big data and cognitive computing

MDPI

Article

Scalable Predictive Modeling for Hospitalization Prioritization: A Hybrid Batch–Streaming Approach

Nisrine Berros [1,*], Youness Filaly [1], Fatna El Mendili [2] and Younes El Bouzekri El Idrissi [1]

[1] Engineering Sciences Laboratory, National School of Applied Sciences, Ibn Tofail University, Kenitra 14000, Morocco
[2] School of Technology, Moulay Ismail University Meknes, Meknes 50050, Morocco
* Correspondence: nisrine.berros@uit.ac.ma

Abstract

Healthcare systems worldwide have faced unprecedented pressure during crises such as the COVID-19 pandemic, exposing limits in managing scarce hospital resources. Many predictive models remain static, unable to adapt to new variants, shifting conditions, or diverse patient populations. This work proposes a dynamic prioritization framework that recalculates severity scores in batch mode when new factors appear and applies them instantly through a streaming pipeline to incoming patients. Unlike approaches focused only on fixed mortality or severity risks, our model integrates dual datasets (survivors and non-survivors) to refine feature selection and weighting, enhancing robustness. Built on a big data infrastructure (Spark/Databricks), it ensures scalability and responsiveness, even with millions of records. Experimental results confirm the effectiveness of this architecture: The artificial neural network (ANN) achieved 98.7% accuracy, with higher precision and recall than traditional models, while random forest and logistic regression also showed strong AUC values. Additional tests, including temporal validation and real-time latency simulation, demonstrated both stability over time and feasibility for deployment in near-real-world conditions. By combining adaptability, robustness, and scalability, the proposed framework offers a methodological contribution to healthcare analytics, supporting fair and effective hospitalization prioritization during pandemics and other public health emergencies.

Keywords: hybrid batch–streaming ML pipeline; dynamic severity scoring; dual-dataset calibration; scalable healthcare analytics

check for updates

Academic Editor: Giuseppe Maria Luigi Sarnè

Received: 22 August 2025
Revised: 18 October 2025
Accepted: 21 October 2025
Published: 25 October 2025

Citation: Berros, N.; Filaly, Y.; El Mendili, F.; El Bouzekri El Idrissi, Y. Scalable Predictive Modeling for Hospitalization Prioritization: A Hybrid Batch–Streaming Approach. *Big Data Cogn. Comput.* **2025**, *9*, 271. https://doi.org/10.3390/bdcc9110271

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In 2020, with the onset of the COVID-19 pandemic, the world healthcare landscape met an unequivocal, monumental challenge [1]. The sudden rise in patient numbers, overwhelming emergency units up to the breaking point, caused a colossal strain on the healthcare infrastructure [2,3], calling for prompt, efficient action for patient handling and for controlling resources. In the given intricate situation, there arose an urgent need: the ability to identify correct hospitalization for the majority of the ones who were affected severely by infectious diseases, while efficiently managing all the data generated during the given health emergency [4]. But current models lacked the flexibility required to cater seamlessly to changing health crises. They had traditionally worked based on fixed standards and even fewer clinical variables, unable to handle the situation with the changing scenario of the pandemic. This brought the need for reinforcing efficient systems that

capture and effectively process patient information under high patient rate situations as generated by the given pandemic. The past decision-making systems based on a fixed set of clinical factors were not designed for the pandemic, as the former contained ample, diverse clinical information [5]. So it emerged that the stewardship of other resources in relation to hospitalization was not only the patient's personal matter of concern, but a measure that could also help offload a Herculean task on the health care systems [6]. The need of the hour was an emergent paradigm shift in the health sector regarding decision-making on hospitalization. Previous to the decision-making model, priority access to hospitalization was defined by a narrow selection of clinical variables. However, this sort of approach quickly became unwieldy due to the spiraling volume and intense complexity of patient data. Experimental results confirm the effectiveness requirement of an ICU ranges from 3% to 5%. Therefore, it became evident that streamlining the stewardship of resources with respect to hospitalization was not only an individual patient concern, but also a measure that could alleviate the gargantuan task facing healthcare systems [7]. The pandemic serves as a spotlight on the suboptimal quality of decisions regarding hospitalization solely based on clinical indicators. A process that at first sight seemed like an everyday routine task soon turned into a painful struggle, seemingly drawing the entire healthcare system into it when the numbers became overwhelming [8]. These processes were interwoven and vitally necessary, however, to develop effective measures to ensure that the patients concerned were provided with adequate care in a timely manner, and try to prevent the highly dramatic consequences that can occur when resources are constrained. However, existing work has several important shortcomings. Firstly, most current hospitalization prioritization models are static, and do not take into account the rapid evolution of pandemics. These models are often based on fixed criteria, making them unsuitable when new variants of the virus emerge or when the severity of cases changes. Secondly, although some work focuses on predicting severity or mortality risk, it fails to integrate these predictions into a practical, adaptable system for prioritizing hospitalization. Finally, the integration of these models into existing healthcare infrastructures is rarely addressed, thus limiting their real applicability in health crisis situations. To address these issues, we built a flexible model that can adapt as new information comes in. This helps ensure healthcare resources are used where they are most needed, based on what is really happening. The model uses a scoring system that combines many clinical and personal factors, so it can work in different situations—from pandemics like COVID-19 to busy times in normal healthcare settings. Figure 1 shows how the model adjusts depending on the situation to help manage resources better. At its core, the model has flexible parts that can be fine-tuned depending on the nature of the health crisis. The "Contextual Adaptability" part of the figure highlights three example scenarios:

During COVID-19, it considers things like new virus variants, lung health, and serious conditions such as diabetes or high blood pressure.

For other pandemics, like a strong flu outbreak, it takes into account factors such as asthma, COPD, patient age, and how long the illness takes to show symptoms.

In regular times, like during seasonal spikes, it adjusts based on ICU space, number of available beds, or local emergencies.

At the bottom of the figure, we list some examples of how specific variables change depending on the situation, which shows how flexible the model is. Thanks to this adaptability, the model can help guide healthcare decisions not just now, but also in future health emergencies. Our goal with this work is to create a smart, cloud-based system that helps decide who needs hospitalization most, based on many real-world factors.

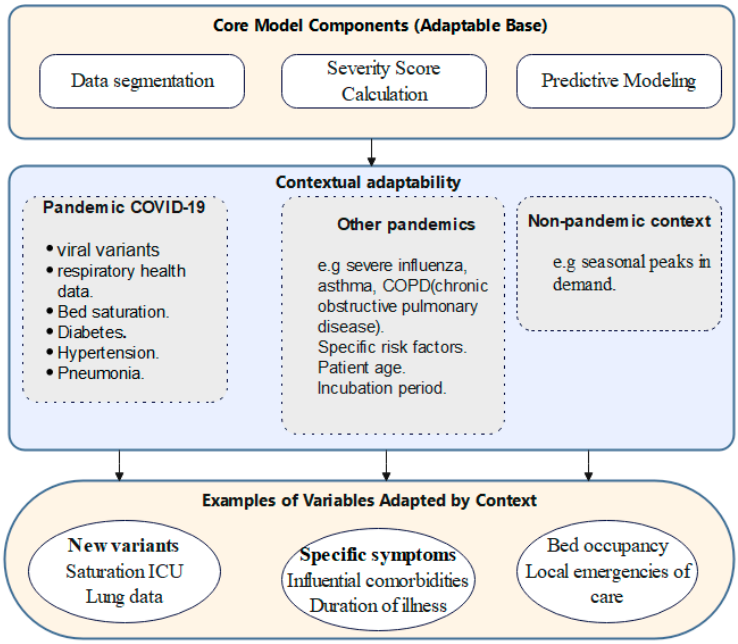

Figure 1. Dynamic and adaptable model for prioritizing hospitalization across health crises.

This new model is built to solve some of the problems seen in past approaches by offering a scoring system that updates and adjusts to the situation. Unlike static approaches, our model continuously recalculates severity scores as new data emerges, ensuring that hospitalization priorities reflect the current state of the pandemic and are not based on static criteria. In this paper, we propose a multifactorial hospitalization prioritization model that is specifically designed for pandemics in the initial acute phase. The model features a dynamic scoring system that aims to reduce systemic barriers in healthcare when demand exceeds available resources. Our approach is distinguished by its flexibility, integrating a comprehensive analysis of clinical and demographic data to give maximum priority to patients. This means that limited healthcare resources are optimally allocated, giving the most urgent patients the attention they need. This research not only provides a timely response to the current pandemic, but also a model for future health crises.

1.1. Related Work

The COVID-19 crisis has clearly highlighted how fragile hospital resource management can be. This context has motivated a growing number of studies seeking ways to improve patient triage and resource allocation through predictive models. In what follows, we outline some of the most relevant contributions and show how our work fits into this research landscape. A first line of research has dealt with disease severity and patient prioritization. The authors of [9], for example, designed the PRIORITY model, which relies on nine clinical variables and logistic regression to estimate the risk of severe COVID-19. Their approach provided a practical tool for hospitals with limited means. The study presented in [10] proposed a predictive framework combining random forest and gradient boosting, able to prioritize admissions while keeping false negatives low by drawing on diverse patient data. In Morocco, reference [11] explored dimensionality reduction with UMAP coupled with XGBoost and random forest, identifying biomarkers such as CRP and D-dimers as crucial indicators of severity. Another contribution, from Brazil [12], tested self-reported symptoms for predicting hospitalization, and the model proved useful in improving triage decisions. The study in [13] examined autoencoders to anticipate mortality, but their approach suffered from low sensitivity, often below 50%. Several

studies focused instead on algorithmic experimentation. The authors in [14] applied a Categorical Naive Bayes classifier, which reached 78.4% accuracy in assessing severity and provided support for admission decisions. In [15], the authors took a different route by combining the Enhanced Slime Mould Algorithm with SVM, attaining 91.9% accuracy, although the study was constrained by a small dataset. The research in [16] analyzed both clinical and blood test data to predict admission types, reporting accuracy levels between 87% and 97.4%. Study [17] turned to XGBoost with four biomarkers, obtaining 84.6% accuracy in detecting severe cases. Other works approached the problem from an optimization angle. Study [18] designed a multi-criteria framework (TNF-AHP and IF-CoCoSo) to rank patients according to urgency. The method allowed patients to be ranked by priority level of required action, providing an efficient process to deal with bounded resources. Paper [19] proposed a machine learning model to forecast COVID-19 mortality for COVID-19-positive patients, utilizing an artificial neural network to ascertain six significant features. The AUC was 0.953, proving to be an efficient method to make use of routine laboratory factors to predict COVID-19 intensity. Broader perspectives have also emerged in recent years. Study [20] proposed a robust optimization framework for hospital bed allocation, which significantly reduced both rejection rates and healthcare costs under uncertain demand. Research work [21] investigated the potential of large language models for real-time forecasting, combining genomic and mobility data to improve predictive responsiveness. Similarly, ref. [22] demonstrated the value of adaptive machine learning updates for severity prediction, achieving an AUC of 0.81 and showing clear advantages over static models. A systematic review [23] highlighted how machine learning and natural language processing methods can enhance triage performance in emergency departments. In [24], the authors introduced robust and sparse logistic regression techniques, leveraging L1 regularization to enhance stability and reaching up to 85% predictive accuracy on healthcare datasets, though the method lacks the dynamic batch–streaming integration needed for rapidly evolving scenarios. In parallel, ref. [6] developed a universal outbreak risk prediction tool based on machine learning, demonstrating scalability with a random forest ensemble that achieved 90% sensitivity in epidemic forecasting. However, this framework does not incorporate real-time streaming pipelines to adapt to sudden changes a gap that our hybrid design explicitly addresses.

