model. This selective imputation approach ensures that the imputation process is directly relevant and accurately aligned with the variables actively used in the modelling. Moreover, this method enhances the effectiveness and precision of our data preparation steps, which is crucial for setting up an automated pipeline for model prediction. This automated pipeline allows for seamless model deployment and real-time predictions, making the modelling process both efficient and robust.

3.4.3 Dealing With Class Imbalance

The positive VAP class was significantly smaller than the negative class, resulting in an imbalance cohort. When training a model, this imbalance must be considered. Training a model with this class distribution would likely result in the model classifying most instances as NO VAP, due to learning more from the more frequent negative cases. To minimize this effect, several methods can be employed, including upsampling, downsampling, or generating synthetic data for upsampling. Each approach has its advantages and disadvantages.

Upsampling involves replicating instances of the minority class to balance the majority class, which preserves all original data but can lead to overfitting and increased training time due to data duplication. Downsampling, on the other hand, reduces the number of majority class instances to match the minority class, which can lead to the loss of valuable

information and potentially reduce the overall accuracy of the model. Generating synthetic data creates new, varied instances of the minority class, which can enhance learning and model generalization. However, producing high-quality synthetic data is complex and may introduce noise and biases if not executed properly. These techniques are critical for addressing class imbalance and improving model performance.

Ultimately, we chose the downsampling method as it obtained the best results for the models.

3.4.4 Hyperparameter Tunning

We employed the scikit-learn library for the RF and SVM models, and the statsmodels library for LR. For hyperparameter tuning, we utilized the Optuna library. Optuna is an open-source optimization library specifically designed to automate the optimization of hyperparameters in machine learning models. It works by efficiently searching through the parameter space to find the most effective values, using techniques like Bayesian optimization, tree-structured Parzen estimators, or evolutionary algorithms. Optuna streamlines the process by providing a framework for defining a search space and managing the trial process to evaluate different configurations. This approach helps in identifying the optimal set of parameters that result in the best model performance, enhancing both the efficiency and accuracy of the machine learning models [27].

In our case, as we prioritized accurately predicting VAP cases, we created a combined score. This combined score was a mix of the accuracy and recall of the model, with an adjustable parameter alpha that could be increased or decreased to give importance to either of the two scores. . The formula we used is as follows:

$$combined\_score = \alpha \cdot recall + (1 - \alpha) \cdot accuracy \tag{2}$$

This approach was chosen to emphasize the importance of identifying as many positive VAP cases as possible, reflecting the critical need in medical applications to avoid false negatives. The combined score was calculated to reward high recall while still considering overall accuracy, thus providing a balanced measure of the model's effectiveness in classifying VAP instances. This scoring strategy supports the clinical goal of maximizing sensitivity (recall) without disregarding the overall correctness of the model's predictions (accuracy), which is essential in ensuring the utility and reliability of the predictive model in a healthcare setting.

For developing the RF models, different hyperparameters were carefully considered and dynamically tuned to optimize their performance. The search space for these parameters included categorical choices for 'max_depth', 'min_samples_leaf', 'min_samples_split', and 'n_estimators'. Specifically, 'max_depth' varied across a wide range of values from 10 to 100, or unrestricted (None), allowing the trees to potentially grow deep depending on the complexity of the data. The 'min_samples_leaf' and 'min_samples_split' were selected from smaller sets of values ranging from 1 to 20 to ensure that the trees do not overfit by creating overly specific leaves. The 'n_estimators' parameter was tested from 200 to 1000 to determine the optimal number of trees in the forest, balancing computational efficiency with predictive performance.

For LR, no hyperparameter optimization was performed. Consequently, the dataset was

directly split into 80% for training and 20% for testing, without a validation set.

In configuring the parameters for SVM models, the regularization parameter 'C' was applied consistently across both linear and non-linear models. This parameter, which helps control the complexity of the model by allowing flexibility to fit the data, was set with potential values ranging from 0.1 to 1000, providing a broad range to assess the impact of regularization. The parameter 'gamma', which determines the influence of individual training examples on the decision boundary, was also configured similarly in both models, offering the choices of 'scale' and 'auto'. These options automatically adjust the value of gamma based on the dataset's characteristics, which is particularly crucial for non-linear kernels.

For the linear SVM model, exclusively the 'linear' kernel was specified, suitable for data presumed to be linearly separable or when a simpler, faster model is preferred. Conversely, for the non-linear SVM model, options for the kernels included 'poly' (polynomial), 'rbf' (radial basis function), and 'sigmoid'. These kernels enable the model to handle more complex and non-linear data patterns, better adapting to various data distributions that do not fit well with linear models. Non-linear SVMs are generally more computationally intensive than linear SVMs because the calculation of these kernel transformations involves more complex mathematical operations and the optimization problem solved is more complex due to the non-linear boundaries that need to be modelled.

3.4.5 Random Forest (RF)

RF is a widely used machine learning ensemble technique that aggregates the outcomes of numerous decision trees to produce a singular conclusion. Known for its straightforward application and versatility, it effectively addresses both classification and regression challenges. This algorithm expands upon the concept of bagging, incorporating both bagging and feature randomization to generate a collection of uncorrelated decision trees. Bagging, an ensemble method, is typically utilized to reduce variance in datasets that exhibit high variability, enhancing the stability and accuracy of the predictive models it supports [28].

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g., whether a patient has tracheostomy or not), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The components of a decision tree are:

- **Target Variable:** This is the variable that the decision tree aims to predict, based on several input variables. In this case is "VAP (1) or NO VAP (0)".
- **Root Node:** This node initiates the splitting process. It identifies the attribute that best divides the target variable into its most distinct classes.
- **Node Purity:** Nodes within the tree can be impure, containing a mix of classes. A pure node contains only one class, making it homogeneous and typically signifies that no further splitting is necessary.
- **Decision Nodes:** These are the nodes resulting from a split at the root or another decision node. They represent the point at which the subset of the dataset is further

split on another attribute. This process is repeated until significant criteria are met, such as achieving node purity or reaching a maximum tree depth.

- **Leaf Nodes (Terminal Nodes):** These nodes are where the tree makes a final decision. Usually, they are pure nodes from which the outcome of the prediction is derived based on the path followed from the root to that leaf.

Decision trees utilize a top-down, greedy approach known as recursive binary splitting. This method is considered top-down because it starts at the top of the tree (the root node) and works its way down by successively splitting the predictor space—each split divides the space into two, represented by two new branches extending downwards from the node. The approach is labelled as greedy because at each step, the algorithm chooses the best split at that particular point without considering future implications; it optimizes for the best immediate result rather than a potentially more optimal overall tree structure [29].

For handling categorical variables, even though they are typically encoded as binary (1 and 0), the splitting process treats them like any other variable. The tree will still evaluate whether splitting on a categorical variable at any given node will effectively separate the data into purer subsets, improving the model's prediction accuracy. This capability allows decision trees to handle a mix of numerical and categorical data effectively, adapting to various data types within a dataset.

The selection of the best variable for node splitting is guided by statistical measures, particularly entropy, which quantifies the disorder or impurity within a node. The algorithm evaluates each variable against this criterion to identify the one that optimally splits the data at each decision node.

Entropy serves as a critical measure in this process. It reflects the level of uncertainty or randomness in the composition of a node. For instance, a node containing an equal number of VAP and NO VAP cases would exhibit high entropy, typically quantified as 1, indicating a high degree of disorder. Conversely, a node with homogenous outcomes (all VAP or all NO VAP) represents low entropy, quantified as 0, signifying no disorder. This entropy-based criterion ensures that Decision Trees progressively move towards more orderly divisions at each step of the splitting process, enhancing the clarity and accuracy of the predictive model [29].

The formula of entropy is:

$$E = -\sum_{i=1}^{n} p_i \cdot log_2(p_i) \tag{3}$$

Where *pi* is the probability of randomly selecting an example in class *i*.

To determine the root node, a Decision Tree calculates the entropy for each variable along with its potential splits. This involves identifying each possible division within a variable, calculating the entropy for each resulting node after the split, and then determining the average entropy across these nodes. The reduction in entropy compared to the original, unsplit node is crucial and is referred to as Information Gain. Information Gain measures the reduction in uncertainty or disorder regarding the target variable that results from splitting on a particular feature. It quantifies the amount of information each feature provides about the class label,

guiding the selection of the most informative feature at each decision point in the tree. Then, the goal is to minimize the entropy of the child nodes as much as possible, which in turn maximizes the Information Gain, leading to a more accurate and efficient decision-making process in the model.

$$Information\ Gain = Entropy_{parent} - Entropy_{children} \tag{4}$$

After each new node is created, the same process of evaluating and splitting based on entropy continues. Each node carries its own entropy, which will as parent entropy to calculate the information gain for further nodes. This division process continues until the node becomes a Leaf Node (Terminal Node) or until we reach the maximum depth set in the model's parameters. This structured progression ensures that each split maximizes the clarity and purity of the classifications, ultimately aiming to create as homogenous groups as possible within each node to enhance the model's predictive accuracy.

