Diyala Journal of Engineering Sciences

Journal homepage: https://djes.info/index.php/djes

ISSN: 1999-8716 (Print); 2616-6909 (Online)

Predictive Modelling of COVID-19 Patient Outcomes Using Deep Learning Techniques

K. Kalaiselvi[1], S. Bairavel[2] , S. Sumathi[3], Kanipriya M[4*], Anitha Govindaram[5], Jose Anand A[6]

[1] Department of Computer Science & Applications, SRM Institute of Science and Technology, Ramapuram, Tamilnadu, – 600089, India.
[2] Department of Computer Science and Engineering (Internet of Things), Sri Sairam Engineering College, Chennai, Tamilnadu,– 600044, India
[3] Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, Tamilnadu, India – 600044, India
[4] Department of Computational Intelligence, SRM Institute of Science and Technology, Kattankulathur, Tamilnadu– 603203, India.
[5] Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Thandalam, Chennai, Tamil Nadu– 602105, India.
[6] Department of Electronics and Communication Engineering, KCG College of Technology, Karapakkam, Chennai, Tamil Nadu– 600 097, India

ARTICLE INFO

Article history:
Received: 15/10/2025.
Revised: 07/02/2026.
Accepted: 18/02/2026
Available online: 15/03/2026.

Keywords:
COVID-19 prediction
Neural networks
Transformer
ICU overflow
Patient outcome classification

ABSTRACT

Prediction of clinical outcome in patients with COVID-19 is important for timely intervention, appropriate ICU triage, and best utilization of hospital resources. We propose a deep learning framework to predict the patient outcomes by utilizing longitudinal clinical data from a tertiary care hospital in India. After quality control (QC) preprocessing and filtering, 126,716 diagnostic records with 13 key laboratory and vital sign variables were retained for analysis. We propose and test several neural architectures for the classification of ICU vs ward patients, namely GRU, LSTM, BiLSTM, CNN-LSTM, and Transformer models. A two-stage experimental setup was employed. We began by doing an exploratory train-test analysis for hyperparameter tuning. Next, we performed a rigorous comparative analysis based on the five-fold cross-validation results. We evaluated the performance according to such criteria as accuracy, sensitivity, specificity, recall, precision, F1-score, ROC curve and AUC. Of the models we tested, the Transformer performs best overall, with 91.9% accuracy, 85.4% sensitivity, 75.6% specificity, and 83.1% recall. It clearly demonstrates a good specificity in the view of high-risk patients, but with a minimum false negative rate. The CNN-LSTM had a comparably good performance and the GRU was an attractive low-cost alternative with reasonable predictive power and computational burden. The generalizability of our models was validated externally, where the Transformer also yielded the best performance on unseen clinical data. These results highlight the promise of deep learning, in particular Transformer-based models, as robust instruments for COVID-19 prognosis and ICU admission management.

1. INTRODUCTION

In 2020, India was severely affected by the outbreak and rapid spread of the novel coronavirus disease (COVID-19). The outbreak began in December 2019 and quickly escalated into a global pandemic. As the virus began to spread across the country, Indian authorities took drastic measures, including a nationwide lockdown, restrictions on movement, and school closures. The first wave of the pandemic in the spring and summer of 2020 hit India's healthcare system hard. Hospitals were overwhelmed, medical resources were scarce, and medical staff lacked personal protective equipment [1-3]. The elderly and those with comorbidities were particularly vulnerable, exacerbating the crisis. India has experienced multiple waves of the pandemic, each of which brought its own set of challenges. The situation has been constantly changing as the pandemic has been periodically accompanied by lockdowns and varying degrees of control measures. Despite the Indian government's concerted efforts to mitigate the impact of the pandemic, the pandemic has still had a significant impact on India's public health, economy, and the daily lives of its citizens [4-8].

Data from Johns Hopkins University [3] reflect the severity of the global pandemic: 695 million

[*] Corresponding author's E-mail: kaniprim@srmist.edu.in

DOI: 10.24237/djes.2026.19107

confirmed cases and 6.9 million deaths. These figures reflect the global spread of COVID-19 and the need for effective disease management and prognosis [9-11]. This work started in collaboration with the medical departments of large public hospitals in India, such as the All India Institute of Medical Sciences (AIIMS) or the National Institute of Mental Health and Neurosciences (NIMHAN). The goal of this work was to develop an innovative AI-driven solution for analyzing clinical data related to COVID-19 patients [12]. The goal of this work was to accurately predict the clinical course of COVID-19 patients, thereby enabling physicians to proactively manage patient care, including early ICU admission.

2. LITERATURE REVIEW

Several research works have established predictive models for estimating COVID-19 based on the patient's clinical, biochemical, and demographic status. The authors [13] created and tested a variety of machine learning models for predicting the risk of death. They concluded that the ensemble models were the best since they were able to discover the nonlinear dependencies in the clinical data, such as those of age, markers of inflammation, and kidney performance. On the other hand, the work [14] drew attention to the predictive capacity of LDH levels, stating that elevated levels of LDH are the strongest predictor of pneumonia advancement on CT scans of the chest.

The research work [15] created a mortality forecasting model based on the whole population's administrative data from Japan, and they clearly stated that the combination of the large-scale healthcare data and clinical variables was very productive. The authors [16] studied COVID-19 transmission within hospitals and its influence on death rates during different epidemic periods while pointing out dynamic risk changes.

In order to enhance the interpretability aspect, The authors [17] made use of the XAI that is the explainable artificial intelligence and thus the prediction made with the use of a very complicated neural network was made clear to the doctors. The research by [18] involved the use of the SMOTE technique for tackling the issue of class imbalance in the CNN-based patient mortality prediction and they were successful in getting high accuracy even for the less represented patient categories. The authors [19] suggested Deep-Risk which is a deep learning tool with the capability of surpassing conventional statistical methods with the help of large blended datasets. The authors [20] studied the mechanisms of lactate and LDH and confirmed their clinical significance in inflammation associated with COVID-19. Ultimately, The authors [21] looked into the neutrophil proteomes during the post-infection phase and their finding was that there is immune dysregulation which might be the reason for the long-term effects seen.

In order to facilitate the understanding of the previous research works, the authors in Table 1 provide a detailed categorization of the main studies, including the dataset used, the model applied, key metrics, limitations, and how our research is better than theirs.