1.2. Positioning of Our Approach

While the reviewed studies have advanced severity prediction and hospital resource allocation, they remain strongly tied to specific datasets and struggle to adapt when patient characteristics or epidemic conditions evolve. Most rely on a narrow set of variables and cannot easily integrate new clinical information, such as emerging variants or biomarkers. Our contribution addresses these gaps by introducing a dynamic scoring framework that recalculates hospitalization priorities in batch mode when major changes occur, and applies the updated rules in real time through streaming. This hybrid design combines accuracy with flexibility, ensuring that prioritization reflects the most recent knowledge while supporting fast triage for new admissions. By uniting batch and streaming processes, the model provides a practical strategy for fairer and more efficient allocation of hospital resources across diverse healthcare contexts.

2. Materials and Methods

2.1. Global Algorithmic Approach for Predicting Infection Severity and Hospitalization Needs During Pandemics

Following the widespread transmission of the pandemic, two of the most critical challenges will involve accurately assessing infection severity levels and determining

hospitalization priorities. We offer an algorithmic approach which is mainly focused on features of patient data (D) for severity classification of infections and giving precise, easily scaled-up hospitalization recommendations. Our approach comprises a number of consecutively undertaken procedures that start with data preprocessing, feature selection, and computing the illness score level, and the integration of these models into existing healthcare applies predictive methods to obtain patient outcomes. Our methodology mainly aims at determining a dynamic and adaptive model that can precisely score the severity of infection from different patient datasets, enabling clinical decision-making processes to be directed accordingly. Figure 2 illustrates the overall workflow of our methodology.

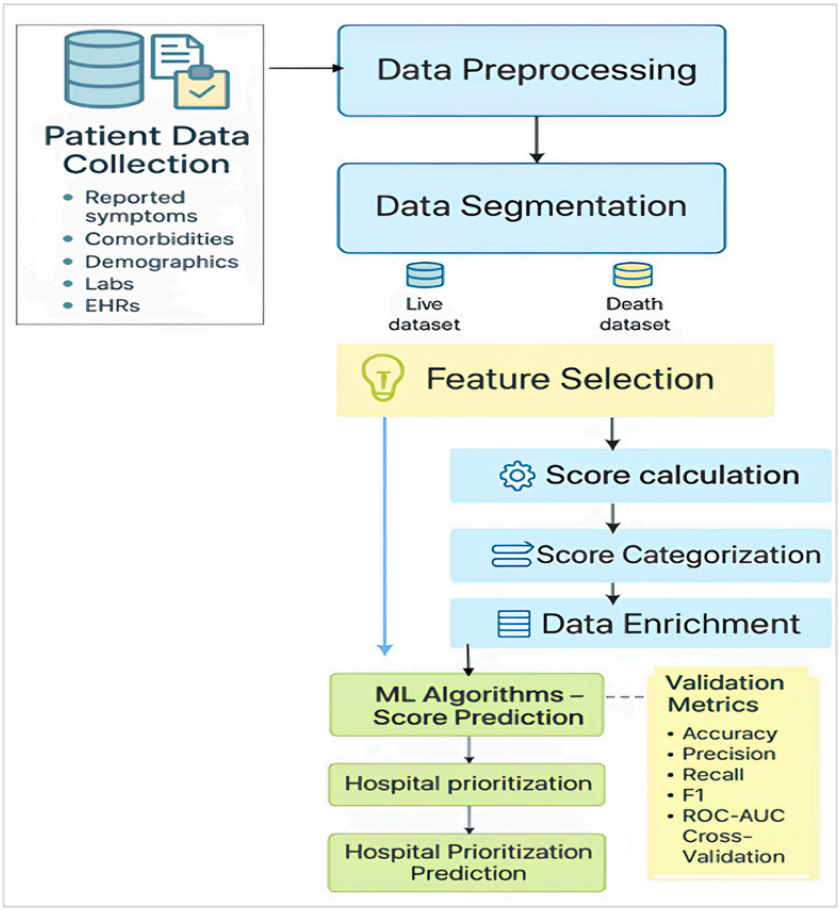

Figure 2. Multi-factor hospitalization priority assessment model.

The severity thresholds are adjustable during retraining, and could also reflect hospital capacity in future implementations.

- **Data Preprocessing (D_clean)**: The initial preprocessing of the patient dataset (D) to remove inaccuracies and achieve uniformity for further analysis.
- **Data Segmentation (D_death, D_live)**: The dataset is split into two subsets, one for patients who died due to the disease and the other for survivors (D_death and D_live), enabling focused analyses.
- **Feature Selection (F_selected)**: Identification and selection of the most critical Features (F) out of Features extracted from D_death based on their significances and effects on patient outcomes.
- **Severity Score Calculation**: For every patient in the deceased subgroup, a severity score (Si) is computed by applying weights (W) to the selected features (F_selected), showing the compounded effect of different clinical indicators.

- **Severity Score Categorization**: The calculated severity scores are then classified into distinguishable severity levels (Ci), thus creating a structured system for risk assessment of the patients.
- **Data Enrichment (D_enriched)**: Integration of the severity scores and categories back into the dataset, adding valuable insights for predictive modeling.
- **Predictive Modeling**: Creating a predictive model using D_enriched expanded cohort as the base, intended for predicting severity classes for the deceased patients.
- **Predicting for Living Patients**: Applying the predictive model to the subset of live patients (D_live), identifying their category of severity and the need for hospitalization.

Following the conceptual overview of our unified algorithmic approach, we present the detailed pseudocode to illustrate the step-by-step process of our methodology. This representation is aimed at providing a clear and executable framework for assessing infection severity and prioritizing hospitalization needs based on dynamic patient data analysis. This Algorithm 1 captures the computational reasoning behind our approach which outlines the sequence of steps performed from data preprocessing to the predictive modeling and severity classification of a patient. The purpose of each one of the steps is to make an improvement to the dataset by adding analysis-based refinement, and then enhance the dataset with analytical insights and by using the trained model to predict the severity categories of those patients who fall outside the initial deceased dataset. One of the main contributions of this work is the integration of the algorithm in our study, which intends to give a scalable and adaptable solution to the problem of efficient management of patient care during a pandemic to assign resources where they are most needed.

Algorithm 1: Assess Infection Severity and Prioritize Hospitalization

Inputs:
D: Dataset of patient data
F: Set of selected features
W: Weights associated with features F
Outputs:
C: Severity categories for each patient
Begin
1. D_clean ← *PreprocessData*(D)
2. (D_death, D_live) ← *SegmentData*(D_clean)
3. F_selected ← *SelectFeatures*(D_death, F)
4. **For each** patient di **in** D_death **do**
   4.1. Si ← *CalculateSeverityScore*(di, F_selected, W)
   End for
5. For each score Si **do**
   5.1. Ci ← *CategorizeSeverityScore*(Si)
   End for
6. D_enriched ← *EnrichData*(D_death, S, C)
7. Model ← *TrainPredictiveModel*(D_enriched)
8. **For each** patient dj **in** D_live **do**
   8.1. Cj ← *PredictSeverityCategory*(dj, Model)
   End for
9. **Return C**
End

Complementing this general process, our approach introduces a hybrid mechanism combining batch processing and streaming. In batch mode, the weights and thresholds of

the severity score are periodically recalculated from all available data, ensuring that the criteria are regularly updated. Depending on the operational context, this retraining can follow a fixed schedule (for example, daily or weekly) or be triggered by events such as a sudden influx of patients or the detection of data drift. Streaming mode, on the other hand, immediately applies these criteria to any new patient entering the system, ensuring near-instant prioritization and reducing clinical reaction time. This combination improves both the accuracy of decisions and their relevance to the rapid evolution of a health crisis.

The retraining policy can be configured either at fixed intervals or activated by specific events (e.g., surge of new patients, data drift).

As illustrated above (Figure 3), the batch mode relies on a complete analysis of historical data to recalculate variable weights and adjust categorization thresholds. The results of this phase are then fed into the streaming mode, which is limited to applying the pre-calculated criteria to the data of a newly received patient. This separation of tasks reduces operational latency, while retaining the ability to adapt when new data or risk factors emerge. This flexibility ensures that the framework can be integrated into different hospital workflows, whether through routine scheduled retraining or through adaptive updates triggered by critical events.

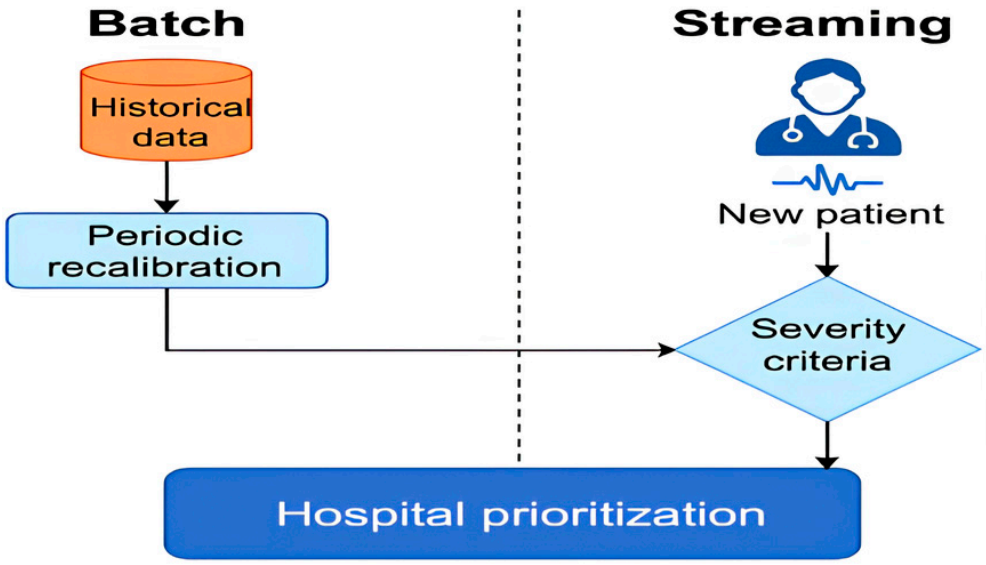

Figure 3. Diagram of the hybrid batch–streaming mechanism.

2.2. Detailed Methodology

1. Dataset Description

Our investigation utilizes a comprehensive dataset obtained from Kaggle, featuring 1,048,575 instances, each delineating the clinical progression of an individual affected by COVID-19. The dataset is a rich tapestry of variables encompassing patient demographics, clinical data, comorbidities, and other pertinent factors, all of which are indispensable for a granular understanding of the disease's trajectory and patient outcomes. The dataset includes several columns that cover different kinds of patient information. Some are administrative, like 'USMER' or 'MEDICAL_UNIT', while others are clinical, such as whether the patient has pneumonia ('PNEUMONIA'), the type of patient ('PATIENT_TYPE'), or if they were intubated ('INTUBED'). It also contains demographic data—for example, gender is recorded under 'SEX' using binary values: '1' usually means "yes" or "present," and '2' means "no" or "absent". Table 1 below is a small sample of the dataset showing the range of values for some of these features:

Table 1. Dataset description.

Column	Values
USMER	1 = Yes (patient treated in USMER unit), 2 = No
SEX	1 = Female, 2 = Male
PATIENT_TYPE	1 = Returned home (ambulatory), 2 = Hospitalized
INTUBED	1 = Yes, 2 = No, 97 = Not applicable, 99 = Unknown
PNEUMONIA	1 = Yes, 2 = No, 99 = Unknown
PREGNANT	1 = Yes, 2 = No, 97 = Not applicable, 98 = Unknown
DIABETES	1 = Yes, 2 = No, 98 = Unknown
COPD	1 = Yes, 2 = No, 98 = Unknown
ASTHMA	1 = Yes, 2 = No, 98 = Unknown
INMSUPR	1 = Yes, 2 = No, 98 = Unknown
HYPERTENSION	1 = Yes, 2 = No, 98 = Unknown
OTHER_DISEASE	1 = Yes, 2 = No, 98 = Unknown
CARDIOVASCULAR	1 = Yes, 2 = No, 98 = Unknown
OBESITY	1 = Yes, 2 = No, 98 = Unknown
RENAL_CHRONIC	1 = Yes, 2 = No, 98 = Unknown
TOBACCO	1 = Yes, 2 = No, 98 = Unknown
CLASSIFICATION_FINAL	1 = Confirmed COVID-19 by lab test, 2 = Confirmed by clinical–epidemiological association, 3 = Confirmed by expert judgment, 4 = Suspected case (no test), 5 = Not confirmed, 6 = Test negative, 7 = Pending test result
ICU	1 = Yes, 2 = No, 97 = Not applicable, 99 = Unknown

Some columns use special codes to handle missing or inapplicable data—for example, '97' means "not applicable," '98' means the data is not available, and '99' is used when the information was not specified. In the 'DATE_DIED' column, most entries are either actual dates of death or the placeholder '9999-99-99', which likely means the patient did not die or the status is unknown. Because of this type of coding, we had to be careful during data preprocessing to make sure our analysis would be accurate and reliable.