3.4.6 Logistic Regression (LR)

LR is a widely utilized statistical method for binary classification that can be extended to handle multiple categories. It is particularly effective for predicting the probability of a binary outcome. At its core, LR models the probability that a given input belongs to a particular category. It is fundamentally a linear regression model adapted to model binary outcomes by incorporating the logistic function.

The logistic function, also called the sigmoid function, outputs a probability score between 0 and 1. The model uses predictors to estimate the odds of the target being in a particular class as opposed to the reference class. This is expressed as:

$$logit(p) = log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n \tag{5}$$

where *p* is the probability of the dependent event occurring, β_0, β_1, ..., β_n are the coefficients, and x_1, x_2, ..., x_n are the predictor variables. We can calculate the Odds Ratios for each variable with e^{β} where β is the coefficient for each variable. The Odds Ratios can be interpreted as the multiplicative change in the odds of the outcome for a one-unit increase in the corresponding predictor variable, holding all other variables constant. These provide insights into the strength and direction of the association between each predictor and the outcome, aiding in understanding the relative impact of each variable on the likelihood of the outcome occurring.

The coefficients of the LR model are typically estimated using Maximum Likelihood Estimation (MLE), a method that estimates the parameters which maximize the likelihood of observing the sample given the parameters.

In LR, the treatment of variables is adjusted to suit the model's requirements, given its capability to handle different types of predictors. Binary predictors are incorporated directly into the model, similar to their usage in linear regression, where they are entered without modification. Continuous predictors, on the other hand, might be used as they are or

transformed depending on their relationship with the log-odds of the outcome variable. Such transformations help in linearizing relationships with the log-odds or in handling non-linear effects. Categorical predictors are generally converted into dummy variables to fit into the regression model effectively. This process involves creating indicator variables that represent the categories numerically, allowing the regression algorithm to interpret these categorical inputs correctly.

In LR, the presence of multicollinearity, which involves high correlation among predictor variables, can significantly distort the reliability and stability of the estimated coefficients. This complication necessitates the implementation of careful feature selection or the use of regularization techniques to mitigate the impact on the model's accuracy.

3.4.7 Support Vector Machine (SVM)

SVMs are a set of powerful, versatile machine learning models capable of performing linear and nonlinear classification. By efficiently handling both linear and nonlinear data, SVMs are extensively used in classification problems where the goal is to find the best separating boundary, or hyperplane, between data points of different classes.

Linear SVMs focus on finding the optimal hyperplane that separates data points of different classes with the maximum margin. This hyperplane is defined in a high-dimensional space, and the goal is to ensure that the distance between the nearest data points (support vectors) from each class to the hyperplane is maximized. The advantage of linear SVMs lies in their simplicity and effectiveness, particularly suitable for linearly separable data.

The key technique in linear SVMs is to solve an optimization problem that minimizes the norm of the weight vector, which correlates to maximizing the margin between classes. Regularization is often employed to avoid overfitting, typically through penalties on the size of the coefficients (L2 regularization).

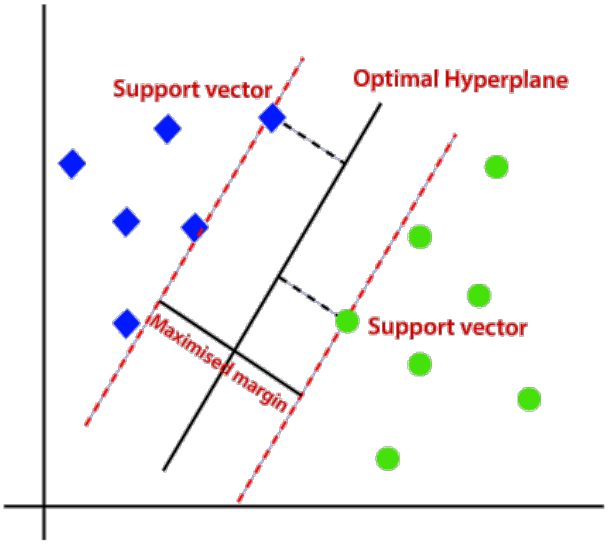

Figure 2. Optimal hyperplane and margin maximization in SVM [30].

When data is not linearly separable, non-linear SVMs come into play by employing kernel functions to map the original features into a higher-dimensional space where a linear separation is possible. This approach allows SVMs to construct non-linear decision boundaries as seen in Figure 3. The kernel trick is central to the functionality of non-linear SVMs. Common kernels include the polynomial kernel, Radial Basis Function (RBF) kernel and sigmoid kernel.

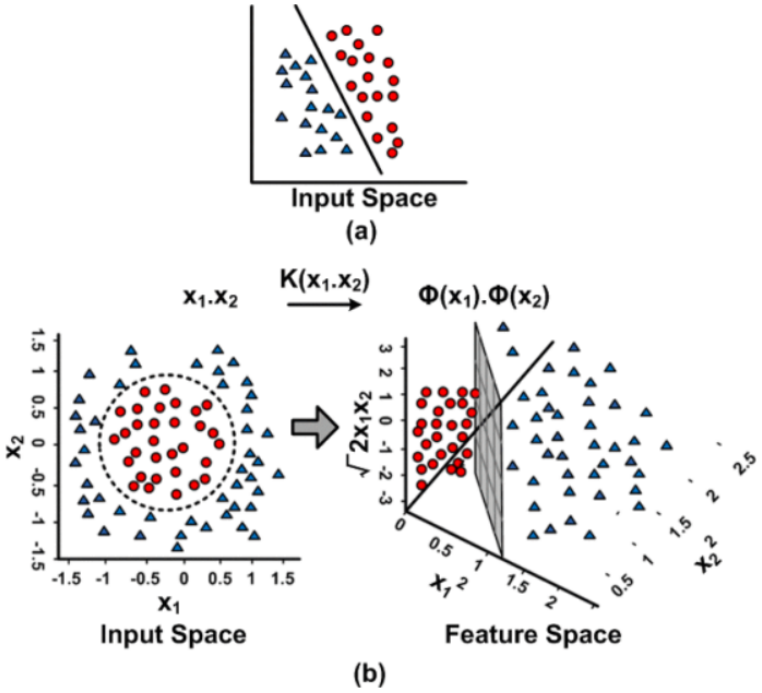

Figure 3. Visualization of SVM classification in input and feature spaces [31].

3.5 Model Implementation in Near Real-Time Pipeline

Once the models were developed, the next step involved integrating them into an automated, near real-time pipeline for ICU patients. The process began with extracting data from current ICU patients based on the information available from the ICU boxes. Each box provided specific data about the bed's status—whether it was active or not—and the patient occupying it, with each box assigned to a unique patient.

After identifying the current patient cohort, an ETL process was implemented for each piece of data needed by the models. It's important to note that since the models were trained on patients who had been ventilated for more than 48 hours, they are only applicable to patients meeting this criterion. The variables extracted for the pipeline were those that had been utilized in any of the models.

Similar to the preprocessing done prior to training the models, the data also required preprocessing before application. The first step in this process involved filling missing data with our own knowledge-based inputs, such as setting missing FiO2 values directly to 21, mimicking the initial data preparation approach. For other data, like laboratory results that are consistently used across all models, imputation was performed using the imputer saved during

the model creation phase. This imputer was configured to handle as many input variables as there were in the model.

After cleaning the data and ensuring no missing values, the variables were prepared and ordered in the same format as they were during the model training phase. This consistency is crucial to ensure accurate model performance. The prepared data was then applied to the three predictive models we developed—24-hour, 48-hour, and 72-hour models. So, at the end we had three probabilities: the likelihood of having VAP on the same day, within the next 24 hours, and within the next 48 hours.

It is necessary to emphasize the importance of this pipeline for validating the models, as by feeding patient data throughout their ICU stay, we can observe the model's performance and behaviour at different stages of the stay.

3.6 Near Real Time Infection Dashboard

The Near Real-Time Infection Dashboard has been designed as a decision-support tool for physicians. In this application, doctors can monitor variables related to a potential patient infection in near real-time. The term "Near Real-Time" refers to a minor delay of minutes from data generation, through storage on the server, to its display on the dashboard, which is very close to real-time. The dashboard is designed in such a way that physicians can quickly assess a patient's status regarding potential infections immediately.

3.6.1 Django Framework

Django is a high-level Python web framework that was used for developing the ICU dashboard application, primarily because of its robust architecture and efficiency in managing data-driven applications. Django is designed to facilitate rapid development and clean, pragmatic design, which is crucial for building complex web applications like the ICU monitoring system.

Django's structure is based on the "Don't Repeat Yourself" (DRY) principle, which aims to minimize code redundancy and promote code reusability. This aspect of Django is particularly beneficial in a healthcare setting where the application needs to be both reliable and easy to maintain. Furthermore, Django's robustness comes from its extensive libraries that help in building secure sites, an essential feature for any application handling sensitive health data. Its built-in security features protect against many common security threats, making it a safe framework for developing web applications that comply with data protection regulations.