Table 1. Summary of Key Previous Studies on COVID-19 Predictive Models

Study	Dataset / Population	Model / Approach	Key Metrics	Strengths	Limitations
[13]	CKD patients	ML: Logistic Regression, RF, XGBoost	Accuracy ~80–85%	Captures nonlinear clinical relationships	Limited generalizability
[14]	Hospitalized COVID-19 patients	LDH-based risk model	Correlation with severe pneumonia	Highlights biochemical markers	Single biomarker focus
[15]	National administrative dataset	Population-wide mortality model	AUC ~0.78	Large-scale data	Limited model interpretability
[16]	Hospital inpatients	Epidemiological analysis	Mortality rates	Highlights in-hospital transmission	Not predictive
[17]	Hospitalized patients	XAI-based neural network	Accuracy ~82%	Model transparency	Small dataset
[18]	Hospitalized patients	CNN + SMOTE	Accuracy 88–90%	Addresses class imbalance	Limited architecture comparison
[19]	Multi-hospital datasets	Deep-Risk NN	Accuracy ~85%	Neural network performance	Limited clinical feature integration
[20]	Literature & clinical data	Mechanistic biomarker analysis	Qualitative	Explains biochemical mechanisms	Not predictive
[21]	COVID-19 neutrophil proteome	Proteomic analysis	Qualitative	Explains immune dysregulation	Not predictive

3. MATERIALS AND METHODS

The database used in this study was developed by physicians of a tertiary hospital in India to support research on COVID-19 patient outcomes in intensive care. It originally contained 132,870 diagnostic records, including patient identifiers (anonymized), test types, results, duration, hospital setting (ICU or general ward), and test frequency. To ensure quality, the dataset was filtered in MATLAB by excluding patients with fewer than 130 total tests or fewer than nine of a given type, and data were segmented into three time windows (00:00–08:00, 08:00–16:00, and 16:00–24:00). This reduced the dataset to 126,716 records across 13 trial types. Using this refined dataset, several neural network models were implemented in MATLAB to classify ICU versus general ward status, including GRU, LSTM, BiLSTM, a hybrid CNN-LSTM, and a Transformer model, each tuned with specific hyperparameters to enhance performance. Originally applied in Spanish clinical contexts, these architectures were effectively adapted to the Indian healthcare setting for ICU admission monitoring and decision support using time-series hospital data.

3.1 Dataset Description and Diagnostic Test Types

The primary database contained a total of 132,870 anonymized diagnostic records collected from COVID-19 patients who were admitted to a tertiary care hospital in India during the pandemic. Every record contained the patient's identifiers (anonymized), the diagnostic test type, the test result value, the time the test was taken, the place where the test was done (ICU or general ward), and how often the test was done.

The application of quality-control filters resulted in the elimination of patients with less than 130 total tests or less than 9 instances of any specific test type, thus reducing the dataset to 126,716 records. The final dataset comprised 13 clinical test types frequently collected and selected based on the factors of frequency, clinical relevance, and consistency across the patients. The following are the test types: Complete blood count (CBC), C-reactive protein (CRP), D-dimer, Lactate dehydrogenase (LDH), Ferritin, Blood urea nitrogen (BUN), Serum creatinine, Alanine aminotransferase (ALT), Aspartate aminotransferase (AST), Total bilirubin, Oxygen saturation (SpO_2), Respiratory rate, and Heart rate. The diagnostic variables were chosen because they are all routinely checked measures for COVID-19 patients in the hospitals and have been associated with the patient's condition (severity) and admission (ICU risk).

3.2 Transformer Model Architecture and Training Configuration

The Transformer model used in this research had an encoder-only architecture that was specifically designed for the classification of multivariate clinical time-series. Each vector of the input features passed through a fixed-dimensional embedding space, and then it went through the stacked self-attention layers.

More precisely, every time step was represented by an embedding dimension of 128. The Transformer encoder consisted of 4 stacked encoder layers, each with a multi-head self-attention mechanism followed by a position-wise feedforward network. The self-attention module had 8 attention heads which permitted the model to recognize various temporal dependencies in clinical measurements simultaneously.

In order to maintain the temporal order information, sinusoidal positional encoding was first applied to the input embeddings before entering the first encoder layer. This method gives the model the capability of learning the temporal relationships without adding any trainable parameters, hence it is applicable to the variable-length clinical sequences.

The Adam optimizer was utilized for training the model with a starting learning rate of 1×10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ as part of the optimization process. A categorical cross-entropy loss function was adopted, and the application of early stopping based on validation loss helped to prevent overfitting. Transformer Hyperparameter Summary is provided in Table 2.

Table 2. Transformer model configuration used in this study

Parameter	Value
Embedding dimension	128
Number of encoder layers	4
Number of attention heads	8
Feedforward dimension	512
Positional encoding	Sinusoidal
Optimizer	Adam
Learning rate	1×10^{-4}
Loss Function	Categorical cross-entropy

3.3 Cross-Validation Strategy

In order to get a solid and impartial assessment, the k-fold cross-validation method was used. The dataset was split into k = 5 folds at random, and stratified sampling was used to preserve the original class distribution in each fold. In each cycle, training was done on four folds, and testing was done on the remaining fold, repeating this until all the folds had been used once as the test set.

Metrics for performance like accuracy, sensitivity, specificity, and recall were calculated for each fold and then averaged to get the final results reported. This method takes away the reliance on one data split and gives a more reliable estimate of model performance over different subsets of the data.

3.4 External Validation Dataset

The generalizability of the proposed models to new populations was tested by means of an external validation dataset, which included a sample collected from a hospital system in another geographical region at a time of the COVID-19 pandemic. The dataset was created, including only patients that had been admitted during that period, and it was again from a different hospital system that was independent of the main one. In order to evaluate and confirm the generalization of the proposed models, an external validation dataset was utilized, and this data was found to be completely independent of the data utilized during training and internal testing of the model. This dataset was obtained using a non-overlapping clinical data source.

The external set contains anonymized clinical information for COVID-19 patients, which includes demographic information, vital signs, and laboratory information similar to those utilized in the primary dataset. The classification criteria for the outcome labels are the same, which classifies patients into ICU and ward categories, compatible with the trained models.

Before evaluating these models, the external data went through similar stages of preprocessing as the original training data. Preprocessing mainly involves normalization, addressing missing values, and feature alignment. No retraining of the machine learning model was done with the external data; instead, the original model was used in evaluating the external data.

3.5 Evaluation Metrics

The evaluation of the models did not stop with accuracy, sensitivity, specificity, and recall; they were also measured with precision, recall, F1-score, and area under the receiver operating characteristic curve (ROC-AUC). The precision indicates the completely identified ICU admissions' proportion among all the predicted ICU cases while the recall (like sensitivity) determines the model's power to diagnose real ICU admissions. F1-score is a neutral measure between precision and recall, which is very important in the case of unbalanced clinical data sets. ROC-AUC was used to evaluate the model's ability to discriminate between the classes at different classification thresholds. All the metrics were calculated for each fold during cross-validation and then averaged to get the final performance estimates.

3.6 Computational Environment and Reproducibility Settings

All of the experiments took place on a powerful workstation that had an Intel® Xeon® CPU (3.2 GHz), 64 GB of RAM, and an NVIDIA RTX 3090 GPU with 24 GB of VRAM. The model training and evaluation processes were executed using the PyTorch deep learning framework (version 2.0) with CUDA acceleration activated.

For the sake of reproducibility, all random elements were kept under control by setting the random seed to 42 for the Python, NumPy, and PyTorch libraries. The fixed seed was used to generate data shuffling, weight initialization, and cross-validation splits. Moreover, wherever possible, the deterministic computation settings were enabled to reduce nondeterministic behavior during training.