2. Data Preprocessing

Our first step in preparing the data was simple but important. We converted all 'yes' and 'no' answers into binary codes, using '1' for yes and '0' for no, to make the dataset easier to handle. Missing values were marked as '−1' throughout so we could spot and treat them quickly. For demographic info, gender was coded as '0' for female and '1' for male, following common practice. We also cleaned the data by removing some columns that were not essential for our analysis, such as ['MEDICAL_UNIT', 'USMER', 'ICU', 'DATE_DIED']. This helped us focus on the most relevant variables linked to our study goals. While checking the dataset carefully, we found nine entries with missing data scattered across the 'deceased' and 'alive' patient groups, even though no single column was missing values. Noticing these details highlighted how important thorough data cleaning is to keep the dataset reliable for accurate analysis.

3. Dataset Segmentation

To tackle the urgent challenge of deciding who should be hospitalized during this health crisis, we split our data into two groups: the 'Death dataset' for patients who sadly passed away from COVID-19, and the 'Live dataset' for those who survived. This division is important because it lets us study each group separately, focusing on their specific traits and outcomes. The 'Death dataset' helps us understand what factors contribute to mortality, which is crucial for prioritizing emergency care. On the other hand, the 'Live dataset' gives insights into recovery patterns, supporting decisions about ongoing care and how resources should be shared. By analyzing these groups separately, our goal is to build a model that better predicts who will need hospitalization, making healthcare delivery more effective during busy time. Here is an overview of the make-up of these datasets (Table 2):

Table 2. Data segmentation.

Dataset	Number of Records
Death Dataset	76,942
Live Dataset	971,633

This dual-dataset design (death/live) resembles a transfer learning step, where mortality-driven severity patterns are first captured and then generalized to survivors. Such structuring increases robustness compared to single-cohort training commonly used in prior works.

4. Feature Selection

We started the feature selection process by exploring the data to look for relationships between the variables. Using a correlation matrix, we were able to spot the key features that showed strong linear links with our main target variable, PATIENT_TYPE. Among the variables present in the dataset, some correspond to administrative information rather than patient severity. For instance, CLASSIFICATION_FINAL indicates the epidemiological classification of the case (confirmed, suspected, discarded) and was therefore not used in predictive modeling. Similarly, MEDICAL_UNIT, which refers to the type of healthcare facility where the patient was treated, was excluded during preprocessing because it reflects organizational structure rather than clinical condition, and could introduce bias into the model. The actual target outcome of our framework was PATIENT_TYPE, which distinguishes between non-hospitalized and hospitalized patients. Table 3 shows the correlation values for each of the variables we included in this analysis.

Variables like 'PNEUMONIA', which showed a strong correlation, were considered especially important. Although 'PATIENT_TYPE' is naturally linked to itself and used as a reference point in our analysis, we also kept variables with weaker correlations when they were clinically meaningful and did not overlap with others. For example, the variable AGE showed a very low linear correlation with hospitalization outcomes in our dataset. This does not reduce its clinical importance, since the relationship between age and severity is well known to be non-linear, with risk remaining low at younger ages and rising sharply at older ages. While this pattern is not reflected in the linear correlation coefficient, it is effectively captured during the modeling phase by the machine learning methods used in our framework, which are able to account for non-linear relationships. Including AGE as a feature was therefore essential from both a clinical and methodological perspective. Figure 4, which shows the correlation matrix, gives a clearer picture of how the variables relate to each other and supports the logic behind our feature choices.

Table 3. Correlation coefficients.

Variable	Correlation
PREGNANT	0.001278
OBESITY	0.010314
AGE	0.011176
COPD	0.013767
CLASSIFICATION_FINAL	0.015668
SEX	0.015680
CARDIOVASCULAR	0.016966
ASTHMA	0.018245
TOBACCO	0.019461
RENAL_CHRONIC	0.019472
INMSUPR	0.022671
DIABETES	0.024625
OTHER_DISEASE	0.027784
HYPERTENSION	0.030368
PNEUMONIA	0.122106
PATIENT_TYPE	1.000000

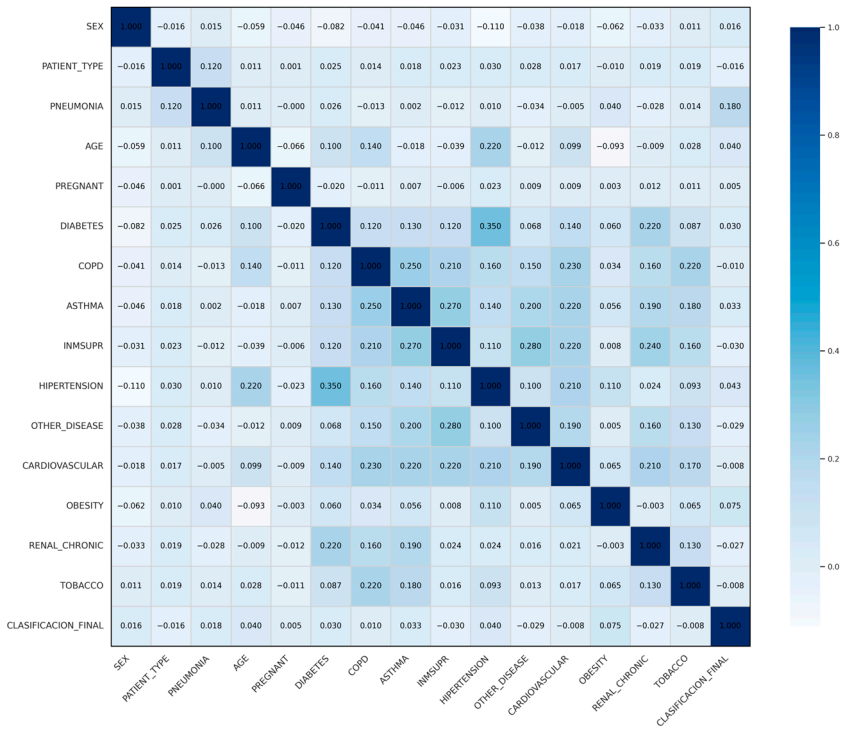

Figure 4. A heat map graph for the correlation matrix.

By choosing features based on the data itself, we made sure our model was not only statistically reliable but also made sense from a clinical point of view.

5. Severity Score Calculation

Calculating the severity score was a key part of our method. We used a multi-factor approach to build an adaptive scoring system, where clinical and demographic variables

were given different weights based on how strongly they were linked to fatal outcomes. For example, even though the correlation matrix showed that 'HYPERTENSION' and 'DIABETES' had lower coefficients compared to 'PNEUMONIA', they were still considered relevant and given appropriate weight in the final score. This mechanism in the score is dynamic in that it adapts to the constantly evolving phase of the pandemic, with weights recalculated in response to new data and emerging clinical evidence. In practice, these weights were directly tied to patient outcomes: they were initialized based on the prevalence differences between deceased and survivor cohorts, and adjusted by correlation coefficients. This ensured that the severity score remained grounded in clinical reality and aligned with observed outcomes. The weights were computed as follows:

- For the risk difference, we simply compared how often each comorbidity appeared among deceased patients versus those who survived, and then took the absolute gap between the two proportions. Here, 'risk difference' refers to the difference in prevalence of a feature between deceased and survivor groups. For example, if diabetes was present in 20% of deceased patients and 10% of survivors, the count difference equals 0.10.
- Correlation Adjustment: Each risk difference was adjusted by multiplying it with the associated correlation coefficient.

$$w_i = \Delta_i \times r_i \tag{1}$$

- In this formula, Δ_i shows how much a feature differs in prevalence between patients who died and those who survived, and r_i is its correlation value. We then adjusted the weights so they stay roughly between -1 and $+1$, making the variables easier to compare. This basic scaling also helps prevent any single factor from dominating the score and keeps the process simple to repeat with other data.

The severity score for each patient was calculated using a weighted sum of selected features, each representing different clinical and demographic factors (Table 4) The formula for the severity score (S) is as follows:

$$S = w_1 \cdot f_1 + w_2 \cdot f_2 + w_i \cdot f_i + \ldots\ldots + w_n \cdot f_n \tag{2}$$

where the following variable definitions apply:

- w_i represents the adjusted weight assigned to feature i, which is derived from its count difference and correlation coefficient. In some cases, such as AGE, the adjusted weight appeared negative in Table 4. This reflects the specific distribution in our dataset, where younger hospitalized patients and older survivors influenced the direction of the adjustment. More broadly, a negative weight should not be interpreted as a sign of protection. It simply means that the corresponding feature appears less frequently among fatal cases than among survivors. The direction of each weight is determined entirely by the data itself, reflecting the observed distributions in both cohorts rather than any predefined notion of risk or protection.

This does not contradict the known clinical importance of age, since non-linear effects are effectively captured by the machine learning models used in later stages.

- f_i shows the standardized value for every variable. Binary ones were set as 0 or 1, and continuous features like age were standardized with z-scores so they stay on a similar range. The sign of each factor came straight from what was observed in the data how often it appeared in deceased compared with surviving patient, not from any manual choice about risk or protection.

Table 4. Normalized risk differences and adjusted feature weights used in severity score calculation.

Feature	Risk Difference (Δ)	Adjusted Weights
PNEUMONIA	+0.12	+0.36
SEX	+0.05	+0.29
AGE	−0.03	−0.37
HYPERTENSION	−0.03	−0.32
DIABETES	−0.04	−0.43
OBESITY	−0.03	−0.40
TOBACCO	−0.07	−1.11
RENAL_CHRONIC	−0.06	−1.15
CARDIOVASCULAR	−0.12	−1.04
COPD	−0.05	−0.85
INMSUPR	−0.06	−1.00
ASTHMA	−0.05	−0.90
PREGNANT	−0.01	−0.09

Equation (2) is designed to be dynamic, allowing periodic recalculations as new data becomes available. The severity score formulation can be seen as an adaptive feature-weighting mechanism, periodically recalibrated to mitigate concept drift. During batch retraining (as described in Section 4.7), the severity score weights were updated at each cycle using new data blocks. This sequential recalibration illustrates how the scoring system adapts over time and mitigates temporal drift, ensuring that the criteria remain aligned with evolving patient outcomes. This positions the model not only as a medical score but as a methodological contribution in dynamic risk scoring. In this study, we tested the model using historical data integrated into Databricks, which does not involve real-time updates. Nonetheless, the structure of the model makes it possible for implementation in real-time settings in which recalculations can be performed upon arrival of new data. Using the Death dataset for this calculation ensures that the feature weights are grounded in confirmed fatal outcomes rather than uncertain or mixed cases, which strengthens the reliability of the resulting severity pattern.

Dynamic Scoring System and Adaptability: The model features an adaptive dynamic scoring system that recalculates the severity score by varying the weights for important clinical and demographic variables in response to new data. Such structure makes it possible for the model to adjust to varied health emergencies by recalibrating variable importance based on their dynamic correlations with outcomes. To illustrate, in case certain factors like age, respiratory markers, and new comorbidities prove predictive in some situation, the model will allot a higher weight to them to determine severity. This flexibility makes it possible for the model to address varied pandemic features, making it not just relevant for COVID-19 but other future health emergencies or seasonal healthcare needs. To enhance readability and ensure direct consistency with the mathematical formulation, Table 4 presents the normalized risk differences (Δ) and the adjusted feature weights (w_i), which were derived from the previously described prevalence values and correlation coefficients. The normalization process was intended to scale all adjusted weights within the [−1, +1] range, but slight deviations were maintained to preserve the natural proportional differences among highly correlated clinical variables.