Django's scalability is very robust. It can handle significant volumes of traffic and interactions effortlessly, which is critical in a hospital setting where the data transfer can be huge. Django's versatility also means it can be adapted for various web applications, from lightweight microservices to large-scale web services.

Regarding data models, Django models were used to define the structure of the ICU application's data. These models are Python classes that define the database structure transparently. Django's ORM (Object-Relational Mapping) facilitates database transactions without manual SQL coding, enhancing development speed and reducing errors.

3.6.2 Architecture and Data Flow

We designed an architecture that significantly enhanced the speed of data loading for our dashboard. Previously, all dashboards were configured to load data directly from the Centricity Critical Care (CCC) server as shown in Figure 4 (a). This setup meant that all data queries were executed directly on the server, aiming to achieve near real-time data retrieval. While this approach may be sufficient for dashboards handling minimal data, it proved inadequate for our infection tracking dashboard due to the volume of data involved.

To address this, we incorporated an intermediate step using Django models. The strategy was to automatically populate this intermediate database every 10 minutes with data from the CCC. This design ensured that the most outdated information was at most 10 minutes old. Once the data was loaded into this database, and a user accessed the dashboard, the query was executed against this new intermediate database, resulting in significantly faster load times. The data flow for this process is illustrated in figure 4 (b). This new architecture not only improves performance but also enhances the user experience by providing timely and efficient access to critical data and effective decision-making process in the ICU.

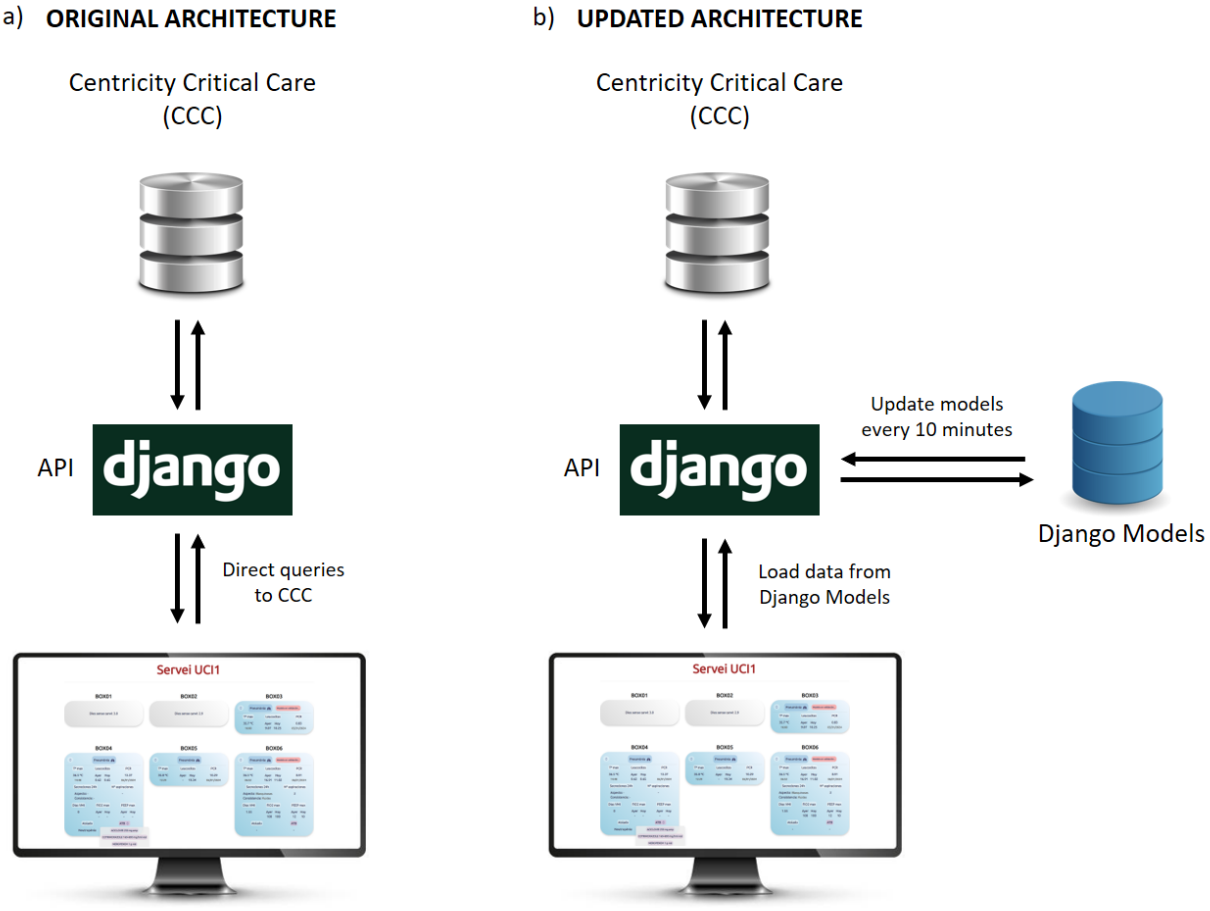

Figure 4. Novel model based intermediate architecture and the application data flow.

To facilitate this update, a service was created in the Docker Compose setup that executes a Python function named update_maps.py. This function runs every 10 minutes, ensuring that the ICU map tables are regularly updated. This regular updating process maintains the freshness and accuracy of the data, which is crucial for critical analysis and decision-making in the ICU.

To evaluate the improved loading speed of the page with the new architecture, we conducted several load tests. Specifically, the loading speed was measured 30 times for both the old and new architectures.

3.6.3 Dashboard Design and Development

The first step to design the dashboard was to select the variables to be displayed. This selection process was done with the help of an intensivist who had prior knowledge of variables directly related to infections. The chosen variables were:

- Maximum temperature of the current day and the time it was recorded.
- Median leukocyte counts from previous and current day.
- The latest PCR result and its corresponding date.
- The appearance and consistency of the patient's lung secretions.
- The number of days the patient has been mechanically ventilated.
- The maximum FiO2 set for the patient yesterday and today.
- The maximum PEEP from yesterday and today.
- Whether the patient is isolated and the reason for isolation.
- A list of antibiotics administered to the patient, if any.

The development of the dashboard was done using HTML and JavaScript. HTML was used to structure the content of the web pages, ensuring that the layout and elements were correctly placed and easily accessible. JavaScript was employed to add dynamic and interactive features to the dashboard, allowing real-time data updates and user interactions.

To enhance the functionality and visual appeal, several libraries and frameworks were utilized:

- **Bootstrap:** This CSS framework was instrumental in providing a robust foundation for the dashboard's design. Leveraging Bootstrap's extensive library of components and utilities, the dashboard gained enhanced styling, layout consistency, and ease of development. Features such as responsive grids, typography, and form controls were utilized to craft a visually appealing and user-friendly interface. Additionally, Bootstrap's extensive documentation and community support facilitated rapid prototyping and customization, contributing to the efficient development of the dashboard.

- **AJAX (Asynchronous JavaScript and XML):** AJAX was instrumental in enabling dynamic updates without requiring a full page reload. This technique facilitated the fetching of near real-time data from the server and its seamless integration into the dashboard. By asynchronously loading data, users experience faster interactions and smoother navigation within the dashboard interface, improving overall usability.

Given the large number of variables, the dashboards were designed so that initially, only the maximum temperature, leukocytes, and PCR were visible to the doctor. This approach was taken to avoid mess and make the dashboard more user-friendly and visually appealing. If the doctor wants to view the remaining variables, a button is available to expand the box and display all the variables.

At the top of the box, a central button was placed that, when clicked, returns the probability of the patient having VAP. This probability, as mentioned previously, will not be visible to doctors until the model has been validated. If the patient has already had a VAP, a red lung symbol is shown with the date when the VAP occurred.

3.6.4 Model Implementation in Dashboards

The best models for 24h, 48h and 72h time-windows were saved locally in files that included the training weights, allowing them to be directly used to predict the probability of VAP. When the pipeline explained in section 3.6 data was applied to the predictive models, it returned three probabilities. These probabilities, like the dashboard data, were also stored in the intermediate tables. Thus, at the end of the process, the tables contained both the variables for the dashboard mentioned in section 3.6.3 and the three probabilities for each of the patients. These tables were updated every 10 minutes, as previously explained, ensuring that both the dashboard variables and the model predictions were kept current.

3.6.5 Deployment

Until now, everything had been developed locally. To make this application accessible to other users, it needs to be uploaded to a server. This is where Docker proves to be crucial. Docker containers ensure that the application runs consistently by maintaining the same environment, including the versions of all tools and dependencies used during development.

To deploy an application, the process begins on a test server. The easiest way to manage this is by using a version control system like Git, where you can track the project's version history and control modifications. Access to the server is achieved via SSH. By cloning the project from Git and having the necessary files such as connection drivers, a requirements file for library versions, and a .env file containing information like IPs and user passwords, the project can run effectively. With all the files in place, running Docker Compose creates the container with the complete application.