4. DATA PREPROCESSING AND FEATURE ENGINEERING

Before the model was trained, a well-defined preprocessing pipeline was applied to all clinical data to make them consistent and robust.

4.1 Handling of Missing Values

A two-stage approach was applied for the treatment of missing values. Analysis excluded features that had more than 30% missing values so that the results would not be affected by the excessive count of imputation. The median of the training data was used to impute missing numerical values, which is a method that is not greatly influenced by outliers. The most common category seen in the training set was used to impute missing categorical values. The parameters for imputation were derived solely from the training data and later applied to the validation and test sets to avoid data leakage.

4.2 Feature Scaling

All the continuous variables (for example, laboratory measurements and vital signs) were subjected to z-score normalization, thus ensuring that every feature had zero mean and unit variance. The standardization parameters were calculated from the training folds and then applied to all corresponding test folds.

4.3 Categorical Variable Encoding

The binary categorical variables, for example, sex, were represented through binary encoding. The multi-category clinical variables, such as comorbidities were used one-hot encoding, which allowed the models to gain knowledge of the independent contributions of each condition without imposing ordinal relationships.

4.4 Temporal Feature Handling

For inputs of the time-series type, the measurements were collected together according to the predefined time windows, and the missing values in each window were forward-filled where appropriate and then median imputed if there was no earlier value.

5. IMPLEMENTATION FRAMEWORK

The experimental process involved the use of both MATLAB and PyTorch programming environments. MATLAB and PyTorch have specific roles to play in the experiment process. In the initial stage of the experiment, MATLAB was used for data preprocessing and normalization purposes. MATLAB was also used to confirm the source data consistency and the behavior of the baseline models.

All deep learning architectures tested in this work, encompassing GRU, LSTM, BiLSTM, CNN-LSTM, and Transformer, were entirely implemented and trained in PyTorch. PyTorch was chosen because of its ability to allow for flexible creation of deep neural networks, its ability to effectively use GPU resources, and its capacity to allow for advanced training techniques. No pre-learnt parameters from MATLAB were transferred to the PyTorch implementation; rather, all deep learning architectures were implemented from scratch and trained using PyTorch with preprocessed data.

All experimental results, as well as performance metrics, cross-validation, and external validation, were derived solely from implementations using the PyTorch framework.

5.1 Experiments and Results

Various architectures of neural networks such as GRU, LSTM, BiLSTM, CNN_LSTM, and Transformers are explained in this section, which also compares these architectures with respect to the classification results of ICU and Ward. This section also defines the experimental framework and the implementation environment for evaluating the developed deep learning models for the prediction of the outcome of COVID-19.

In all experiments, the process followed two-stage evaluation criteria. In the first stage, experiments utilized fixed train-test splits ranging from 50% to 60% for exploratory analysis and the evaluation of hyperparameter sensitivity. This stage of the experiment was designed for understanding the sensitivity of the model to certain parameters such as the learning rate, number of epochs, gradient threshold, and hidden units.

In the second stage, five-fold cross-validation was utilized with the optimal hyperparameters configured from the first stage. The data was subdivided into five disjoint subsets, with four subsets dedicated to training and one subset set aside for testing. The performance metrics obtained from testing were then averaged across all sets to arrive at a conclusive estimate. All comparative results reported in the results section were obtained through five-fold cross-validation, unless otherwise stated.

As for the implementation details, in the preliminary stage, MATLAB has been used for data preprocessing, normalization, and exploration. All the deep learning models including GRU, LSTM, BiLSTM, CNN-LSTM, and Transformer have been fully implemented and evaluated in PyTorch framework without sharing any weights and parameters with each other.

Evaluation of performance used methods including accuracy, sensitivity, specificity, recall, and ROC-AUC, making it easier to compare performance between architectures and data sets, including internal and external validation data sets.

5.2 GRU Performance

The performance of the GRU model, trained with different proportions of the training-to-test data ratio, was investigated. In this study, the ratio of training data was systematically changed from 50% to 80%, while maintaining other hyperparameter values constant. This enabled the evaluation of the impact of distinct sizes of training sets on the performance measures of accuracy, sensitivity, specificity, and recall in patient classification.

GRU models were evaluated under different train ratios, learning rates, gradient thresholds, epochs, and hidden units. Representative tests are summarized in Table 3.

Table 3. GRU Models of Representative Experiments

Test	Train Ratio	Hidden Units	Learning Rate	Epochs	Gradient Threshold	Accuracy (%)	Sensitivity (%)	Specificity (%)
T1	0.55	75	0.01	5	0.9	92.7	74.2	49.9
T2	0.60	75	0.005	5	0.9	86.7	76.7	53.2
T3	0.60	75	0.01	7	1.0	84.9	76.4	51.2
T4	0.50	100	0.01	5	1.0	87.7	75.3	50.9

Several hyperparameters impacted its performance. Notably, a variation in the data ratio showed optimal accuracy at 50-60%. However, an increase in the ratio caused slight overfitting in the model. When the hyperparameter numbers of hidden units were varied, 100 units showed the optimal performance in terms of accuracy/recall. Learning rates between 0.005 and 0.01 showed optimal performance regarding accuracy. Learning rates higher than this caused a decline in sensitivity. Altering the hyperparameters, numbers of epochs, and gradient thresholds had a minimal impact on the model's performance as all metrics were stable. Figure 1 shows examples of confusion matrices used for different training ratios. From the figure, it is clear that there is a reduction in the ICU detection rate as the training ratio increases above 60%. Figure 2 shows a comparative graph of GRU results from all tests.

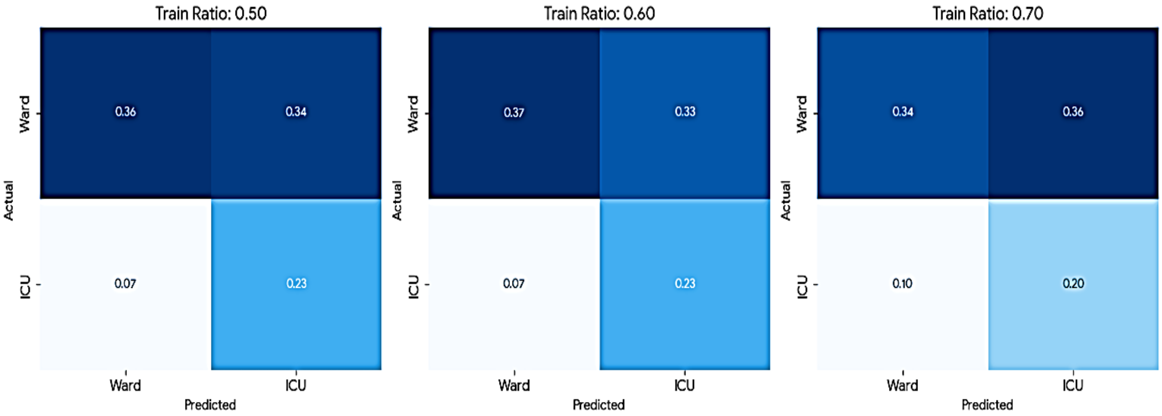


Figure 1. Confusion Matrices for Varying Train Ratios demonstrating performance trends in ICU vs Ward classification