This scoring system is particularly relevant in pandemic condition management, where hospital resources tend to most commonly be in short supply and must therefore

be distributed responsibly. This system enables choice-makers to directly ascertain those individuals to whom hospitalization is to be admitted and those on whom follow-up can ideally be performed on an outpatient basis or postponed. Our model is thus not confined to patient prioritization on the basis of their condition alone but also offers involuntary flexibility and agility in handling new variants or altering treatment regimens.

6. Severity Score Categorization

The severity score grouping used in our research was duly specified to reflect the clinical nuances and therapeutic implications relevant to COVID-19. By examining in depth the dataset that consists of deceased patients (Death dataset) as well as patients who survived (Live Dataset), we derived concrete thresholds to define severity groups: −24–79, 79–123, and >123. This was achieved by conducting an in-depth statistical analysis of severity scores which emphasizes our structured manner in determining patients' profiles. The use of this approach was motivated by sophisticated clustering and segmentation analysis techniques, enabling us to capture any intrinsic groupings between severity scores. It allowed us to identify the thresholds which are relevant in the sense that they distinguish patients in terms of their health progress. A number of sensitivity and specificity assessments were performed to substantiate these thresholds, thus confirming their clinical validity and ability to improve models' predictive accuracy. Moreover, our adaptable method is capable of dynamically adjusting these categorization thresholds as new patterns emerge. This can be achieved by incorporating new data or by responding to the changing perception of disease. This flexibility is crucial to enable the continuous readjustment of severity criteria in the wake of a changing pandemic. In this study, the recalibration of severity thresholds was based solely on patient-level outcomes, since operational indicators such as hospital bed availability were not available in the dataset. Even so, the framework was designed with enough flexibility to accommodate such variables in future applications. By doing so, the thresholds separating severity categories could evolve not only with clinical data but also with hospital capacity, making the prioritization process more responsive to real-world constraints during crises. The severity score was first built using data from the Death group. To make sure it also applied well in practice, we compared it with the Live data and verified that the limits 24–79, 79–123, and over 123 still made sense from a clinical point of view and matched the same kind of patient patterns. Through this score categorization step inclusion in our workflow, we completed our COVID-19 severity analysis. This inclusion enhances the capacity to accurately identify the patients who need urgent hospitalization and, thus, leads to improved usage of scarce medical resources. Our methodology demonstrates that we are strongly committed to an approach that is data-based and adaptive. This method underscores the system's flexibility in adapting to the evolving context of COVID-19 management. The following table (Table 5) presents the defined severity score intervals:

Table 5. Score categorization.

Severity Categorization	Score Intervals	Description
Low	[−24–79]	low risk of hospitalization
Moderate	[79–123]	moderate risk, special care required
High	[>123]	high risk, priority hospitalization

2.2.1. Hybrid Batch–Streaming Pipeline for Severity Scoring and Prioritization

To operationalize the proposed scoring framework, we implemented a hybrid batch–streaming pipeline (Figure 5). In batch mode, historical data are periodically processed to update factor weights, thresholds, and predictive models whenever major shifts occur, such as the appearance of new variants or significant clinical changes. Once updated, the recalibrated model and scoring rules are stored and made available to the streaming module.

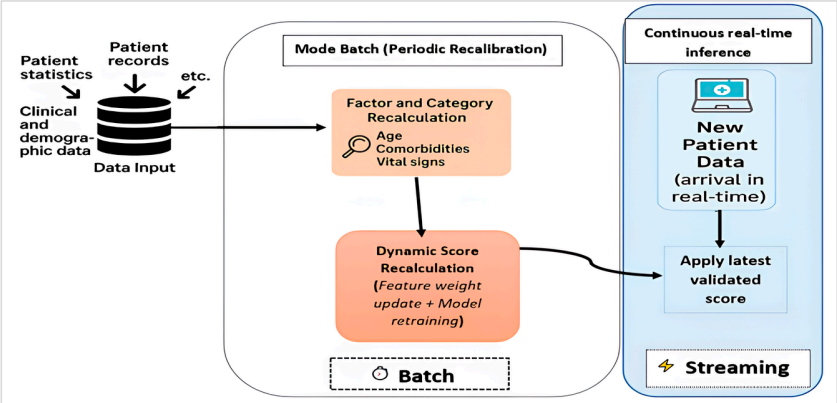

Figure 5. Hybrid batch–streaming pipeline for severity.

In streaming mode, newly arriving patient data are classified in real time using the most recent validated version, ensuring timely triage while preserving consistency with the latest recalibration. This design balances robustness, through periodic updates, with responsiveness, by enabling immediate inference on new cases.

The batch module performs periodic recalibration and model retraining, while the streaming module applies the latest score to incoming patient data in real time.

7. Data Enrichment

After establishing the severity scores, we progressed to the data enrichment phase. At this point, we added the severity scores to our dataset, creating a composite that captures individual health signals as well as an overall severity measure. This addition adds a new layer to our dataset, enabling richer modeling and analysis. During the process of enrichment, we added a severity score to each patient based on their demographic and clinical data. The scores were computed from an aggregation of chosen features across components with weights, resulting in one measure indicative of patient condition severity. This measure was further categorized to levels (e.g., low, medium, high severity) to facilitate stratified analysis as well as to inform the creation of targeted healthcare interventions. As part of the data enrichment process, we also grouped age into different ranges that reflect various life stages. This added more detail to our analysis and helped us better understand how severity scores vary across age groups. The input data consist of structured clinical records (demographics, comorbidities, treatments). After preprocessing and severity scoring, the data are fed into various machine learning models. The pipeline is implemented in a scalable distributed environment.

8. Predictive Score Modeling

At this key stage, we carefully prepared the dataset by dividing it into two clear subsets:

- Death Dataset: This group included data from patients who had passed away. We used it mainly to train our prediction models, with 80% of the data set aside for training and 20% for testing. This split was essential for building models that could accurately predict severity scores.

- Live Dataset: This group contained data from patients who survived. It was mainly used to test how well our models performed. Like the Death dataset, it was divided into −75% for training and 25% for testing. This setup helped us check how well the model could group patients by hospitalization needs, one of our main goals.

The choice to handle the Death and Live datasets separately was intentional, based on medical reasoning. The Death dataset contained patients with known outcomes, which gave us the chance to find the variables most linked to fatal cases and to build the severity score from confirmed records. The Live dataset, instead, included patients who were still in treatment, similarly to what happens in hospitals where the end result is not yet certain. By testing the model on this group, we reproduced a real triage context where predictions must be made before knowing the outcome. This separation also helped to avoid label leakage and reflects how such a system would normally be used in practice: models trained on complete data from past cases guide the assessment of new patients as they come in.

9. Model Selection and Evaluation

After preparing and splitting our dataset into the 'Death' and 'Live' subsets, we moved on to testing and comparing different machine learning algorithms. The goal was to find the most suitable models for predicting COVID-19 severity scores with good accuracy—a key factor for guiding patient care and making smart use of medical resources. It is important to note that the severity score predicted here is the same value defined earlier in Equation (1). In practice, the score was first computed on the Death dataset using the weighted formula, and then used as the target label to train the models. These models were subsequently applied to the Live dataset to estimate the corresponding severity scores for surviving patients. We explored a variety of algorithms known for their strong performance in classification and predictive tasks. Our selection was based on three main criteria:

✓ Interpretability, to provide clinical insights;
✓ Predictive accuracy, to ensure reliable severity scoring;
✓ Computational efficiency, to make sure models could be used in busy healthcare settings.

Logistic Regression [16]: Chosen for its simplicity, speed, and ease of interpretation in binary classification tasks.

Decision Trees [25]: Useful for showing clear decision paths and highlighting key factors influencing outcomes.

Random Forests [26]: Selected for its ensemble approach, which improves prediction accuracy and stability by combining results from multiple trees.

Support Vector Machines (SVMs) [27]: Known for handling high-dimensional data well and working effectively in complex classification problems.

Artificial Neural Networks (ANNs) [28]: Used for their deep learning power and ability to capture complex, non-linear patterns in the data.

Gaussian Naive Bayes [29]: Appreciated for its speed and simplicity, especially when features are independent.

XGBoost [30]: Recognized for its high performance, scalability, and efficiency in a wide range of prediction tasks.

We tested each model on both the Death and Live datasets to see how well it performed across different patient groups. Our main focus was on how accurately the algorithms could assign severity scores and how effectively they could classify patients based on clinical characteristics. During testing, we had to find the right balance between accuracy, complexity, interpretability, and processing speed. This careful comparison gave us a clear understanding of the strengths and weaknesses of each model and helped us choose the most suitable ones for real-world clinical use. The following section, Model Evaluation

Measures, explains how we assessed model performance and which criteria guided our final selection.

2.2.2. Model Evaluation Metrics

We selected the algorithms and, for performance evaluation, relied on the same metrics traditionally used in medical research [31]: accuracy, precision, recall, F1-score, and AUC. We also used cross-validation to ensure that our results were reliable, as it helps decrease overfitting and provides a fairer estimate of how well the model will perform on new incoming data. Finally, all these elements in concert define a criterion for comparing models and singling out the one that fits best with our objectives.

10. Model Parameters for Reproducibility

To improve reproducibility, we provide the main model parameters used in our experiments. For standard classifiers such as logistic regression, decision trees, naïve Bayes, and SVM, we kept the default implementations from scikit-learn and Spark MLlib, with the exception of using class_weight = balanced to account for class imbalance. More detailed settings are reported for random forest, XGBoost, and ANN, as these algorithms are more sensitive to hyperparameters and were central to our comparative analysis. In the robustness experiment (Section 4.7), the ANN was chosen as the reference model, and its architecture and hyperparameters remained fixed during retraining cycles; only the network weights were updated with new data. This ensured that the performance variations observed were attributable to adaptive retraining and not to parameter changes. Table 6 summarizes the key parameters used in this study.

Table 6. Summarizes the key parameters.