Once the application has been verified to work correctly on the test server, it can be deployed on the production server, where users will access the application. If new bugs are discovered, the process involves returning to the local project, resolving the errors, updating the project in Git, recreating the project on the test server, and verifying that no further errors

exist. Finally, the same process is performed on the production server, ensuring the application runs smoothly for end users.

4 Results and Discussion

4.1 Population Inclusion and Exclusion Criteria

The information corresponding to this section has been deleted for confidentiality reasons.

Figure 5. Inclusion and exclusion criteria pipeline of ICU patients.

4.2 Dealing With Missing Values

First, variables that not included at least 40% of the patients were removed:

- pct_24h
- pct_48h
- pct_72h
- noradrenaline_24h
- noradrenaline_48h
- noradrenaline_72h
- dobutamine_24h
- dobutamine_48h
- dobutamine_72h

The data corresponding to this section has been deleted for confidentiality reasons.

4.3 Exploratory Data Analysis (EDA)

4.3.1 VAP distribution

The information corresponding to this section has been deleted for confidentiality reasons.

Figure 6. Distribution of VAP. The pie chart illustrates the VAP distribution showcasing a significant imbalance.

4.3.2 Laboratory Results

We wanted to examine the differences between the two groups in terms of laboratory results for leukocytes, lymphocytes, and polymerase chain reaction (PCR) across the three time periods: 24 hours, 48 hours, and 72 hours. To do this, we created a line graph which

allowed me to visualize the trends in the results for both groups. This graphical representation was crucial for identifying any significant changes or patterns in the laboratory markers over time, providing insight into the physiological responses associated with ventilator-associated pneumonia.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 7. Median leukocyte count over the three-time windows. In orange trend of VAP group.

In the analysis of leukocyte counts across two groups—those with VAP and those non-VAP—the data revealed distinctive trends over the periods of 24, 48, and 72 hours. The group with VAP showed a marked increase in leukocyte count from 24 to 72 hours, suggesting an intensifying inflammatory response as the body attempts to fight the infection. This rise in leukocyte levels is indicative of the immune system's mobilization to combat the lung infection associated with mechanical ventilation. In contrast, the leukocyte counts in the non-VAP group remained relatively stable, slightly decreasing over time, which aligns with the absence of infection.

These trends not only highlight the potential of leukocyte counts as biomarkers for detecting and monitoring VAP in patients undergoing mechanical ventilation but also underscore their utility in assessing the severity and progression of the infection. The stability of leukocyte counts in patients without VAP serves as a baseline against which the immune response in VAP patients can be measured. The increasing leukocyte count in the VAP group could prompt clinicians to consider more aggressive or targeted treatment options to manage the infection effectively. This analysis forms a critical part of understanding the immune dynamics in patients with VAP, potentially guiding clinical decisions regarding their management and treatment.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 8. Median lymphocyte count over the three-time windows.

In lymphocyte counts analysis across VAP and non-VAP, the trends over the intervals of 24, 48, and 72 hours were notably different. The group diagnosed with VAP displayed a consistent decline in lymphocyte counts over the observed time periods, suggesting a possible immunosuppression or a high physiological stress response to infection, which is common in severe infections like pneumonia. In contrast, the lymphocyte counts in the non-VAP group remained relatively stable across all time points.

This stability in lymphocyte levels among the non-VAP patients likely reflects the absence of severe infection, whereas the decreasing trend in the VAP group underscores the impact of the infection on the immune system's lymphocyte response. Despite the incremental differences between the time points not being substantial, the overall difference in the medians between the two groups is quite pronounced. This observation supports the hypothesis that lymphopenia (a reduction in lymphocyte count) could be a marker of infection severity in patients with VAP, potentially aiding in the clinical assessment and monitoring of these patients. Such findings are crucial for understanding the immune dynamics in mechanically ventilated patients, offering insights that could influence treatment strategies and clinical outcomes.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 9. Median PCR over the three-time windows.

We observed distinct trends for PCR analysis between the two groups. For VAP patients, the PCR levels show a significant upward trend over time, indicating a continuous inflammatory response as the body reacts to the infection associated with mechanical ventilation. This progression reflects the typical response to an infection, where PCR levels increase as the body's immune system reacts to bacteria or other pathogens.

Conversely, the PCR levels in non-VAP patients remained consistently low across all time points, suggesting the absence of an acute inflammatory response. This stability indicates that these patients likely did not develop new infections or inflammatory conditions during the observed period.

The pronounced difference in the trajectories of CRP levels between the two groups underscores the utility of PCR as a potential biomarker for the presence and severity of VAP. The significant rise in PCR levels in the VAP group could provide clinicians with valuable information for diagnosing and managing infections in mechanically ventilated patients, potentially guiding therapeutic decisions and monitoring treatment efficacy.

4.3.3 Body Temperature (maximum)

The box plot visualizes the maximum temperature measurements across three-time intervals—24, 48, and 72 hours—for VAP and non-VAP groups. From the graph, it is evident that the maximum temperature is consistently higher in the VAP group across all time points compared to the non-VAP group.

In the 24-hour interval, temperatures in the VAP group are notably higher, indicating a feverish response, which is a common symptom of infection, as it is explained in the literature review section. This trend continues into the 48-hour and 72-hour marks, with the VAP group maintaining higher temperatures throughout. The relatively stable and lower temperatures in the non-VAP group suggest the absence of infection-related fever.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 10. Boxplot of maximum temperature values across different time windows (24h, 48h, 72h) relative to VAP status.

4.3.4 FiO2 (maximum)

From the box plots for maximum FiO2, it is evident that the levels are consistently higher in the VAP group (Group 1) across all time intervals compared to the non-VAP group (Group 0). This suggests an increased need for supplemental oxygen in patients with VAP, likely due to compromised lung function and increased oxygen requirements due to infection.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 11. Boxplot of maximum FiO2 values across different time windows (24h, 48h, 72h) relative to VAP status.

4.3.5 SpO2/FiO2 (median)

The median SpO2/FiO2 values are higher in the VAP group across all three-time intervals, indicating potentially less efficient gas exchange or greater respiratory distress in these patients. This metric, which considers both the FiO2 and Peripheral Oxygen Saturation (SpO2), reflects the challenges in maintaining optimal oxygenation in VAP patients.

These findings underscore the clinical complexities associated with managing VAP, particularly in adjusting ventilatory support to address the altered pulmonary physiology. The higher FiO2 requirements and altered SpO2/FiO2 values in VAP patients could serve as important markers for VAP detection.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 12. Boxplot of SpO2/FiO2 median values across different time windows (24h, 48h, 72h) relative to VAP status.

4.3.6 Simple Correlation with VAP

The graph in Figure 13 illustrates simple correlations associated with VAP. Variables that exhibit a positive correlation are linearly and positively associated with VAP, that is, as these variables increase, so does the likelihood of VAP. Conversely, negatively correlated variables indicate that as these variables increase, the likelihood of VAP decreases. A notable observation is that, in this time window, mechanical ventilation (MV) is not directly associated with VAP. However, this does not imply that MV is unrelated to VAP, as variables can be related in a non-linear manner, which is further explored in the models.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 13. Simple correlation of 24h time window variables with VAP.

4.3.7 Binary (0, 1) and Categorical Variables

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 14. Number of observations of binary variables within VAP group.

The bar chart illustrates the prevalence of various comorbid conditions in patients with and without VAP. It is clear from the graph that conditions such as arterial hypertension, metabolic disorders, and obesity are more frequently observed in the VAP group, suggesting a potential correlation between these comorbidities and the incidence of VAP.

This visual representation aids in understanding the complex relationship between chronic conditions and susceptibility to VAP in hospitalized patients. For example, the higher instances of arterial hypertension and metabolic disorders in the VAP group might indicate that these patients have a predisposition to develop complications due to their pre-existing health conditions, which may compromise their immune system or respiratory function, thus making them more vulnerable to infections like VAP.

It is important to note, for instance, that while arterial hypertension appears more frequently in patients with VAP, it is also a common condition in patients without VAP. This suggests that while arterial hypertension may seem like a potential predictive variable due to its higher occurrence in VAP cases, its overall prevalence across all patients must be considered. This understanding is crucial for clinicians in identifying high-risk patients and potentially guiding preventive strategies to reduce the incidence of VAP in patients with significant comorbidities.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 15. Number of observations of dummy categorical variables within VAP group.

In this case the bar chart displays the frequency of categorical variables across patients with and without VAP. We observe higher instances of certain variables such as ”aspecto_secreciones_24h_purulentas” (purulent appearance of secretions at 24 hours) and ”aspecto_secreciones_48h_purulentas” (at 48 hours) in patients with VAP compared to those without. This could indicate that purulent secretions are more common in patients suffering from VAP, which aligns with the infection profiles typically seen in pneumonia.

Variables like ”AdmType_urgente” (urgent admission type) also show higher frequencies in the VAP group, suggesting that patients who are urgently admitted are more likely to develop VAP. This could be reflective of the severity at admission or delays in receiving appropriate care.