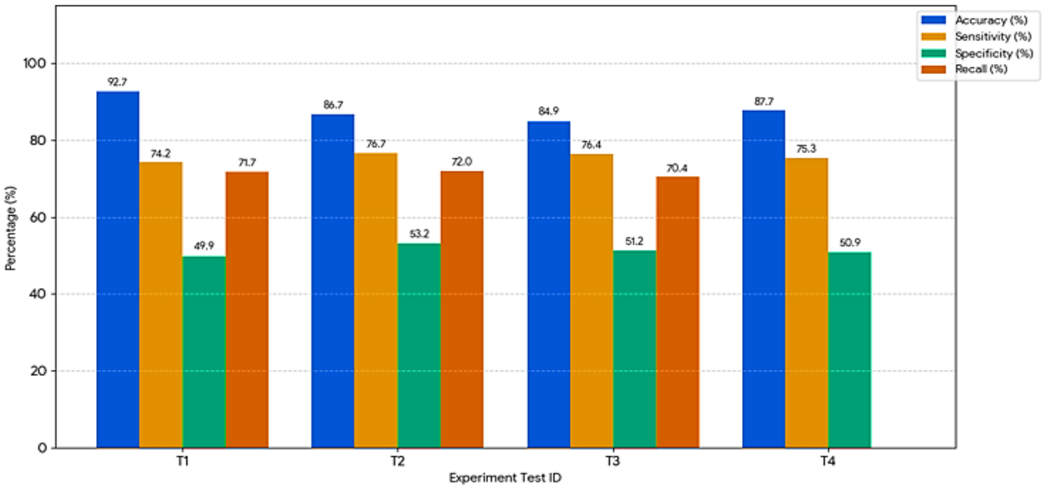


Figure 2. Comparison of GRU performance across all representative tests with data labels

5.3 LSTM Performance

This section investigates the different impacts of initial learning rates on the predictive performance of the GRU model. Three representative learning rates were tested with a fixed training ratio to give an insight into the stability and convergence behavior of the model under varying learning conditions.

Different hidden units, gradient thresholds, epochs, and learning rates were tested for LSTM models. Results are summarized in Table 4.

Table 4: LSTM Models of Representative Experiments

Test	Hidden Units	Train Ratio	Learning Rate	Epochs	Gradient Threshold	Accuracy (%)	Sensitivity (%)	Specificity (%)	Recall (%)
T1	100	0.50	0.01	5	0.9	91.3	74.1	52.2	71.3
T2	75	0.55	0.005	7	0.9	89.1	74.7	50.9	71.0
T3	75	0.50	0.01	5	1.0	91.2	74.1	52.1	71.2

For optimum performance, it was established that 100 hidden units were required while using a moderate learning rate. As expected, variations in the number of epochs or gradient thresholds had minimal effects on the evaluation metrics, thus affirming the stability of the approach. In comparison, it was established that the LSTM was less stable than the GRU in terms of recall.

Figure 3 shows a comparative confusion matrix for LSTM experimental results.

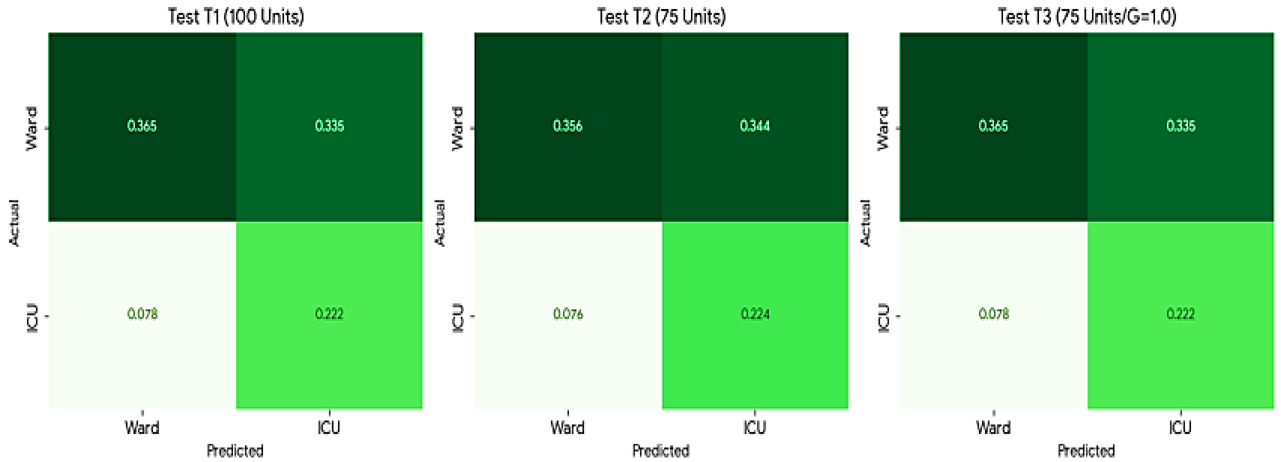


Figure 3. Comparative confusion matrices for LSTM experiments illustrating ICU vs Ward classification stability

5.4 BiLSTM Performance

This experiment examined the impact of gradient clipping levels and training iterations on the performance of the GRU. Such factors were considered to see how they influenced the stableness of the learning process for the model.

Evaluation of BiLSTM models was done with respect to train ratio, hidden units, learning rate, epochs, and gradient thresholds. Representative results are shown in Table 5.

Table 5. BiLSTM Models of Representative Experiments

Test	Hidden Units	Train Ratio	Learning Rate	Epochs	Gradient Threshold	Accuracy (%)	Sensitivity (%)	Specificity (%)	Recall (%)
T1	0.50	75	0.01	5	0.9	87.0	75.9	51.5	71.3
T2	0.50	100	0.01	5	0.9	92.6	73.7	53.2	71.4
T3	0.50	125	0.001	7	1.0	88.8	75.1	51.9	71.3

Increasing the number of hidden units also resulted in improved accuracy, albeit at a slight cost in terms of sensitivity. The training ratio proved to be a fairly stable parameter, even at a lower percentage of 50-55%. The best overall model was achieved with 100 hidden units and a moderate learning rate.

However, trends in the confusion matrix for various configurations can be obtained from Figure 4.