Model	Main Hyper Parameters
Artificial Neural Network (ANN)	3 hidden layers [256, 128, 64]; activation ReLU; Dropout 0.2; optimizer Adam; learning rate = 0.001; batch size = 1024; epochs = 20; early stopping (patience = 3); class_weight = balanced
Random Forest	n_estimators = 300; max_depth = None; min_samples_split = 2; class_weight = balanced
XGBoost	n_estimators = 400; max_depth = 6; learning rate = 0.1; subsample = 0.8; colsample_bytree = 0.8
Logistic Regression	solver = lbfgs; max_iter = 1000; class_weight = balanced
SVM (RBF)	kernel = rbf; C = 1.0; gamma = scale; class_weight = balanced
Decision Tree	criterion = gini; max_depth = None; min_samples_split = 2; class_weight = balanced
Naïve Bayes	var_smoothing = 1×10^{-9} (default)

3. Implementation Environment

The system was implemented on Databricks, joining data storage, large scale processing, and ML in one place. The patient records were placed on the DBFS (Databricks File System), which was linked with Amazon S3, to facilitate storing of the data safely while at the same time allowing scalability. For processing, we combined two complementary modes. The batch module was used to retrain the models and adjust the thresholds of the severity score whenever new historical data became available. In our experiments, retraining was implemented periodically using successive data blocks, which corresponds to the case of scheduled updates. In a real deployment, however, the retraining policy can be adapted: it may follow a fixed schedule (for example, daily or weekly) or be triggered by events

such as a sudden influx of patients or the detection of data drift. This flexibility ensures that the framework remains robust while fitting different hospital workflows. In practice, the retraining process uses the features and outcomes that are immediately available at admission (such as demographics, comorbidities, and laboratory tests). Outcomes that are only confirmed later, such as long-term hospitalization status or mortality, are progressively incorporated in subsequent retraining cycles. This ensures that the model remains fully operational in real time, while continuously improving its performance as more validated outcomes become available. The streaming module, on the other hand, handled the arrival of new patient data. To mimic real hospital conditions, we fed the validation dataset step by step—record by record or in small groups—using Spark Structured Streaming. Each incoming record went through the same preparation steps as in the training phase (cleaning, encoding, scaling) before being classified by the deployed model. This kept the predictions consistent with the most recent batch recalibration while providing near-instant results. The data used in these tests followed the same format as the original hospital records: semi-structured files (CSV/Parquet) containing demographic, clinical, and laboratory information, with hospitalization status serving as the reference outcome. In this way, the streaming process simulated the continuous flow of new admissions into the system. We also checked how the system behaves at scale. With more than one million records, the infrastructure stayed responsive and accurate, showing that it can handle the pressure of a large-scale health crisis. By implementing the workflow in a distributed cloud environment (Databricks + Spark MLlib), the system demonstrates methodological scalability, showing that the architecture can process over one million records with low latency. This operational aspect distinguishes the framework from prior static or small-scale implementations. For preprocessing and modeling, we relied on common Python and Spark libraries. Scikit-learn supported the basic transformations, while Spark MLlib was used for distributed training and inference. All experiments were conducted using Databricks Runtime 12.2 (Apache Spark 3.3.2) with Python 3.9 and Spark MLlib 3.3.2. This setup allowed us to validate the hybrid batch–streaming architecture and to deliver the final output: a hospitalization prioritization tool accessible through dashboards and alerts. The implementation relies on a hybrid design that separates batch retraining from real-time inference. Figure 6 summarizes this architecture, highlighting the data flow from sources to clinical applications.

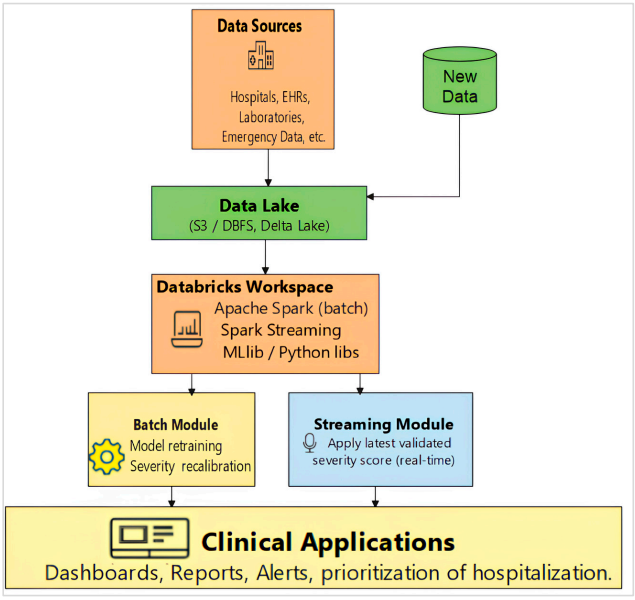

Figure 6. Overview of the implementation workflow (batch and streaming modules).

4. Results

This section gives a more precise overview of outcomes gained from our research, notably with regard to the demographics of patients, severity score allocation, case categorization, and correlation with performance levels of various different algorithms utilized by machine learning computational models.

4.1. Age Category Analysis

We conducted an age categorization analysis to describe the demographic breakdown in COVID-19 patient data. The age values were discretized into fixed ranges to enable an organized consideration of the demographics in the dataset. The age ranges used were 0–30, 31–50, 51–70, 71–90, and 91–119 years, spanning the whole age range from infancy to elderly. Each patient was assigned an age group, and this was then utilized for aggregating and counting the records in the dataset. This stratification is shown in Figure 7.

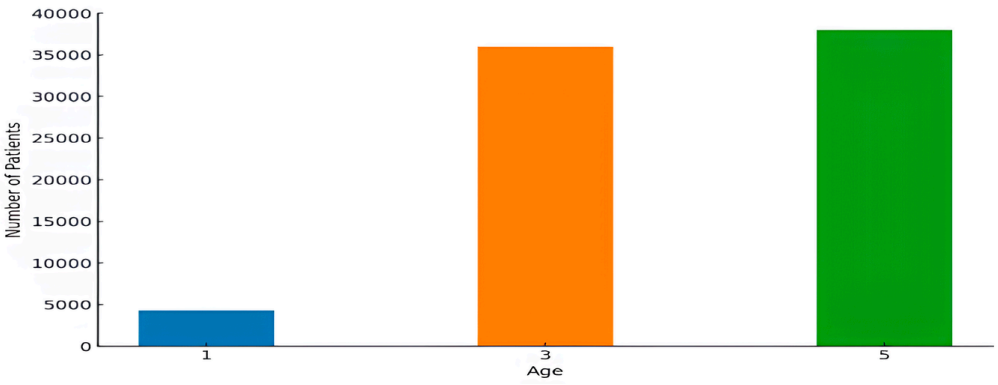

Figure 7. Age categorization.

4.2. Age Distribution and Hospitalization Patterns

Our analysis reveals a marked trend in the age distribution of patients when crossed with the "PATIENT_TYPE" variable, as shown in Figure 8. Notably, age categories "3" and "5"—corresponding to the 50–70 and 70–90 age brackets—predominate among hospitalized patients (type 1). This strong predominance of older age groups suggests a higher propensity for hospitalization, which may reflect increased vulnerability or a higher risk of serious consequences in these demographic groups.

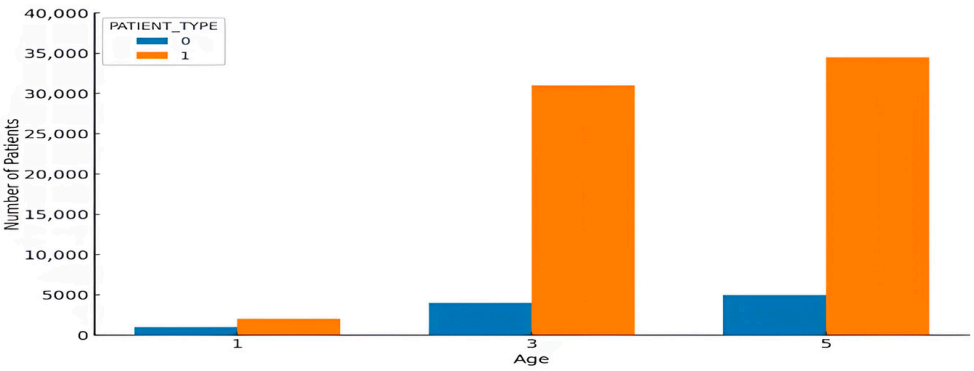

Figure 8. Age group distribution by patient type.

4.3. Analysis of Severity Scores by Age Category

Based on the comprehensive review of age-related risks, our attention shifted to constructing a proper severity rating, which is a requirement for any study. This score represents a group of clinical markers to assess the risk to patient outcomes in relation

to COVID-19. From the analysis of the severity scores distribution (Figure 9), it becomes clear that severity scores are divided into three groups of intervals for different age groups. Surprisingly, the majority of the data is contributed to the oldest age group (5), where the highest number of patients are observed for all the severity levels, which shows that severity may be related to age. On the other hand, the first age bracket has the least number of patients regardless of the severity level. The middle-aged group (3) portrays a balanced distribution, but with a density at the severity score of 2 suggesting that the middle-aged population largely experience a moderate level of severity. This analysis also supports the study conducted on the relationship between age and condition of the patients as a reminder of the need to develop management strategies for every age group.

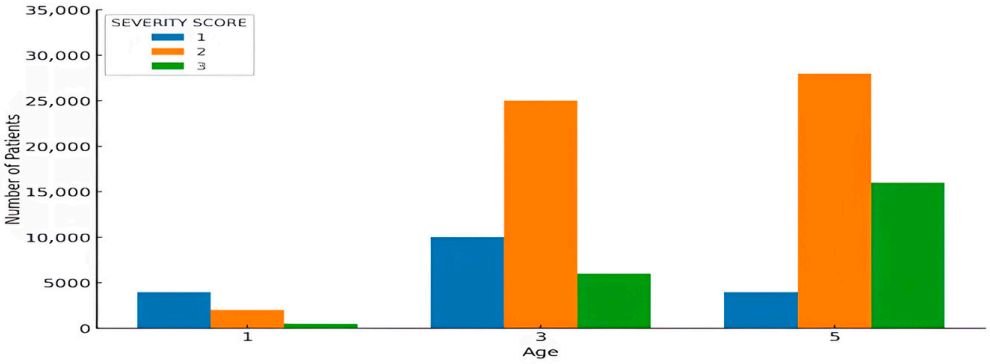

Figure 9. Patient distribution by severity category.

4.4. Severity Score Categorization

Significant trends in the distribution of patients based on risk are revealed by our study's classification of severity ratings. The low severity category (1) has the fewest individuals, as seen in Figure 10, highlighting the more restricted frequency of less severe instances. The category of moderate severity (2) represents the largest proportion of patients, indicating a substantial occurrence of cases that require targeted medical attention, whereas the group of high severity (3) accounts for a smaller but critical share, representing patients who demand immediate and intensive management within the hospital setting. These findings demonstrate how well our approach stratifies risk, which is a crucial step in allocating medical resources efficiently as possible.

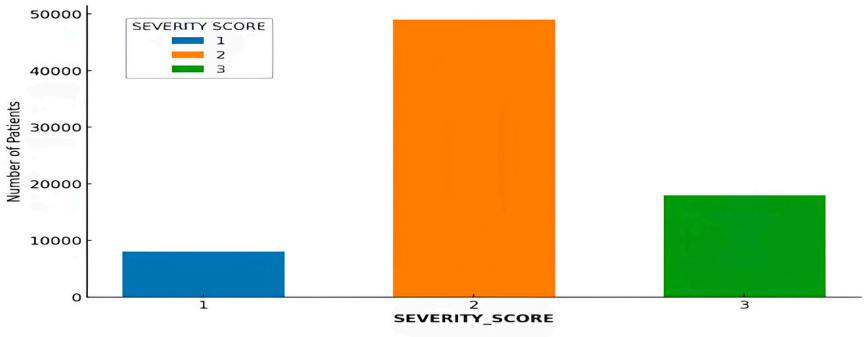

Figure 10. Distribution of scores severity by age.

4.5. Correlation Analysis

An in-depth analysis of correlations within the "dead" dataset in relation to the "SEVERITY_SCORE" variable reveals a range of impactful relationships (Figure 11). The attribute "PREGNANT" manifests the lowest correlation at 0.031, suggesting a relatively minor influence on severity scores. Progressing upwards, attributes such as "OTHER_DISEASE" and "INMSUPR" show slightly more robust correlations, at around

0.036, while conditions such as "COPD" and "CARDIOVASCULAR" reveal moderate correlations of 0.056 and 0.082, respectively. The correlation between "TOBACCO" and "PATIENT_TYPE" is more pronounced at 0.114 and 0.126, indicating a more significant effect on severity scores. Notably, "AGE" showed a significant correlation of 0.465, underlining its essential role in determining severity. Of particular interest, "COVID-19_STAGE" appears with the highest correlation at 0.609, underlining a powerful link with severity scores. Such discernible correlations play a key role in identifying the essential factors influencing the severity scoring system. Such information is invaluable in driving the development of effective predictive models and refining patient care protocols.