Conversely, variables such as ”aspecto_secreciones_24h_no_purulentas” (non-purulent appearance of secretions at 24 hours) are more common in the non-VAP group, pointing to a less severe infection state or better initial pulmonary health.

Given the abundance of categorical and binary variables in the dataset, we decided to conduct chi-square tests to streamline the feature selection process for predicting VAP. Among all the categorical and binary variables, we retained those which had a direct statistical significance with VAP group:

Categorical variables:

Table 3. Summary of Significant Categorical Variables with p-values and Degrees of Freedom

Variable Name	p-value	Degrees of Freedom (DOM)

The information corresponding to this table has been deleted for confidentiality reasons.

Binary variables (0, 1):

Table 4. Summary of Significant Binary Variables with p-values and Degrees of Freedom

Variable Name	p-value	Degrees of Freedom (DOM)

The information corresponding to this table has been deleted for confidentiality reasons.

4.4 Model Training

The dataset was divided into three subsets: training, test, and validation sets. Each subset was carefully constructed to maintain the distribution of the variable of interest (VAP) across all sets. For LR, a standard split of 80% for training and 20% for testing was applied. It's worth noting that the 20% allocated for testing in LR essentially combines the validation and test sets used for other models. Here are the details of each subset:

- **Training Set:** Comprised 79.98% of the data.
- **Test Set:** Accounted for 10.01% of the data.
- **Validation Set:** Also represented 10.01% of the data.

The values of the hyperparameter alpha ranged between 0.6 and 0.8, inclusive, during the model training process. This approach ensured that the model was trained, tested, and validated on representative and balanced subsets of the dataset.

4.5 Selected Model Variables

For the models developed at different time intervals (24h, 48h, and 72h), different sets of variables were utilized, reflecting how pneumonia characteristics can change significantly depending on the day.

4.5.1 24h Time Window Models

- Random Forest (RF):

  The information corresponding to this figure has been deleted for confidentiality reasons.

- Logistic Regression (LR), Linear SVM, and Non-Linear SVM:

  The information corresponding to this figure has been deleted for confidentiality reasons.

4.5.2 48h Time Window Models

- For all models (RF, LR, Linear SVM, and Non-Linear SVM), the same variables were used:

  The information corresponding to this figure has been deleted for confidentiality reasons.

4.5.3 72h Time Window Models

- Random Forest and Non-Linear SVM:

  The information corresponding to this figure has been deleted for confidentiality reasons.

- Logistic Regression (LR) and Linear SVM:

  The information corresponding to this figure has been deleted for confidentiality reasons.

These selected variables reflect the most relevant clinical parameters at each respective time point, tailored to optimize the predictive power of each model. The slight differences in the variables used for the different models and time intervals were based on their specific characteristics and the nature of the predictions required at each stage.

4.6 Model Results

The results of our four models (RF, LR, Linear SVM, and Non-Linear SVM) can be seen in Figure 16 and 17. The evaluation metrics considered were Accuracy (to assess overall model performance), Recall (to evaluate the model's ability to predict the positive class), Confusion Matrix (to provide a detailed view of true positives, false positives, true negatives, and false negatives), and AUC (Area Under the Curve, which measures the model's ability to distinguish between classes). It is important to note that for LR, no hyperparameter tuning was conducted; therefore, only train and test sets were used without a validation set. This is why the population in the training and test sets for LR was larger than for the other models.

For the 24-hour period, the RF model achieved an accuracy of 0.85 and a recall of 0.86, with an AUC of 0.87. Its confusion matrix indicated a strong balance between true positives and true negatives, showcasing its ability to accurately classify instances. LR, with an accuracy

of 0.83 and a recall of 0.73, presented a slightly lower performance. The Linear SVM matched the RF in accuracy (0.85) but had a lower recall (0.71) and the Non-Linear SVM outperformed the accuracy of all models with a score of 0.86 and had a recall of 0.79, coupled with an AUC of 0.90.

At the 48-hour mark, RF maintained a good performance with an accuracy of 0.79 and a recall of 0.79, indicating consistent performance over time. LR saw a slight drop in accuracy to 0.76 and a recall of 0.68, suggesting some degradation in its predictive ability. Linear SVM's accuracy decreased to 0.73, with a recall of 0.68, while the Non-Linear SVM remained robust with an accuracy of 0.85 but exhibited a notable drop in recall to 0.46, reflecting potential issues with model overfitting or generalization over time.

For the 72-hour period, RF showed an accuracy of 0.73 and maintained a recall of 0.79, indicating its robustness in longer-term predictions. LR and Linear SVM both had an accuracy of 0.73, with recalls of 0.71 and 0.68, respectively. These models displayed higher numbers of false negatives, impacting their recall rates negatively. The Non-Linear SVM's performance dropped, with an accuracy of 0.84 and a recall of 0.43, along with a lower AUC of 0.70, highlighting its reduced effectiveness over time.

Overall, the RF model demonstrated the most balanced and robust performance across all metrics and time periods. It consistently achieved high accuracy, recall, and AUC, making it the most reliable model for this dataset. The Non-Linear SVM showed high initial accuracy, but its performance degraded over time, making it less reliable for longer-term predictions. LR and Linear SVM provided moderate performance but were outperformed by RF in most aspects.

For future work, it may be beneficial to explore ensemble methods or further tune the hyperparameters of the Non-Linear SVM to improve its generalization capabilities. These findings underscore the importance of model selection and the potential need for model tuning or enhancement to maintain predictive accuracy over varying time periods.

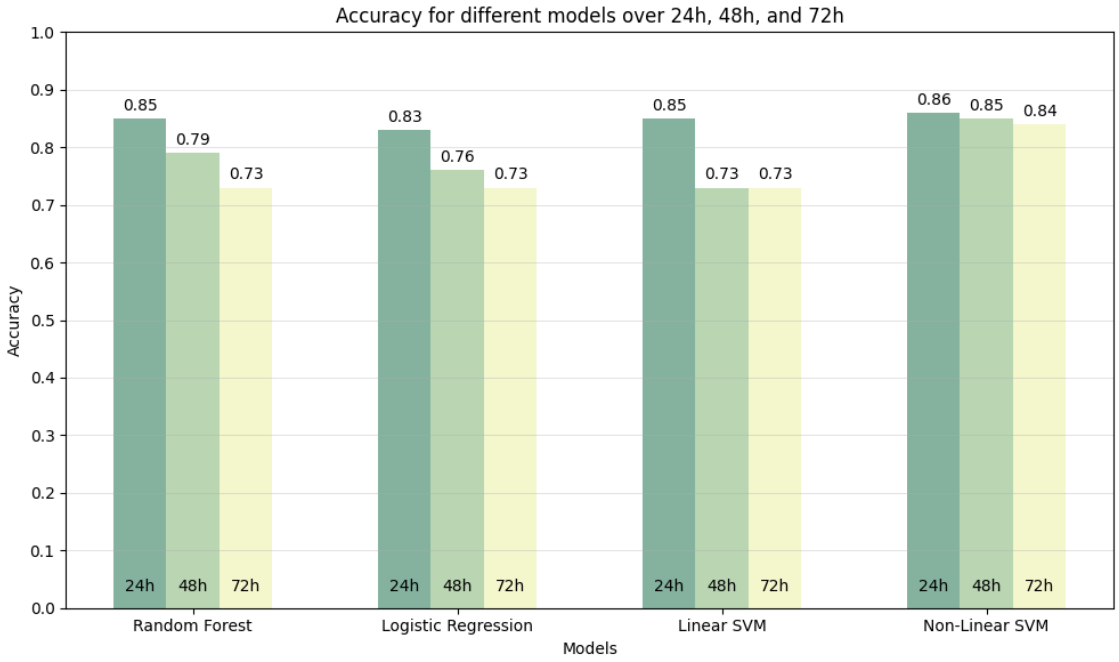

Figure 16. Accuracy results of RF, LR, Linear SVM and Non-Linear SVM for 24h, 48h and 72h time windows.