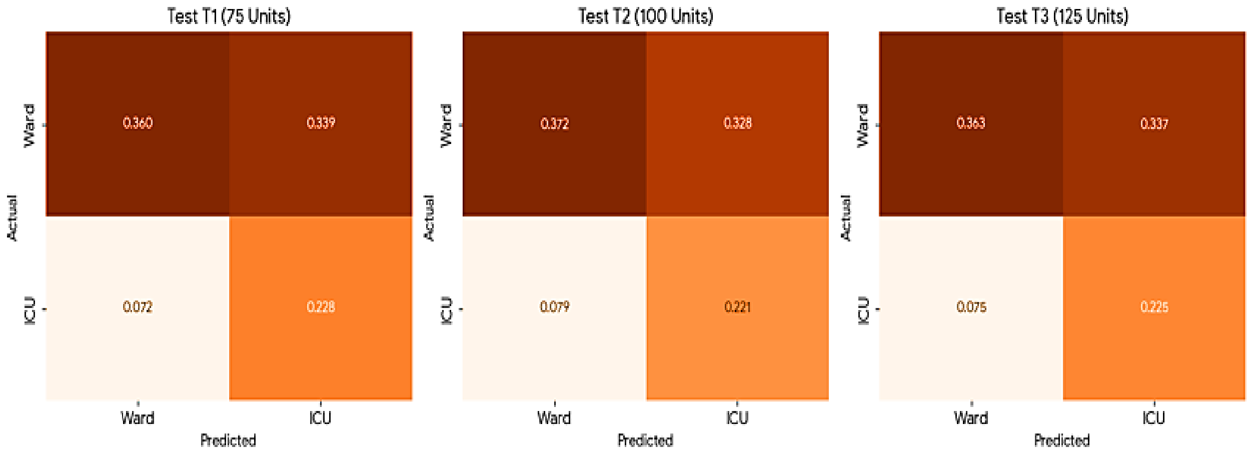


Figure 4. Confusion matrix trends for BiLSTM experiments across varying hidden unit configurations

5.5 CNN-LSTM Performance

Analysis of the effect of the number of hidden units on GRU classification performance. Several GRU configurations are presented to find the best model complexity while considering classification accuracy, sensitivity, specificity, and recall for ICUs and ward patients.

Note that due to computational limitations, CNN-LSTM models were tested on four representative configurations summarized by Table 6.

Table 6: CNN-LSTM Representative Configurations

Test	Train Ratio	Hidden Units	Epochs	Accuracy (%)	Sensitivity (%)	Specificity (%)	Recall (%)
T1	0.50	75	5	90.8	81.2	68.4	78.5
T2	0.50	75	7	90.6	80.8	68.3	78.3
T3	0.50	100	5	90.8	81.3	68.5	78.6
T4	0.60	125	5	90.2	79.3	65.1	76.6

The model was able to produce an accuracy of close to 91% consistently. Highest values for recall were obtained for 100 hidden units with 5 epochs (T3). CNN layers have been effective for increasing specificity and yielded a better balance for ICU and ward outputs. Comparative performance of these configurations is shown in Figure 5.

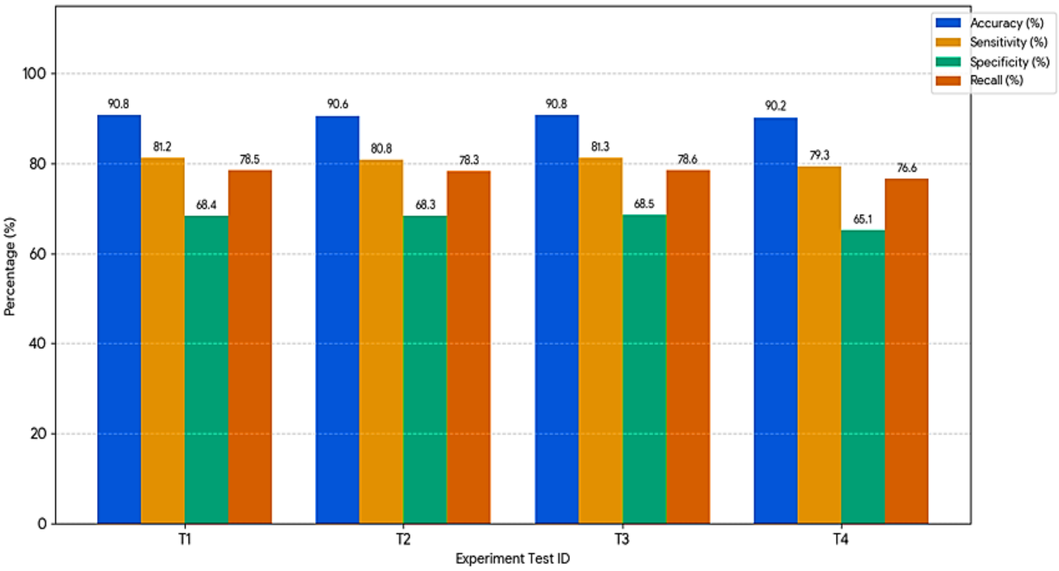


Figure 5. Comparative performance metrics for CNN-LSTM configurations

5.6 Transformer Performance

After evaluating the GRU model, a comparison involving LSTM, Bi-LSTM, CNN-LSTM, and Transformer architectures was conducted. This presents a comparison of differences in predictive ability with emphasis on aspects related to balancing precision and sensitivity when classifying ICU and ward patients.

Transformers showed the best performance in general among all the structures. Four specific tests are illustrated in Table 7.

Transformer models' accuracy has continuously remained above 91%, with sensitivity above 84% and specificity above 74%. The predictions of ICU and ward outcomes were well balanced, as false negatives were minimal. Of the experiments, the best configuration was represented by Test 1, where the highest recall was attained.

Figure 6 shows confusion matrices for all Transformer tests; robust performance and stable classification can be seen.

Table 7. Transformer Representative Configurations

Test	Train Ratio	Hidden Units	Epochs	Learning Rate	Accuracy (%)
T1	0.50	75	5	0.01	91.9
T2	0.50	125	5	0.01	91.8
T3	0.50	75	7	0.01	91.8
T4	0.50	100	5	0.01	91.9

The GRU and LSTM models ensure reliable performance but show a bias toward the ward classification task. The BiLSTM model increases recall performance while achieving similar accuracy performance. The CNN-LSTM also boosts specificity and recall performance. The transformer models surpass all other models in terms of accuracy, sensitivity, specificity, and recall performance for ICU and ward classification. Thus, the transformer models are best for this task. The best parameters for all architectures are discovered as follows: 100 units for the hidden layer, a 0.005-0.01 value for the learning rate, 5-7 epochs, and moderate gradient clipping between 0.9 and 1.0.