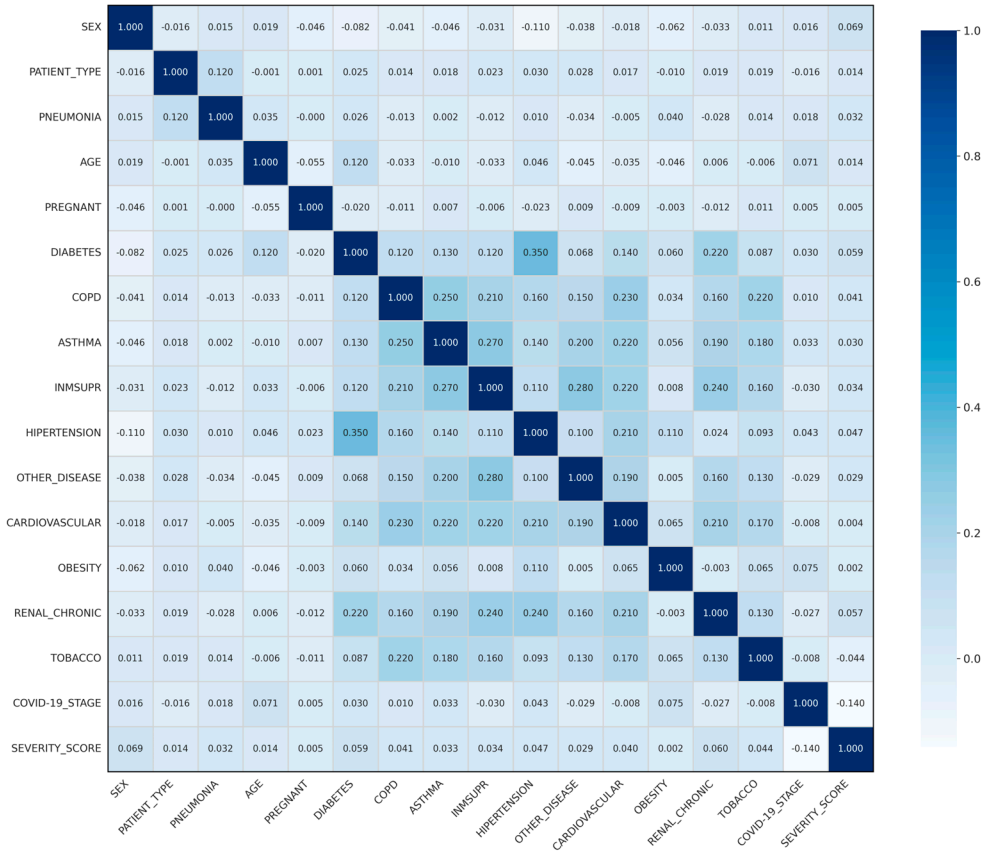

Figure 11. Analysis of correlations within the dead data set in relation to the "SEVERITY_SCORE".

4.6. Comparative Analysis of Algorithm Performance

In this study, multiple ML techniques were used to identify the outcomes of COVID-19 with the help of multiple and diverse patient features. The examples of the algorithms are logistic regression, random forest, k-nearest neighbors, naive Bayes, decision trees, cross-boost with gradient tree boosting, and support vector machines. The above set of metrics are used to have comparative benchmarks for their actual performance with three different category scores which includes precision (P), recall (R), and F1 score (F), and overall ROC AUC score and cross-validation mean accuracy were also calculated.

4.6.1. Death Dataset:

In the evaluation of machine learning algorithms for predicting COVID-19 severity across three risk categories—low, medium, and high—we analyzed precision, recall, F1 scores, ROC AUC scores, and cross-validation mean accuracy. As shown in Table 7, logistic regression excelled in all categories, achieving nearly perfect scores and

the highest ROC AUC score (0.9999) and cross-validation accuracy (0.9953), confirming its exceptional reliability.

Table 7. Comparative performance of machine learning algorithms on COVID-19 severity prediction.

Metric/Algorithm	Logistic Regression	Random Forest	ANN	Naive Bayes	Decision Trees	XGBoost	SVM
Category 1 Metrics							
p	0.99	0.98	0.97	0.69	0.84	0.94	0.90
R	0.98	0.99	0.98	0.40	0.87	0.93	0.88
F	0.99	0.99	0.98	0.51	0.87	0.94	0.89
Category 2 Metrics							
P	1.00	1.00	1.00	0.93	0.84	0.95	0.91
R	1.00	1.00	0.99	0.92	0.87	0.94	0.89
F	1.00	1.00	1.00	0.93	0.87	0.95	0.90
Category 3 Metrics							
P	0.99	0.99	0.98	0.62	0.91	0.94	0.90
R	0.99	0.99	0.98	0.75	0.89	0.93	0.88
F	0.99	0.99	0.98	0.68	0.90	0.94	0.89
ROC AUC Score	0.9999	0.9999	0.9980	0.9351	0.9531	0.9351	0.9451
Cross-Validation Mean Accuracy	0.9953	0.9946	0.9905	0.82	0.93	0.96	0.92

Close behind, random forest and XGBoost demonstrated robust performances, with random forest achieving slightly superior recall in the low-risk category. However, to complement this approach and tackle the more complex, non-linear relationships within the data, we also employed artificial neural networks (ANNs). While logistic regression provided robust and interpretable results, ANNs facilitated deeper insights and enhanced predictive power in scenarios where the interaction between variables was more intricate.

This dual strategy of leveraging both logistic regression and ANNs enabled a more comprehensive analysis and prediction of COVID-19 severity, particularly in the "Death dataset." The use of ANNs alongside other models demonstrates the flexibility of our approach in handling different levels of data complexity and the potential for application in various pandemic-related scenarios. Naive Bayes showed variability with significantly lower scores in the low-risk category, indicating challenges with datasets where the assumption of independence among features may not hold. Meanwhile, decision trees and SVM delivered moderate performances, with decision trees outperforming SVM but both unable to reach the effectiveness of the top models, particularly in identifying low-risk cases.

4.6.2. Test 2 Live Dataset:

The study moved further into the living patient data set through working on data preparation as well as processing steps, with similar approaches as the deceased patient data set. Our model, which was trained using the data from the patients that have succumbed to the disease, served us in mimicking the process of data segmentation and optimization to include predictive severity scores. It was then applied to the current status of patients to estimate the hospitalization needs. This particular stage allowed us to ensure the model's validity and its generalization capacity for cases not previously encountered. This analysis

is visualized in Figure 12, which shows the distribution of patients by severity score and type of hospitalization.

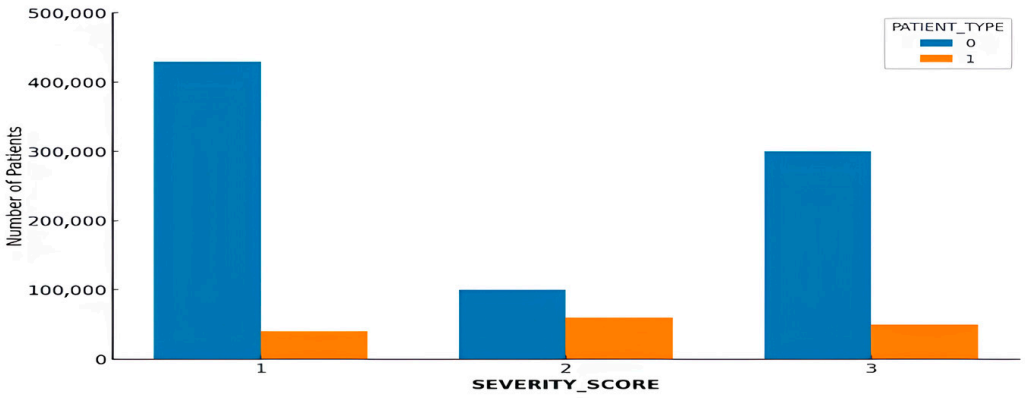

Figure 12. Distribution of patients by severity score and type hospitalization.

We conducted a thorough comparative evaluation of multiple machine learning models for predicting patient type in the context of COVID-19 (as shown in Table 8), separating those who require immediate hospitalization (Type 1) from those who may be handled at home (Type 0). This section describes the performance of each model. The study conducted showed the accuracy of the algorithms regarding patients assigned to home treatment (PATIENT TYPE = 0). Significant findings indicate that the ANN (artificial neural network) has superior performance perceptually (P = 0.98 and R = 0.97) as opposed to other algorithms such as the logistic regression and the random forest algorithms. The results indicate that the logistic regression and random forest algorithms performed with a score of 0.93 in precision and 0.96 in recall.

Table 8. Comparative performance of machine learning algorithms for the prediction of priority hospitalization.

Metric/Algorithm	Logistic Regression	Random Forest	ANN	Naive Bayes	Decision Trees	XGBoost	SVM
Patient type 0 Metrics							
p	0.93	0.93	0.98	0.94	0.89	0.95	0.90
R	0.96	0.96	0.97	0.90	0.92	0.96	0.93
F	0.95	0.95	0.95	0.92	0.90	0.95	0.91
Patient type 1 Metrics							
P	0.77	0.79	0.99	0.50	0.70	0.82	0.65
R	0.51	0.50	0.53	0.65	0.55	0.60	0.48
F	0.61	0.61	0.73	0.57	0.61	0.69	0.55
ROC AUC Score	0.81	0.84	0.958	0.81	0.85	0.93	0.82
Cross-Validation Mean Accuracy	0.9116	0.9114	0.9869	0.8642	0.89	0.92	0.88

The reception of this result proves that ANNs can recognize very general and even subtle clinical signs of COVID-19 disease with less severe infection. As far as patient type 1 (requiring hospitalization) is concerned, the ANN certainly proved to be advantageous with a precision score of 0.99 and recall of 0.53, thus providing improved identification of such cases when compared with models like random forest (P = 0.79, R = 0.50) and

logistic regression (P = 0.77, R = 0.51), or even decision trees (P = 0.70, R = 0.55) and XGBOOST (P = 0.82, R = 0.60). Table 8 below shows that the ANN achieved a remarkable score, indicating it is extremely good at determining who might need to go to the hospital and who might not. With an average accuracy rate of nearly 99% across multiple tests, it proved to be both consistent and dependable, doing a better job than any other methods we looked at. The findings clearly show that the artificial neural network (ANN) is the top choice for predicting what we need it to, thanks to its accuracy, ability to correctly identify cases, and standout scores in key tests. This makes an ANN an incredibly valuable tool for doctors and hospital staff, helping them make better decisions and manage hospital resources more efficiently, especially during the pandemic.

While the performance of our model, particularly with an ANN, is statistically impressive (with accuracies of up to 99%), it is crucial to demonstrate how these figures translate into concrete improvements in patient care and hospital resource management. A high accuracy rate, such as that achieved with our model, means that we can identify with great reliability those patients who require immediate intervention (patient type 1). This translates into a more judicious allocation of hospital resources, such as ICU beds, to the patients who need them most, reducing the risk of unnecessary overloads and lack of resources for critical cases. However, we recognize that high model accuracy does not automatically guarantee better clinical outcomes in all situations. During a pandemic, when the situation can change rapidly, it is not enough for a model to be accurate—it also has to stay flexible. That is why we did not stop at building a model that just predicts well. We made sure it could adjust in real time, as new information came in. This way, resources can be managed smarter and sooner, instead of always playing catch-up. And just as important, it helps clinicians make quicker decisions and ensures the whole system remains highly responsive to whatever comes next.

4.7. Robustness and Real-Time Performance Evaluation

Beyond the evaluation on live test data, we designed two additional experiments to examine both the robustness and the real-time applicability of the proposed model. The first was a temporal validation, in which the dataset was split into successive chronological blocks to mimic deployment conditions. Since the ANN had already shown the best performance in earlier experiments (ROC AUC 0.958, accuracy 0.9869), this model was selected as the reference for the robustness test. The goal was not to re-run all candidate models but to illustrate how the framework can adaptively retrain the best-performing classifier over successive data inflows. The Live dataset (971,633 patients) was divided into four balanced subsets of about 243,000 patients each. Block 1 was used exclusively for initial training (no evaluation metric was reported on this block). ANN v1 was then tested on Block 2 (AUC = 0.958), ANN v2 was retrained on Blocks 1–2 and was tested on Block 3 (AUC = 0.945), and ANN v3 was retrained on Blocks 1–3 and was tested on Block 4 (AUC = 0.932). This sequential partitioning was designed to simulate successive patient inflows and provides a proof-of-concept for adaptive retraining under evolving conditions. The second experiment focused on operational feasibility. Using Databricks, we compared the latency of predictions in a classical batch setting (Spark) with that obtained in a streaming environment (Spark Structured Streaming). The results confirmed that the ANN preserved a high level of performance across the retraining cycles, with the AUC decreasing only slightly from 0.958 to 0.932, thus confirming robustness despite temporal drift. Concurrently, the latency benchmark via Databricks underlines its operational advantage over batch by lowering the average delay from approximately 120 ms with Spark batch processing down to just 15 ms with Spark Structured Streaming. Collectively, these

results highlight the robustness of the predictions forwarded by our model and their utility for timely incorporation into clinical decision support systems (Figure 13).