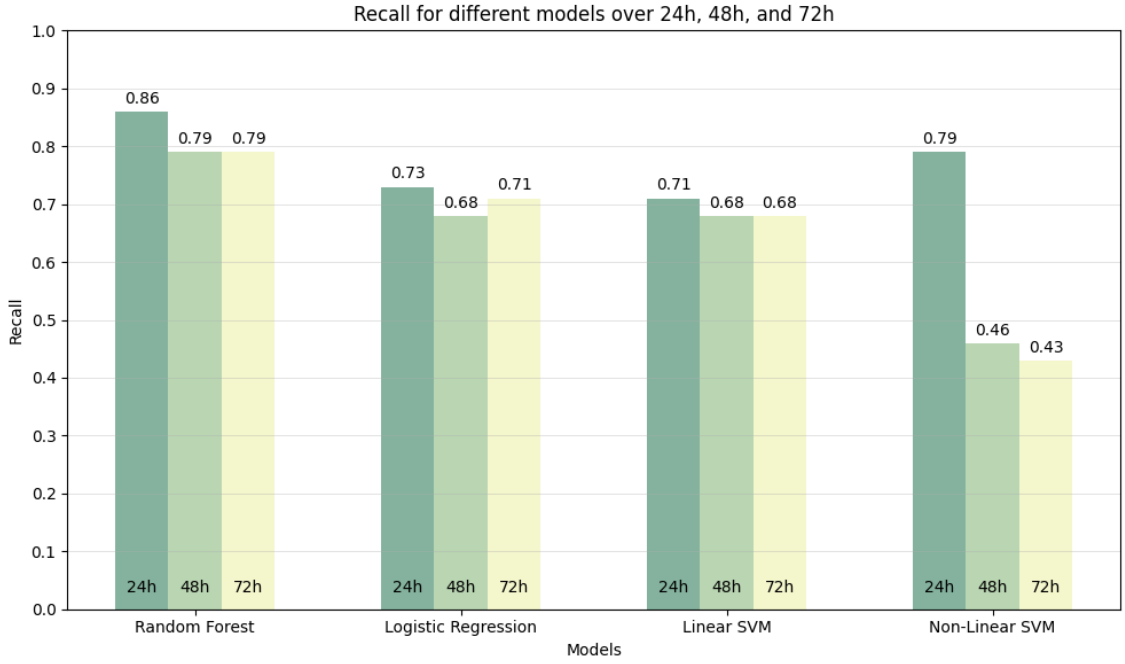

Figure 17. Recall results of RF, LR, Linear SVM and Non-Linear SVM for 24h, 48h and 72h time windows.

Table 5. Accuracy, Recall and AUC scores and Confusion Matrix results of RF, LR, Linear SVM and Non-Linear SVM for 24h, 48h and 72h time windows.

Model	24h	48h	72h
Random Forest	Accuracy: 0.85	Accuracy: 0.79	Accuracy: 0.73
	Recall: 0.86	Recall: 0.79	Recall: 0.79
	Confusion matrix: $\begin{bmatrix} 24 & 4 \\ 34 & 186 \end{bmatrix}$	Confusion matrix: $\begin{bmatrix} 22 & 6 \\ 47 & 173 \end{bmatrix}$	Confusion matrix: $\begin{bmatrix} 22 & 6 \\ 61 & 159 \end{bmatrix}$
	AUC: 0.87	AUC: 0.80	AUC: 0.79
Logistic Regression	Accuracy: 0.83	Accuracy: 0.76	Accuracy: 0.73
	Recall: 0.73	Recall: 0.68	Recall: 0.71
	Confusion matrix: $\begin{bmatrix} 41 & 15 \\ 70 & 370 \end{bmatrix}$	Confusion matrix: $\begin{bmatrix} 38 & 18 \\ 103 & 337 \end{bmatrix}$	Confusion matrix: $\begin{bmatrix} 40 & 16 \\ 116 & 324 \end{bmatrix}$
	AUC: 0.88	AUC: 0.76	AUC: 0.78
Linear SVM	Accuracy: 0.85	Accuracy: 0.73	Accuracy: 0.73
	Recall: 0.71	Recall: 0.68	Recall: 0.68
	Confusion matrix: $\begin{bmatrix} 20 & 8 \\ 28 & 192 \end{bmatrix}$	Confusion matrix: $\begin{bmatrix} 19 & 9 \\ 58 & 162 \end{bmatrix}$	Confusion matrix: $\begin{bmatrix} 19 & 9 \\ 59 & 161 \end{bmatrix}$
	AUC: 0.89	AUC: 0.75	AUC: 0.72
Non-Linear SVM	Accuracy: 0.86	Accuracy: 0.85	Accuracy: 0.84
	Recall: 0.79	Recall: 0.46	Recall: 0.43
	Confusion matrix: $\begin{bmatrix} 22 & 6 \\ 28 & 192 \end{bmatrix}$	Confusion matrix: $\begin{bmatrix} 13 & 15 \\ 22 & 198 \end{bmatrix}$	Confusion matrix: $\begin{bmatrix} 12 & 16 \\ 23 & 197 \end{bmatrix}$
	AUC: 0.90	AUC: 0.69	AUC: 0.70

The evaluation of the four models offers valuable insights into their predictive performance and stability over time. The distinct performance metrics and variable selections highlight the nuances and challenges in predicting pneumonia characteristics based on the time elapsed since the onset.

4.6.1 Random Forest

The RF model demonstrated consistent and robust performance across all time periods. Its gradual decline in accuracy as the prediction window increases is expected, yet the model's recall rates remained relatively high. This indicates the model's strong capability to identify

true positive cases of pneumonia over time. The consistent performance can be attributed to RF's ability to handle complex interactions among variables, making it a reliable choice for this type of medical prediction.

The variable that stands out in the RF models across all three-time intervals is PCR results, consistently showing the highest importance. Lymphocytes and days on mechanical ventilation are also important variables that appear in all three models. The remaining variables vary based on the time window under consideration.

Notably, the binary variables, which are the last two variables in each graph, exhibit very low importance. This highlights the role of prior knowledge, as these variables would have been automatically discarded by the RF if not manually included due to their low importance. Temperature, for instance, does not appear in the 72-hour model, likely because it is uncommon for patients to have fever spikes many hours before VAP onset. Similarly, leukocytes are not important in the 72-hour window, as the EDA indicated no significant differences between positive and negative classes.

Purulent secretions, associated with VAP, do not become significant until closer to the VAP onset, specifically in the 24–48-hour windows. As we approach the onset of VAP, the number of days the patient has been ventilated and the presence of a tracheostomy become less important. Instead, laboratory results, signs of fever, and the need for increased oxygen percentage through FiO2 become more relevant.

This analysis underscores the dynamic nature of pneumonia characteristics and the importance of contextual variable selection based on the time frame for prediction. Understanding these temporal changes is crucial for improving the predictive power and clinical relevance of the models.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 18. Feature importances of best variables in 24h time window.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 19. Feature importances of best variables in 48h time window.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 20. Feature importances of best variables in 72h time window.

4.6.2 Logistic Regression

LR, while simpler than the other models, performed adequately but exhibited some limitations. The model's performance dropped slightly over time, indicating potential issues with capturing the complexity of the data over longer periods. The absence of hyperparameter tuning might have also contributed to these results. Nevertheless, LR's relatively stable recall rates suggest it can still reliably identify positive cases, but less effectively than RF. Its simplicity and interpretability, however, make it a valuable model in clinical settings where understanding the decision process is crucial.

The linear equations and the corresponding Odds Ratios of the coefficients were calculated for each time period to understand the variable importance for each of the windows (24h ,48h and 72h).

The information corresponding to this equation has been deleted for confidentiality reasons.

The OR values for each variable are as follows: *The information corresponding to the OR has been deleted for confidentiality reasons.*

When variables are binary, such as the aspect of purulent secretions (e.g., present or absent), the odds ratio (OR) of 1.1843 indicates that the odds of the event of interest (e.g., developing VAP) are approximately 1.1843 times higher in patients with purulent secretions compared to those without, all other factors being equal.

In the case of continuous variables, such as maximum temperature in the last 24 hours, the odds ratio (OR) of 1.2366 indicates that for every one-degree Celsius increase in maximum temperature, the odds of the event of interest (e.g., developing VAP) increase by approximately 23.66%, all other factors being equal.

The information corresponding to this equation has been deleted for confidentiality reasons.

The information corresponding to the OR has been deleted for confidentiality reasons.

The most notable variables for predicting VAP within the 48-hour window, which essentially equates to predicting VAP with a 24-hour lead time, appear to be the aspect of secretions, whether the patient has undergone tracheostomy during this period, and the median temperature of the patient.

The information corresponding to this equation has been deleted for confidentiality reasons.

The information corresponding to the OR has been deleted for confidentiality reasons.

During this time-period, the SpO2/FiO2 ratio played an important role due to its significant protective factor. Lymphocytes also acted as a protective factor to consider, while the number of aspirations the patient had within this window of hours was a risk factor.

In general, it can be observed that as we move further away from the VAP event, the OR values for the variables tend to approach 1. This indicates that the differences become smaller, which is expected since the further we are from the event, the harder it is to predict.

4.6.3 Support Vector Machine

The Linear SVM showed competitive initial performance but faced a notable decline over time. This highlights that while Linear SVM can handle linear separability in the data, it struggles with the increased complexity and non-linear relationships that might emerge over longer prediction windows. The consistent drop across metrics suggests that additional feature engineering or the inclusion of non-linear kernels could potentially improve its performance.

The Non-Linear SVM initially outperformed other models in accuracy but showed significant performance degradation over time. The sharp drop in recall indicates issues with generalization and potential overfitting. While the initial high accuracy suggests strong performance on the training set, the subsequent decline points to difficulties in maintaining predictive accuracy as the prediction horizon extends. This behaviour underscores the importance of model validation and the need for regularization techniques to enhance the generalizability of Non-Linear SVMs.