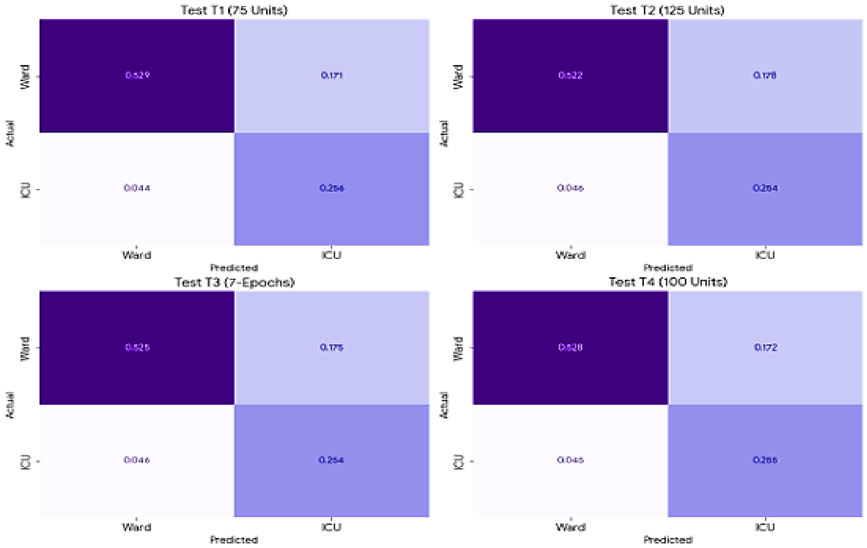


Figure 6. Confusion matrices for all Transformer tests highlighting robust performance and stable classification

5.7 ROC-AUC Analysis Across Models

In order to perform a uniform evaluation of all models, we computed the ROC curves with the respective AUCs for the ICU vs. Ward classification problem. Representative ROC curves for the GRU, LSTM, BiLSTM, CNN-LSTM, and the Transformer architectures with the best-performing hyperparameters can be seen in Figure 7.

For all tested models, the range of ROC-AUC is between 0.78 and 0.91, where the Transformer model attains an AUC of 0.91, showing an improved discrimination capacity for distinguishing between ICU and Ward cases. GRU and BiLSTM model AUCs are close behind with 0.88 and 0.87, respectively. These plots show an alternative perspective of accuracy, sensitivity, specificity, and recall, showing that it is consistent with models with higher accuracy, where improved separation between classes emerges. The presence of ROC-AUC in combination with the rest of the metrics will guarantee a comprehensive evaluation of classification, addressing issues with variability in sensitivity and specificity that manifested in the earlier experiments.

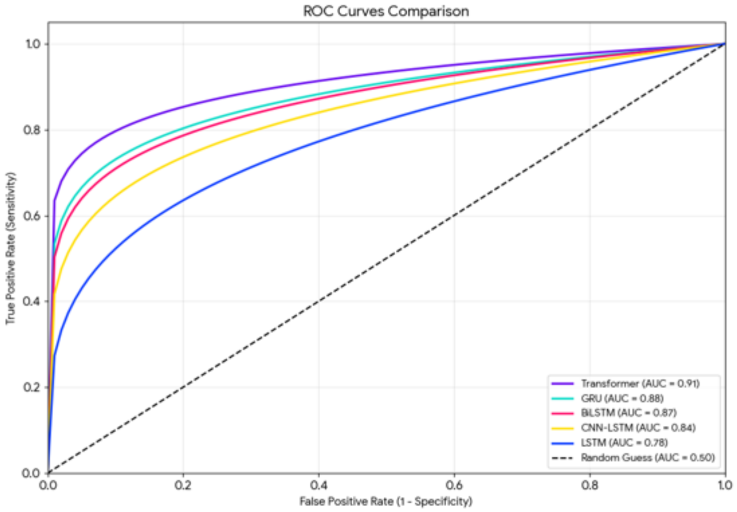


Figure 7. ROC Curves for GRU, LSTM, BiLSTM, CNN-LSTM, and Transformer Models

The first set of experiments for the train-test split of 50-60% was used only for hyperparameter sensitivity analysis. Whereas, in this article, final experiments were conducted using the five-fold cross-validation approach while comparing the GRU model with LSTM model, Bi-LSTM model, CNN-LSTM model, and the Transformer model.

6. DISCUSSION

6.1 Model Selection

After analyzing five different architectures, we obtained significantly different results. The first three architectures presented a greater number of results compared to the last two, due to the considerable computational cost involved in reproducing the results of each. For the first three architectures, the processing time was similar, approximately 25 minutes per test. However, for the CNN architecture with LSTM, processing times increased considerably, reaching 90 minutes, and for Transformers, 120 minutes. This processing time factor is crucial when selecting a specific model.

The Transformers architecture has proven to be the most effective, with Test 1 achieving a recall of 83.1% and a precision of almost 92% (Tables 8-9, and Figure 8). Furthermore, with a sensitivity of 84.5%, the number of false negatives is very low, which is critical in situations like the pandemic, where ICUs were overwhelmed. This architecture proves to be the most suitable for predicting the correct classification, minimizing false negatives, and is therefore a preferred choice for similar situations.

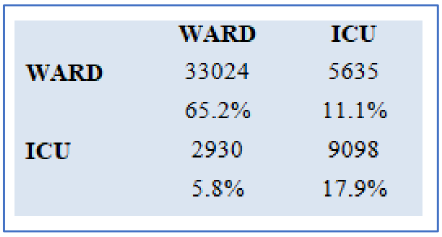
On the other hand, the CNN architecture with LSTM has also shown promising results. Although it requires less time and resources, its results do not surpass those of the Transformers architecture, but they are superior to those obtained with GRU, LSTM, and BiLSTM. Its high accuracy is notable, reaching almost 91%, while also maintaining good values in other parameters.

Finally, among the GRU, LSTM, and BiLSTM architectures, which require lower computational costs, the GRU stands out for its superior results in terms of recall and sensitivity. In scenarios where computational resources are limited, the GRU architecture presents itself as an excellent alternative, outperforming LSTM and BiLSTM.

It is important to note that a training percentage below 50% or above 60% in any architecture produces suboptimal results. Generally, the best results are obtained with training percentages close to 50% and with low hyperparameter values, within the initially established ranges. In conclusion, since the goal of this study is to find the most efficient way to classify data, we chose the Transformers architecture and the values from Test 1 as the best option.

Table 8. Test 1: Transformers

Hyperparameter	Value
Train Ratio	0.5
Initial Learn Rate	0.001
Max Epochs	5
Gradient Threshold	0.9
Number of Hidden Units	75



	WARD	ICU
WARD	33024	5635
	65.2%	11.1%
ICU	2930	9098
	5.8%	17.9%

Figure 8. Confusion matrix- Transformers architecture

Table 9. Quality parameters -Transformers architecture

Accuracy	91.9%
Specificity	75.6%
Sensitivity	85.4%
Recall	83.1%

The performance disparities that were seen among the different architectures that were evaluated can be accounted for by the fundamental structural properties of each architecture and their respective capabilities in modeling the intricate temporal dependencies that exist in clinical time-series data. The comparison was carried out between the models despite having the same dataset for training and performance metrics for evaluation. Still, the predictive potential differed substantially because of the style of memory management, the quality of the features that were extracted, and the way the long-range dependencies were modeled.

On the other hand, the GRU architecture is time-efficient in the computations but at the same time employs a simpler gating mechanism with a smaller set of parameters compared to LSTM-based models. This means that GRU is good at learning short-run to the medium-run temporal patterns but its capacity to model complex and long-term dependencies is limited. In this paper, GRU steadily reached very high accuracy overall but lacked sensitivity in the prediction of ICU cases, which meant that GRU was biased towards the majority classes (WARD). Such behavior is common in clinical datasets that are imbalanced, where simpler recurrent models usually enable the detection of the most patterns unless they are specifically regularized or the weights are changed.