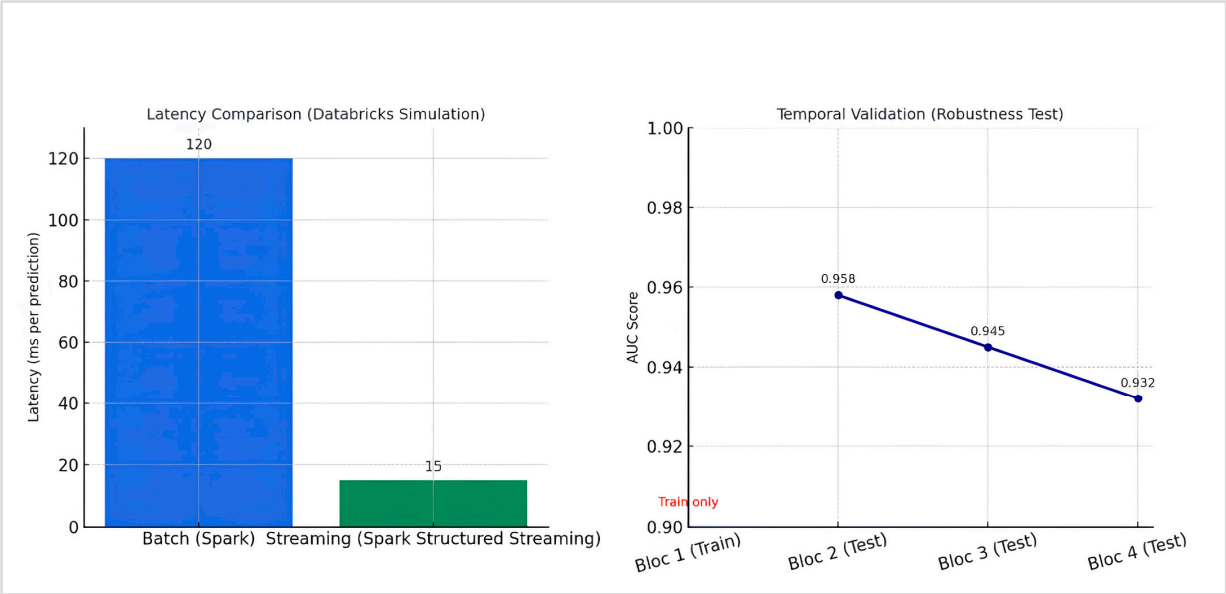

Figure 13. Robustness assessment through temporal validation and latency evaluation in batch versus streaming modes.

5. Discussion

We tested different machine learning models to see which one best predicts how severe COVID-19 cases might become and whether hospitalization is needed. Among them, the artificial neural network (ANN) stood out as the most reliable. It adjusts well to new data and evolving virus variants, which makes it practical for use in real healthcare settings. Compared to past studies, our approach gave better results. While [10] reached over 90% accuracy in predicting mortality, they did not focus on severity levels. Other models, like those used in [13,19], also showed lower performance results. Our ANN, however, reached an average accuracy of 98.69%, which shows how solid and consistent it is. To strengthen our model, we split the data into two groups—patients who died and those who survived. This allowed us to fine-tune how we selected features and assigned weights, improving how well the model predicts outcomes. Running this system on Databricks made it easier to work with large datasets and connect it smoothly with healthcare platforms. All in all, this model offers strong, adaptive performance for real-world pandemic management and helps healthcare providers use their resources more wisely both now and in future crises. The following comparative table highlights how our model outperforms other studies. This comparative analysis further validates the superior performance and flexibility of our model, reinforcing its role as a key resource for the effective management of healthcare resources during pandemics. In addition to its numerical accuracy, the model's high predictive capability for hospitalization prioritization can help hospitals make critical decisions in times of resource shortage. Timely recognition of high-risk patients enables better ICU resource utilization and more appropriate triage. To check the robustness of our model, we performed a temporal validation by splitting the dataset into chronological blocks. The model maintained a high level of performance on later groups of patients, showing that it remains stable over time and is less affected by changes in data distribution. We also tested the architecture in a real-time setting on Databricks. By comparing batch processing (Spark) with streaming processing (Spark Structured Streaming), we found that the system was able to deliver predictions with very low latency, showing that the model

can be realistically integrated into decision-support workflows in healthcare. If, during a high epidemic, even modest improvements in the speed or accuracy of prioritization help reduce overload and improve patient outcomes, the potential benefits could outweigh the cost and effort of implementing the algorithm. These results demonstrate the model's ability to enhance operational responses in future health crises, though it has not yet been tested in real hospital operations. Beyond the numerical results, the real contribution of this work lies in what it can bring to day-to-day hospital practice. A model that reliably detects severe cases reduces the risk of overlooking patients who urgently need care, which in turn helps avoid preventable complications or deaths. At the same time, by limiting false alarms, the system ensures that intensive care beds and critical resources are not occupied by patients who could be safely monitored elsewhere. The ability to deliver predictions in real time also matters: in an emergency ward where decisions must often be taken within minutes, even a small gain in speed can make a difference in patient outcomes. Taken together, these elements show that our framework is not only about reaching high accuracy rates, but about supporting faster triage, fairer allocation of beds and equipment, and more timely interventions. In this sense, the approach demonstrates its value as a practical decision-support tool for healthcare systems facing the pressure of a major crisis. Finally, although this study was conducted on COVID-19 data, the framework is flexible. The batch recalibration process makes it possible to update severity scoring when new diseases, variants, or clinical factors appear, which means the system can also be applied to other healthcare crises beyond the current pandemic.

Table 9 includes studies that, while employing different datasets and feature sets, share important conceptual similarities with our study in terms of severity scoring and hospitalization prioritization. Although these studies use various data sources, with some relying on self-reported or auto-declared data, which can introduce variability, they all contribute valuable insights into the effectiveness and limitations of diverse approaches for COVID-19 severity and hospitalization prediction. Our dataset, by contrast, incorporates a broader range of clinical and demographic variables and is based on a larger sample size, enhancing the robustness of our model's predictions. Despite these dataset differences, the comparative analysis in Table 9 provides a meaningful context, highlighting our model's adaptability and robustness in handling extensive datasets and a diverse array of pandemic-related features. This comparison underscores our model's strength in prioritizing resources and adapting to dynamic healthcare demands, even though performance metrics across studies may not always align perfectly due to dataset variations. In addition, the complementary tests we performed (temporal validation and real-time simulation) further support the robustness and practical feasibility of our approach.

Beyond the accuracy figures, the approach also brings a methodological contribution. By separating the datasets into Death and Live cohorts during feature selection, the model captures severity patterns associated with mortality before extending them to survivors. This structure acts like a transfer step, strengthening robustness compared to single-cohort models and allowing the scoring system to adapt more effectively to diverse patient outcomes.

In addition, the pipeline itself reflects a generalizable pattern in healthcare machine learning: periodic batch recalibration to adjust scores when new factors emerge, combined with real-time streaming inference to handle incoming patients with minimal delay. This dual design responds to two challenges often overlooked in previous works—scalability and adaptability. Implemented on Databricks with Spark MLlib, the system processed more than one million records while maintaining low latency, showing that the framework is not only accurate but also operationally feasible for large-scale crises.

Table 9. Comparative analysis of techniques and limitations in related works.

Reference and Year	Techniques/Methods	Best Algorithm	Accuracy	Recall	F1 Score	Limitation
Gerevini et al., 2023 [10]	- Ensemble Methods - SHAP - Model calibration for uncertainty thresholds	Decision Trees (Bagging, Boosting)	90%+ (81.9% on 4th day)	-	-	- Focuses on predicting mortality among hospitalized COVID-19 patients; does not address prediction of COVID-19 severity.
Igor Miranda et al., 2024 [12]	- ML Algorithms: decision trees, neural networks (NNs) and support vector machines - Nested cross-validation	NN	84.7%	84.6%	-	Use of self-reported data, difficulty in accurately predicting hospitalizations.
Li et al., 2020 [13]	AI methods for individual-level fatality prediction using autoencoders	Autoencoder	-	97% (on Wolfram dataset)	-	Sensitivity below 50% in death prediction. Models struggle to accurately predict death.
Jain et al., 2021 [14]	- Classification models: decision tree, KNeighbors, naïve Bayes	Naïve Bayes	78.4%	-	-	The model might not adapt well to rapidly evolving situations or different phases of the pandemic.
Shi et al., 2021 [15]	- ESMA method for feature selection - SVM model	SVM	91.91%	91.91%	-	- Small dataset - Does not delve into the practical implications of these findings in terms of hospital resource allocation or prioritization.
Darapaneni et al., 2020 [16]	- ML Algorithms: random forest, SVM, logistic regression	SVM	82%	-	-	The utilization of blood-related features in this approach may entail a substantial time investment for analysis due to their intricate and detailed nature, potentially impacting prediction timeliness.
Kang et al., 2021 [19]	- Artificial neural network (ANN)	ANN	-	85.7%	96.4%	- Relies on a relatively small dataset, potentially limiting the broad applicability of the findings. Does not address the practical implications of the results in terms of hospital resource allocation or prioritization, which is a critical aspect of effective pandemic management.
Shams Eddin and El Hajj 2025 [20]	Robust optimization framework for hospital bed allocation under uncertain demand	-	-	-	(cost reduction ~50%)	High computational complexity and reliance on precise demand data. Lacks predictive ML for dynamic feature integration, unlike our scalable Spark pipeline.

Table 9. *Cont.*

Reference and Year	Techniques/Methods	Best Algorithm	Accuracy	Recall	F1 Score	Limitation
Ayvaci et al. 2025 [22]	Adaptive machine learning updates for severity prediction	XGBoost	-	-	81%	Complex contextual updates with potential historical bias. Not scalable for big data, addressed by our Spark recalibration.
Cornilly et al. 2023 [24]	Robust and sparse logistic regression techniques with L1 regularization for stability	Logistic Regression	-	-	Up to 85% (predictive accuracy)	Static model lacking dynamic batch-streaming integration. Not suited for rapidly evolving scenarios, unlike our hybrid design.
Tianyu et al. 2024 [6]	Universal outbreak risk prediction tool based on machine learning ensembles	Random Forest Ensemble	-	90%	-	No real-time streaming pipelines to adapt to sudden changes. Lacks the dual-dataset robustness of our hybrid framework.
This Study, 2025	Dynamic scoring model; Big data infrastructure; feature selection and recalibration; predictive modeling for severity and hospitalization	ANN	98.69%	-	-	- Validated only on retrospective data. - Limited representation of pediatric and pregnant patients. - Real deployment would require HIPAA/GDPR compliance.

Limitations

This study has several limitations that should be acknowledged. First, although the model demonstrated high predictive performance, it was validated only on retrospective data from a large public COVID-19 dataset. Prospective validation in real hospital environments is still needed to confirm its effectiveness in live clinical workflows. Second, while age categories were comprehensively analyzed, the dataset contained very few pediatric and pregnant patients. As a result, the model's applicability to these sensitive subgroups remains limited, and further testing with dedicated cohorts will be required before clinical adoption for these populations. Third, although our experiments relied on anonymized and publicly available data, any real-world deployment would need to comply with strict data protection regulations such as Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR). This includes secure storage, anonymization procedures, and robust privacy safeguards. These considerations highlight the need for additional clinical studies and regulatory alignment, but they do not diminish the methodological contributions of our framework, which remain robust, scalable, and adaptable for healthcare crisis management.