4.6.4 Model Comparison

Comparing the models, the RF consistently emerged as the most reliable across all time periods, balancing accuracy, recall, and AUC effectively. Its performance stability makes it a strong candidate for real-world applications where consistent prediction over time is crucial. The Non-Linear SVM, despite its initial high accuracy, requires further tuning and regularization to improve its long-term predictive capabilities. LR and Linear SVM, while simpler, demonstrated moderate performance, indicating their potential utility in scenarios where interpretability and simplicity are prioritized.

4.7 Dashboard

The loading time measurements (mean and standard deviation) results are illustrated in Figure 21. The old architecture loading time mean was around 13 seconds, while new architecture obtained loading times around 0.36 seconds. From these results, it is evident that the new architecture is significantly faster than the old one, demonstrating a nearly 97.3% increase in speed.

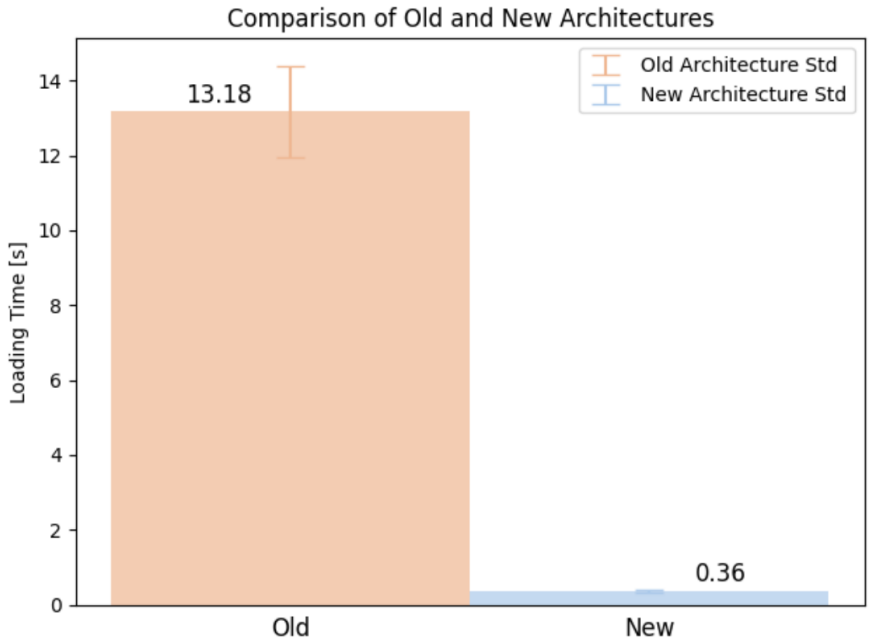

Figure 21. Loading speed of old architecture vs new architecture.

The results of the dashboard can be viewed in Figures 22 and Figure 23. Specifically, in Figure 22, we observe the dynamic aspect of the dashboard. Image "a" in this figure showcases the most crucial patient data in a condensed format. By clicking on the dropdown menu in the top-left corner, the same dashboard provides further patient information, allowing for a more detailed view (image "b"). Additionally, there is also a dropdown menu for antibiotics, which, if applicable, provides a list of antibiotics currently administered to the patient (image "c").

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 22. Dynamic functionality of a single box dashboard. The orange squares point out interactive map buttons, which when pressed expand or contract new information.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 23. General overview infection dashboard for six ICU boxes.

By designing the dashboard in this way, we ensured that it remained functional and efficient, allowing doctors to quickly access the most pertinent information without being overwhelmed by data. This design consideration is crucial in a high-pressure ICU environment where quick and accurate decision-making is vital.

4.8 Further Implementations

Cross-model validation was conducted, where all models were tested across different time windows. However, no improvement in performance was observed compared to models trained specifically for their respective time windows.

Additionally, differential variables, representing the difference in values between two temporal points, were introduced to capture trends. Nevertheless, this approach did not yield superior results.

5 Limitations

- **Data Collection Method for VAP:**
    - The VAP diagnosis is recorded manually. Often, VAP is diagnosed but not immediately documented in the server. This means that many cases of VAP are recorded much later than when they actually occurred, with delays ranging from hours to days. Such delays significantly impact the predictive accuracy of the models.
- **Scope and Complexity of Variables:**
    - The large number of variables considered for the model, including their maximums, minimums, and other derived metrics, posed a substantial challenge. It is impossible to explore all possible combinations comprehensively, which made it difficult to leverage each algorithm to its fullest potential. This complexity could lead to suboptimal model performance due to the underutilization of potentially informative variables.
- **Imbalanced Dataset:**
    - The dataset is significantly imbalanced, with a much higher number of patients not having VAP compared to those who do. To address this, many non-pneumonia cases were removed from training set, which greatly reduced the population size. This reduction can lead to loss of valuable information and affect the model's performance.
- **Missing Laboratory Variables:**
    - Laboratory variables, which are likely among the most important predictors, had a high rate of missing values. Many of these missing values had to be imputed, which could introduce biases and reduce the accuracy of the models. Increasing the frequency and completeness of laboratory data collection would likely enhance the model's predictive power and reliability.

6 Conclusions

6.1 To Develop a Predictive Model

- The robustness of the RF model across different time windows not only confirms its reliability but also establishes it as the optimal choice for predicting Ventilator-Associated Pneumonia (VAP) in our study.
- Within the 24-hour prediction window, the RF model demonstrated high performance, correctly classifying 86% of cases overall and 87% of positive VAP cases. As the prediction window extended to 48 hours, performance slightly decreased but remained effective, with the model maintaining a 79% accuracy rate for both VAP and non-VAP cases.
- The ability of the RF model to provide accurate predictions 24 hours in advance has significant clinical implications. This period may allow healthcare providers to take preventive measures, potentially reducing the incidence and severity of VAP.
- The model's effectiveness is facilitated by its use of a minimal set of nine variables. While most of these variables are universally gathered across ICUs, the number of patient aspirations and the appearance of secretions are exceptions. Specifically, the 24-hour and 48-hour models include the appearance of secretions as a variable, whereas the 72-hour model incorporates the number of aspirations. The remaining variables are applicable to other ICUs.

6.2 To create Real-time Prediction Pipeline

- The development of a real-time prediction pipeline has significantly advanced the efficiency of data handling, enabling the immediate application of predictive models in a clinical setting.
- The incorporation of date and time stamping ensures that the predictions are both timely and relevant, laying a robust groundwork for future model validations.
- The implementation of this pipeline has effectively reduced dashboard loading times, crucial for enhancing user experience within clinical environments.
- The adaptability of the pipeline supports future enhancements and updates, facilitating continuous improvement without disrupting existing workflows.

6.3 To design an Infection-related Frontend Dashboard Application

- The dashboard is designed to be near real-time, dynamic, and user-friendly, providing healthcare professionals with essential information rapidly and effectively.
- The seamless integration of the real-time prediction pipeline with the dashboard ensures efficient data display, crucial for timely clinical decision-making.

- By displaying both current and historical patient data, the dashboard aids clinicians in monitoring patient conditions and identifying infection trends, which could indicate the onset of an infection.
- This dashboard tool is instrumental in proactive patient management, enabling clinicians to quickly assess and respond to potential VAP risks.

7 Future Work

- Although ANN models were tested, the evaluation was not exhaustive. Initial results were promising (24h: Accuracy: 0.82, Recall: 0.89; 48h: Accuracy: 0.73, Recall: 0.75; 72h: Accuracy: 0.74, Recall: 0.64), but these models were based on the features used in the RF. ANN models might perform better with different feature combinations or more extensive tuning. Further exploration and optimization of ANN could potentially improve predictive performance.
- A different approach for predicting VAP could involve using Long Short-Term Memory (LSTM) architectures. By introducing variables at different time points in an ordered manner, it would be possible to model variable trends over time, which would likely enhance VAP prediction accuracy.
- We should emphasize the importance of continuous model validation and updating to ensure accuracy and relevance over time. Regular updates and validations are essential to adapt to changing clinical environments and patient populations.
- Given the novelty of this method, direct comparisons with other studies are limited. Future research should explore various combinations of variables and algorithms, benchmarking against existing predictive models to validate findings and enhance generalizability.
- Model validation is crucial for real-world application. One approach is to apply the models to patients who were not included in the model development or to newly admitted patients. For instance, applying the model to these patients every 6 hours to obtain VAP probability estimates for the next 24, 48, and 72 hours. This would help to reveal the models' behaviours and fluctuations, demonstrating their real-world predictive capabilities.
- Once the model is validated, it would be beneficial to test it in different centers. This would address generalizability, requiring other ICUs to collect variables such as the number of aspirations and secretion appearance, ensuring the model's applicability across various clinical settings. This would add more credibility, generability and robustness to the models.
- To determine the model's utility, it is essential to test its application in a real-world clinical setting. In our ICU, there are three critical care units: ICU 1 (12 beds), ICU 2 (6 beds), and ICU 3 (6 beds). A practical approach could involve:

8 References

[1] T. Craven, G. Wojcik, J. McCoubrey, *et al.*, "Ventilator-associated pneumonia surveillance using two methods," *Journal of Hospital Infection*, vol. 104, no. 4, pp. 522–528, 2020, ISSN: 0195-6701. DOI: https://doi.org/10.1016/j.jhin.2020.01.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S019567012030044X.