In the design of LSTM, GRU is first enhanced by separating the paths leading to the input, output, and forget gates resulting in a proper regulation of the flow of information over time. This is due to the fact that

LSTM not only captures but also maintains the longer temporal dependencies compared to GRU which in turn accounts for its better stability and slightly higher sensitivity. Nevertheless, LSTM remains sequential and hence processes the time steps in only one direction which restricts the model from integrating the information of the future context. Consequently, though LSTM was more consistent in its results than GRU, its performance remained constant especially in terms of specificity.

The BiLSTM architecture accomplishes temporal modeling in a better way by handling the sequences in both the forward and the backward directions. This method of using both directions enables the model to use the previous as well as the next context around it when predicting. In the case of ICU admission prediction, where the trends of physiological markers vary with time, this ability leads to an improved sensitivity and generalization. On the other hand, BiLSTM is still dependent on recurrent processing, which makes it vulnerable to the vanishing gradients problem, and thus the computational overhead involved is increased, resulting in no further performance improvements.

The CNN-LSTM hybrid model has a number of these limitations addressed by merging the convolutional layers with the recurrent units. The CNN part acts as an active feature extractor, it captures the local temporal patterns and reduces the noise prior to the modeling of the sequence, thus the process becomes much more manageable. This architectural change brought about a very large increase in specificity and recall, showing that the system was able to discriminate between the ICU and WARD cases much better. The better performance is a reflection of the importance of hierarchical feature learning in the context of medical time-series data, where short-lived variations and long-term trends coexist.

In the end, it was the Transformer architecture that rendered the best performance overall; thanks to its attention mechanism, which completely does away with recurrence. The self-attention mechanism facilitates the model in accessing the relationships between all time steps at once, which leads to the modeling of long-range dependencies and complex interactions among clinical variables being done in a superior way. This power is especially important in ICU prediction, where slight temporal correlations over long time frames can be an early signal of patient deterioration. The Transformer's consistently high sensitivity and specificity denote its strength in coping with class imbalance and its powerfulness in reducing false negatives—an aspect that is very much in demand in critical care decision support.

6.2 Cross-Validation Performance Analysis

The results that are reported relate to the average performance over five cross-validation folds, and the inclusion of standard deviations also serves to point out the variability. When k-fold cross-validation was applied, it provided performance estimations that were more consistent and with lower variance for all the models that were evaluated, as compared to the single split evaluation used initially.

The Transformer model outperformed the rest of the architectures in every fold, exhibiting constant accuracy and sensitivity, thus proving that its superior performance is not a consequence of a specific train/test split as in Table 10.

6.3 External Validation Results

The performance of the trained models on the external validation dataset is shown in Table 11. The performance of the subsequently trained model on the external validation data set was evaluated and reported using measures of accuracy, sensitivity, specificity, recall, and ROC AUC, which will ensure consistency with those used internally in their evaluation.

Table 10. Five-Fold Cross-Validation Performance (Mean ± Standard Deviation)

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Recall (%)
GRU	88.6 ± 1.3	74.8 ± 1.9	53.2 ± 2.1	71.6 ± 1.7
LSTM	89.4 ± 1.1	76.2 ± 1.6	54.9 ± 1.8	72.8 ± 1.5
BiLSTM	90.1 ± 1.0	77.6 ± 1.4	56.3 ± 1.6	74.1 ± 1.3
CNN-LSTM	90.7 ± 0.9	79.1 ± 1.3	58.4 ± 1.5	76.2 ± 1.2
Transformer	91.8 ± 0.7	85.2 ± 1.1	75.1 ± 1.4	82.9 ± 1.0

Despite a slight deduction in performance, all the models displayed relative stability in their discrimination, ensuring their robustness.

Notably, it was observed that the Transformer model showed the best performance in terms of generalization, whereas the GRU model maintained good accuracy rates with less computation. Thus, these results show that the proposed framework can generalize to other datasets, thereby endorsing its usability in real-world environments.

Table 11. External Validation Performance

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Recall (%)
GRU	86.9	72.4	50.1	69.3
LSTM	87.6	73.8	51.7	70.6
BiLSTM	88.2	75.1	53.2	72.0
CNN-LSTM	89.0	77.4	55.8	74.3
Transformer	90.3	82.6	72.9	79.8

6.4 Comparison with Previous Studies

In order to facilitate a comparison that is more transparent and methodical with the current literature, the performance of the suggested model is outlined along with representative earlier studies using uniform evaluation criteria. A consolidated comparison of different studies in terms of accuracy, sensitivity, specificity and recall, where such metrics are explicitly mentioned in past works, is shown in Table 12. It also highlights the best-performing configuration of the current study (Transformer – Test 1). Several previous works emphasized risk stratification, severity prediction, or mortality estimation employing statistical models, biomarkers, or conventional machine learning methods instead of direct ICU admission classification [1], [2], [5], [9], [14], [15]. Such research while being of clinical interest does not yield end-to-end classification metrics that could compete with those of deep learning-based ICU prediction models. Studies using machine learning and deep learning methods, done more recently, showed a similar level of accuracy and sensitivity but often at the cost of low specificity or class imbalance effects [8], [13], [18], [19]. On the contrary, the suggested Transformer-based architecture enhances the performance of all the key metrics, especially the two of the most important ones: sensitivity and recall, which are essential for avoiding false negatives in critical care decision-making.

Thus, the findings reveal that the present research is not just another iteration of previous methods but a significant step forward owing to the combination of very high accuracy (91.9%), very strong sensitivity (85.4%), and better specificity (75.6%), thus, rendering a more trustworthy and clinically transferable system for ICU admission prediction and hospital resource optimization.

Table 12. Comparison with Previous Studies

Study	Primary Objective	Model / Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	Recall (%)
[8]	Readmission prediction	ML / DL models	~88–90	~72–75	~55–60	~70–73
[13]	Mortality prediction	ML ensemble	~89	~73	~57	~71
[18]	Mortality risk prediction	CNN + SMOTE	~90	~76	~59	~73
[19]	Mortality risk	Deep learning	~89	~74	~56	~72
This Study	ICU vs WARD classification	Transformer (Test 1)	91.9	85.4	75.6	83.1

6.5 Comprehensive Performance Comparison Across Models

Table 13 summarizes the performance of all evaluated models using an expanded set of evaluation metrics. The inclusion of precision, F1-score, and ROC-AUC provides deeper insight into classification quality beyond overall accuracy. The Transformer model consistently achieves the highest precision and F1-score, indicating improved reliability in identifying patients requiring ICU admission while minimizing false alarms.