6. Conclusions

Indeed, this research proposal presents a series of factors with acceptable reliability and validity indices suitable for the prioritization of hospitalizations and has proposed a dynamic scoring model that can be adjusted to the type of health emergency situation. In this paper, we have provided a brief review of machine learning, brief details of the models used in the study, such as logistic regression, random forest, and artificial neural networks, and the improvement that our proposed approach brings to the prioritization of hospitalization. Among these, the highest accuracy was observed in the artificial neural network, equal to 98% in this particular case, and also high accuracy in both precision and recall, thereby showing the high accuracy of the model in critical decision-making situations. The application of our methodology is invaluable in high-patient-volume environments stressing healthcare systems by increasing the efficiency of decision-making activities, the reliability of data, and the rational expenditure of resources. Thus, by using a complex set of varying clinical and other demographic parameters, our versatile model is better suited for the changing nature of pandemics and remains a valuable tool for assisting clinicians in bio-medical decision-making regarding the management of patients. The work does not only focus on predictive accuracy. It also explores how routine batch recalibration can run together with live inference inside a scalable cloud setup. This structure was built in a way that it could later be reused for different kinds of healthcare data analyses. While the study was primarily designed for healthcare prioritization, it also brings methodological value to broader machine learning and data engineering practice. The proposed hybrid batch–streaming architecture demonstrates, in operational terms, how retraining mechanisms can coexist with real-time inference to support evolving data streams. By drawing on two complementary datasets—the Death and Live cohorts—the framework achieves a type of transfer calibration that limits imbalance and mitigates concept drift. The severity scoring mechanism, based on adaptive feature weighting, further contributes to model stability as input distributions shift. The implementation on Spark/Databricks, evaluated on over one million records in both modes, confirms the robustness and scalability of the overall design. These elements go beyond the specific medical case and reflect general principles that are transferable to other domains of data-intensive decision support. In the future, we aim to strengthen this framework in two main directions: first, by validating the model in prospective and multi-center studies to confirm its reliability in real hospital environments; second, by extending its use beyond COVID-19 to other health crises where rapid triage

and resource prioritization are essential. These steps will help ensure that the approach remains both scientifically rigorous and practically useful for healthcare systems facing diverse challenges.

Author Contributions: Methodology design and Conceptualization, data preprocessing, model development, original draft writing, N.B.; Support in data analysis, assistance with literature review, contribution to manuscript editing, Y.F.; Supervision, methodological guidance, and critical revision of the manuscript, validation of results, F.E.M. and Y.E.B.E.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this study was originally obtained from a publicly available Kaggle repository of COVID-19 patient records. The original link is no longer accessible. An anonymized subset can be shared by the authors upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Filip, R.; Gheorghita Puscaselu, R.; Anchidin-Norocel, L.; Dimian, M.; Savage, W.K. Global Challenges to Public Health Care Systems during the COVID-19 Pandemic: A Review of Pandemic Measures and Problems. *J. Pers. Med.* **2022**, *12*, 1295. [CrossRef]
2. Ndayishimiye, C.; Sowada, C.; Dyjach, P.; Stasiak, A.; Middleton, J.; Lopes, H.; Dubas-Jakóbczyk, K. Associations between the COVID-19 Pandemic and Hospital Infrastructure Adaptation and Planning—A Scoping Review. *Int. J. Environ. Res. Public Health* **2022**, *19*, 8195. [CrossRef]
3. Pearce, S.; Marchand, T.; Shannon, T.; Ganshorn, H.; Lang, E. Emergency Department Crowding: An Overview of Reviews Describing Measures, Causes, and Harms. *Intern. Emerg. Med.* **2023**, *18*, 1137–1158. [CrossRef]
4. Jovanović, A.; Klimek, P.; Renn, O.; Schneider, R.; Øien, K.; Brown, J.; DiGennaro, M.; Liu, Y.; Pfau, V.; Jelić, M.; et al. Assessing Resilience of Healthcare Infrastructure Exposed to COVID-19: Emerging Risks, Resilience Indicators, Interdependencies and International Standards. *Environ. Syst. Decis.* **2020**, *40*, 252–286. [CrossRef]
5. Negro-Calduch, E.; Azzopardi-Muscat, N.; Nitzan, D.; Pebody, R.; Jorgensen, P.; Novillo-Ortiz, D. Health Information Systems in the COVID-19 Pandemic: A Short Survey of Experiences and Lessons Learned from the European Region. *Front. Public Health* **2021**, *9*, 676838. [CrossRef]
6. Zhang, T.; Rabhi, F.; Chen, X.; Paik, H.; MacIntyre, C.R. A Machine Learning-Based Universal Outbreak Risk Prediction Tool. *Comput. Biol. Med.* **2024**, *169*, 107876. [CrossRef]
7. Dziegielewski, C.; Talarico, R.; Imsirovic, H.; Qureshi, D.; Choudhri, Y.; Tanuseputro, P.; Thompson, L.H.; Kyeremanteng, K. Characteristics and Resource Utilization of High-Cost Users in the Intensive Care Unit: A Population-Based Cohort Study. *BMC Health Serv. Res.* **2021**, *21*, 1312. [CrossRef]
8. McCabe, R.; Schmit, N.; Christen, P.; D'Aeth, J.C.; Løchen, A.; Rizmie, D.; Nayagam, S.; Miraldo, M.; Aylin, P.; Bottle, A.; et al. Adapting Hospital Capacity to Meet Changing Demands during the COVID-19 Pandemic. *BMC Med.* **2020**, *18*, 329. [CrossRef] [PubMed]
9. Leung, C.K.; Mai, T.H.D.; Tran, N.D.T.; Zhang, C.Y. Predictive Analytics to Support Health Informatics on COVID-19 Data. In Proceedings of the 2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE), Kragujevac, Serbia, 25–27 October 2021; pp. 1–9. [CrossRef]
10. Gerevini, A.E.; Maroldi, R.; Olivato, M.; Putelli, L.; Serina, I. Machine Learning Techniques for Prognosis Estimation and Knowledge Discovery from Lab Test Results with Application to the COVID-19 Emergency. *IEEE Access* **2023**, *11*, 83905–83933. [CrossRef]
11. Laatifi, M.; Douzi, S.; Bouklouz, A.; Ezzine, H.; Jaafari, J.; Zaid, Y.; El Ouahidi, B.; Naciri, M. Machine Learning Approaches in COVID-19 Severity Risk Prediction in Morocco. *J. Big Data* **2022**, *9*, 5. [CrossRef] [PubMed]
12. Miranda, I.; Cardoso, G.; Pahar, M.; Oliveira, G.; Niesler, T. Machine Learning Prediction of Hospitalization due to COVID-19 Based on Self-Reported Symptoms: A Study for Brazil. In Proceedings of the 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), Athens, Greece, 27–30 July 2021; pp. 1–5. [CrossRef]
13. Li, Y.; Horowitz, M.A.; Liu, J.; Chew, A.; Lan, H.; Liu, Q.; Sha, D.; Yang, C. Individual-Level Fatality Prediction of COVID-19 Patients Using AI Methods. *Front. Public Health* **2020**, *8*, 587937. [CrossRef] [PubMed]

14. Jain, L.; Gala, K.; Doshi, D. Hospitalization Priority of COVID-19 Patients Using Machine Learning. In Proceedings of the 2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS), Greater Noida, India, 17–18 September 2021; pp. 150–155. [CrossRef]
15. Shi, B.; Ye, H.; Zheng, J.; Zhu, Y.; Heidari, A.A.; Zheng, L. Early Recognition and Discrimination of COVID-19 Severity Using Slime Mould Support Vector Machine for Medical Decision-Making. *IEEE Access* **2021**, *9*, 121996–122015. [CrossRef]
16. Darapaneni, N.; Singh, A.; Paduri, A.; Ranjith, A.; Kumar, A.; Dixit, D.; Khan, S. A Machine Learning Approach to Predicting COVID-19 Cases amongst Suspected Cases and Their Category of Admission. In Proceedings of the 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), Rupnagar, India, 26–28 November 2020; pp. 375–380. [CrossRef]
17. Zheng, Y.; Zhu, Y.; Ji, M.; Wang, R.; Liu, X.; Zhang, M.; Liu, J.; Zhang, X.; Qin, C.H.; Fang, L.; et al. A Learning-Based Model to Evaluate Hospitalization Priority in COVID-19 Pandemics. *Patterns* **2020**, *1*, 100092. [CrossRef] [PubMed]
18. Perez-Aguilar, A.; Ortiz-Barrios, M.; Pancardo, P.; Orrante-Weber-Burque, F. A Hybrid Fuzzy MCDM Approach to Identify the Intervention Priority Level of COVID-19 Patients in the Emergency Department: A Case Study. In *Proceedings of the Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management*; Duffy, V.G., Ed.; Springer: Cham, Switzerland, 2023; pp. 284–297. [CrossRef]
19. Kang, J.; Chen, T.; Luo, H.; Luo, Y.; Du, G.; Jiming-Yang, M. Machine Learning Predictive Model for Severe COVID-19. *Infect. Genet. Evol.* **2021**, *90*, 104737. [CrossRef]
20. Eddin, M.S.; Hajj, H.E. An Optimization-Based Framework to Dynamically Schedule Hospital Beds in a Pandemic. *Healthcare* **2025**, *13*, 2338. [CrossRef] [PubMed]
21. Du, H.; Zhao, Y.; Zhao, J.; Xu, S.; Lin, X.; Chen, Y.; Gardner, L.M.; Yang, H.F. Advancing real-time infectious disease forecasting using large language models. *Nat. Comput. Sci.* **2025**, *5*, 467–480. [CrossRef]
22. Ayvaci, M.U.S.; Jacobi, V.S.; Ryu, Y.; Gundreddy, S.P.S.; Tanriover, B. Clinically Guided Adaptive Machine Learning Update Strategies for Predicting Severe COVID-19 Outcomes. *Am. J. Med.* **2025**, *138*, 228–235. [CrossRef]
23. Porto, B.M. Improving triage performance in emergency departments using machine learning and natural language processing: A systematic review. *BMC Emerg. Med.* **2024**, *24*, 219. [CrossRef]
24. Cornilly, D.; Tubex, L.; Van Aelst, S.; Verdonck, T. Robust and Sparse Logistic Regression. *Adv. Data Anal. Classif.* **2023**, *18*, 663–679. [CrossRef]
25. Kotsiantis, S.B. Decision Trees: A Recent Overview. *Artif. Intell. Rev.* **2013**, *39*, 261–283. [CrossRef]
26. More, A.S.; Rana, D.P. Review of Random Forest Classification Techniques to Resolve Data Imbalance. In Proceedings of the 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM), Aurangabad, India, 5–6 October 2017; pp. 72–78. [CrossRef]
27. Nandi, A.; Ahmed, H. *Condition Monitoring with Vibration Signals: Compressive Sampling and Learning Algorithms for Rotating Machine*, 1st ed.; Wiley: Hoboken, NJ, USA, 2019. [CrossRef]
28. Emambocus, B.A.S.; Jasser, M.B.; Amphawan, A. A Survey on the Optimization of Artificial Neural Networks Using Swarm Intelligence Algorithms. *IEEE Access* **2023**, *11*, 1280–1294. [CrossRef]
29. Ontivero-Ortega, M.; Lage-Castellanos, A.; Valente, G.; Goebel, R.; Valdes-Sosa, M. Fast Gaussian Naïve Bayes for Searchlight Classification Analysis. *NeuroImage* **2017**, *163*, 471–479. [CrossRef] [PubMed]
30. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
31. Rainio, O.; Teuho, J.; Klén, R. Evaluation Metrics and Statistical Tests for Machine Learning. *Sci. Rep.* **2024**, *14*, 6086. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.