[2] J. Hunter, "Ventilator associated pneumonia. bmj 344(e3325):e3325," *BMJ (Clinical research ed.)*, vol. 344, e3325, May 2012. DOI: 10.1136/bmj.e3325.

[3] H. Keyt, P. Faverio, and M. Restrepo, "Prevention of ventilator-associated pneumonia in the intensive care unit: A review of the clinically relevant recent advancements," *The Indian journal of medical research*, vol. 139, pp. 814–21, Jun. 2014.

[4] R. J. Keneally, T. J. Peterson, J. R. Benjamin, K. Hawkins, and D. Davison, "Making ventilator associated pneumonia rate a meaningful quality marker," *Journal of Intensive Care Medicine*, vol. 36, no. 11, pp. 1354–1360, 2021, PMID: 32885716. DOI: 10.1177/0885066620952763.

[5] B. Al-Omari, P. Mcmeekin, A. Allen, *et al.*, "Systematic review of studies investigating ventilator associated pneumonia diagnostics in intensive care," *BMC Pulmonary Medicine*, vol. 21, Jun. 2021. DOI: 10.1186/s12890-021-01560-0.

[6] A. Chouhdari, S. Shokouhi, F. Bashar, *et al.*, "Is a low incidence rate of ventilation associated pneumonia associated with lower mortality? a descriptive longitudinal study in iran," *Tanaffos*, vol. 17, pp. 110–116, Feb. 2018.

[7] N. Manga, R. Oppong, E. Senanayake, *et al.*, "Cost of treating ventilator associated pneumonia post cardiac surgery in the national health service: Results from a propensity-matched cohort study.," *Journal of the Intensive Care Society*, vol. 19, Sep. 2017. DOI: 10.1177/1751143717740804.

[8] J. P. Wiener-Kronish and H. I. Dorr, "Ventilator-associated pneumonia: Problems with diagnosis and therapy," *Best Practice Research Clinical Anaesthesiology*, vol. 22, no. 3, pp. 437–449, 2008, Infectious Disease and Perioperative Infections, ISSN: 1521-6896. DOI: https://doi.org/10.1016/j.bpa.2008.05.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1521689608000347.

[9] A. Rea-Neto, N. Youssef, F. Tuche, *et al.*, "Diagnosis of ventilator-associated pneumonia: A systematic review of the literature," *Critical care (London, England)*, vol. 12, R56, Feb. 2008. DOI: 10.1186/cc6877.

[10] I. Porzecanski and D. L. Bowton, "Diagnosis and treatment of ventilator-associated pneumonia," *Chest*, vol. 130, no. 2, pp. 597–604, 2006, ISSN: 0012-3692. DOI: https://doi.org/10.1378/chest.130.2.597. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0012369215518825.

[11] S. Fernando, A. Tran, W. Cheng, *et al.*, "Diagnosis of ventilator-associated pneumonia in critically ill adult patients — a systematic review and meta-analysis," *Intensive Care Medicine*, vol. 46, pp. 1170–1179, Apr. 2020. DOI: 10.1007/s00134-020-06036-z.

[12] K. Kawamoto, C. Houlihan, A. Balas, and D. Lobach, "Improving clinical practice using clinical decision support systems: A systematic review of trials to identify features critical to success," *BMJ (Clinical research ed.)*, vol. 330, p. 765, Apr. 2005. DOI: 10.1136/bmj.38398.500764.8F.

[13] J. Osheroff, J. Teich, D. Levick, *et al.*, *Improving Outcomes with Clinical Decision Support: An Implementer's Guide, Second Edition*. Feb. 2012, ISBN: 9781498757461. DOI: 10.4324/9781498757461.

[14] D. Bates, M. Cohen, L. Leape, J. M. Overhage, M. Shabot, and T. Sheridan, "Reducing the frequency of errors in medicine using information technology," *Journal of the American Medical Informatics Association : JAMIA*, vol. 8, pp. 299–308, Jul. 2001. DOI: 10.1136/jamia.2001.0080299.

[15] Z. Obermeyer and E. Emanuel, "Predicting the future — big data, machine learning, and clinical medicine," *The New England journal of medicine*, vol. 375, pp. 1216–1219, Sep. 2016. DOI: 10.1056/NEJMp1606181.

[16] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, 2019. DOI: 10.1056/NEJMra1814259. eprint: https://www.nejm.org/doi/pdf/10.1056/NEJMra1814259. [Online]. Available: https://www.nejm.org/doi/full/10.1056/NEJMra1814259.

[17] S. Khozin, G. M. Blumenthal, and R. Pazdur, "Real-world Data for Clinical Evidence Generation in Oncology," *JNCI: Journal of the National Cancer Institute*, vol. 109, no. 11, djx187, Sep. 2017, ISSN: 0027-8874. DOI: 10.1093/jnci/djx187. eprint: https://academic.oup.com/jnci/article-pdf/109/11/djx187/23699015/djx187.pdf. [Online]. Available: https://doi.org/10.1093/jnci/djx187.

[18] A. Rajkomar, M. Hardt, M. Howell, G. Corrado, and M. Chin, "Ensuring fairness in machine learning to advance health equity," *Annals of Internal Medicine*, vol. 169, Dec. 2018. DOI: 10.7326/M18-1990.

[19] C. Giang, J. Calvert, K. Rahmani, *et al.*, "Predicting ventilator-associated pneumonia with machine learning," *Medicine*, vol. 100, e26246, Jun. 2021. DOI: 10.1097/MD.0000000000026246.

[20] A. Samadani, T. Wang, K. van Zon, and L. A. Celi, "Vap risk index: Early prediction and hospital phenotyping of ventilator-associated pneumonia using machine learning," *Artificial Intelligence in Medicine*, vol. 146, p. 102 715, 2023, ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2023.102715. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0933365723002294.

[21] Y. Liang, C. Zhu, C. Tian, *et al.*, "Early prediction of ventilator-associated pneumonia in critical care patients: A machine learning model," *BMC Pulmonary Medicine*, vol. 22, Jun. 2022. DOI: 10.1186/s12890-022-02031-w.

[22] V. Zaydfudim, L. A. Dossett, J. M. Starmer, *et al.*, "Implementation of a Real-time Compliance Dashboard to Help Reduce SICU Ventilator-Associated Pneumonia With the Ventilator Bundle," *Archives of Surgery*, vol. 144, no. 7, pp. 656–662, Jul. 2009, ISSN: 0004-0010. DOI: 10.1001/archsurg.2009.117. eprint: https://jamanetwork.com/journals/jamasurgery/articlepdf/405157/sws90007_656_662.pdf. [Online]. Available: https://doi.org/10.1001/archsurg.2009.117.

[23] M. Fathi, H. Moghaddasi, A. Hosseini, and M. E. Aghdam, "Developing a dashboard software for the icus and studying its impact on reducing the ventilator-associated pneumonia," *The Open Medical Informatics Journal*, vol. 12, no. 1, pp. 42–50, Oct. 2018. DOI: 10.2174/1874431101812010042.

[24] H. Moghaddasi, "Application of a real-time dashboard to reduce ventilator-associated pneumonia in intensive care units," *International Journal of Pulmonary Respiratory Sciences*, vol. 1, Feb. 2017. DOI: 10.19080/IJOPRS.2017.01.555557.

[25] Apr. 2024. [Online]. Available: https://docs.docker.com/get-started/overview/.

[26] M. Sauthier, G. Tuli, P. Jouvet, J. Brownstein, and A. Randolph, "Estimated pao2: A continuous and noninvasive method to estimate pao2 and oxygenation index," *Critical Care Explorations*, vol. 3, e0546, Sep. 2021. DOI: 10.1097/cce.0000000000000546.

[27] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 2623–2631, ISBN: 9781450362016. DOI: 10.1145/3292500.3330701. [Online]. Available: https://doi.org/10.1145/3292500.3330701.

[28] Sep. 2021. [Online]. Available: https://www.ibm.com/topics/bagging.

[29] S. Dash, *Decision trees explained - entropy, information gain, gini index, ccp pruning..* Nov. 2022. [Online]. Available: https://towardsdatascience.com/decision-trees-explained-entropy-information-gain-gini-index-ccp-pruning-4d78070db36c.

[30] A. Saini, *Guide on support vector machine (svm) algorithm*, May 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/.

[31] M. A. Bin Altaf and J. Yoo, "A 1.83 μj/classification, 8-channel, patient-specific epileptic seizure classification soc using a non-linear support vector machine," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 1, pp. 49–60, 2016. DOI: 10.1109/TBCAS.2014.2386891.

Appendices

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 24. Heatmap of missing values before data cleaning. Yellow color indicates a missing value while purple indicates a filled value.

The information corresponding to this figure has been deleted for confidentiality reasons.

Figure 25. Heatmap of missing values after data cleaning. Yellow color indicates a missing value while purple indicates a filled value.