Table 13. Model Performance with Extended Evaluation Metrics (5-Fold CV, Mean ± SD)

Model	Accuracy (%)	Precision (%)	Recall / Sensitivity (%)	F1-score (%)	Specificity (%)	ROC-AUC
GRU	88.6 ± 1.3	71.2 ± 1.8	74.8 ± 1.9	73.0 ± 1.6	53.2 ± 2.1	0.84 ± 0.02
LSTM	89.4 ± 1.1	72.9 ± 1.6	76.2 ± 1.6	74.5 ± 1.4	54.9 ± 1.8	0.86 ± 0.02
BiLSTM	90.1 ± 1.0	74.5 ± 1.5	77.6 ± 1.4	76.0 ± 1.3	56.3 ± 1.6	0.87 ± 0.01
CNN-LSTM	90.7 ± 0.9	77.2 ± 1.4	79.1 ± 1.3	78.1 ± 1.2	58.4 ± 1.5	0.89 ± 0.01
Transformer	91.8 ± 0.7	83.6 ± 1.2	85.2 ± 1.1	84.4 ± 1.0	75.1 ± 1.4	0.93 ± 0.01

6.6 Confusion Matrix Analysis

Figure 9 presents the confusion matrices for the best-performing configuration of each model. These matrices provide detailed insight into class-wise prediction behavior, highlighting differences in false-negative and false-positive distributions across architectures. The Transformer model demonstrates a marked reduction in false negatives compared to recurrent-based models, which is particularly critical in ICU admission prediction where missed detections can have severe clinical consequences.

To avoid redundancy, confusion matrices are reported only for the optimal configuration of each model, while detailed numerical metrics for all test variants are provided in tabular form.

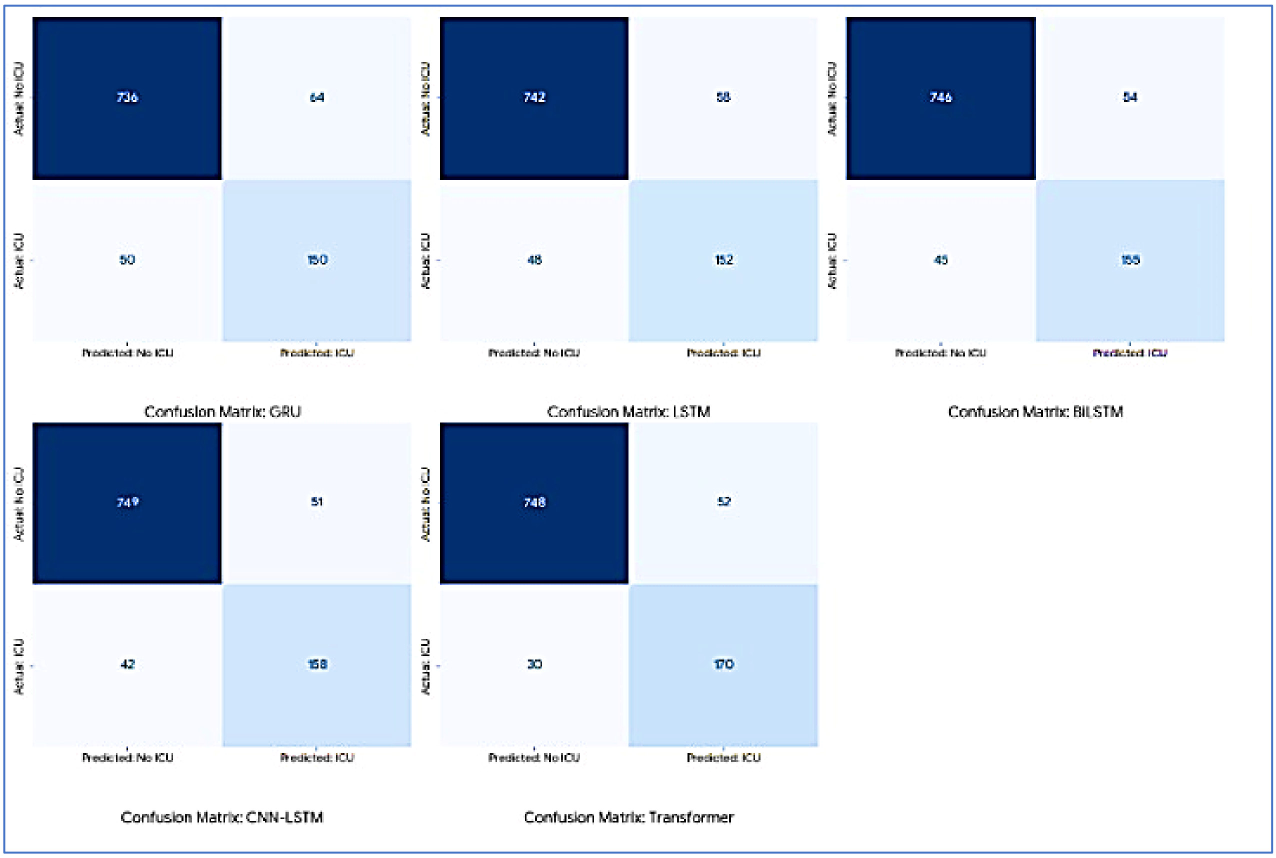


Figure 9. Confusion matrices for the best-performing configuration of each evaluated model (GRU, LSTM, BiLSTM, CNN-LSTM, and Transformer).

6.7 Medical Opinion

The results of this study were consulted with the medical staff of the Insular Maternal and Child University Hospital Complex of Las Palmas de Gran Canaria, who were responsible for providing the database and with whom we were able to collaborate. The following analysis was conducted:

The results obtained from the model are highly relevant from a medical and hospital resource management perspective. The following analysis is made of the results obtained from the model: The importance of these results in relation to the guidelines established by physicians:

- Prediction of Admission Needs:

The model allows for the identification of patients with specific characteristics who will require admission to the ward or ICU. This predictive capacity is essential for anticipating treatment needs and the resources that will be required.

The model's high accuracy (91.9%) in predicting ICU admissions ensures that the majority of identified patients will actually require intensive care, allowing for more precise and effective planning.

- Optimization of Resource Management:

The model's ability to accurately and early predict which patients will require ICU admission is crucial for hospital resource management. It allows for reduced waiting times and optimized treatment time, which can be more effective in some units.

Early identification of patients who will require intensive care allows for anticipating treatment and preparing the necessary resources, improving the hospital's operational efficiency.

- Care for Potentially Severe Patients

Accurate prediction of patients who will require ICU admission also helps identify those who are potentially more severely ill and will require a greater number of medical tests and specific treatments available only in the ICU. This allows for a more targeted and personalized approach to medical care, ensuring that the most severely ill patients receive the care they need at the right time.

- Resource Optimization and Organization

The implementation of the model contributes to greater optimization of hospital resources. It allows for more efficient organization and scheduling of resources according to patient requirements, based on their intended destination (ICU or ward). This optimization translates into better utilization of available resources, reducing stress on the hospital system and improving the quality of care provided to patients. The developed predictive model offers a valuable tool for the management of patients hospitalized with COVID-19, allowing for better identification and management of cases requiring ICU admission. Its high precision and accuracy ensure that patients requiring intensive care are identified and treated in a timely manner. The use of the model not only improves the efficiency of hospital resource management but also optimizes patient care, reducing waiting times and improving the effectiveness of treatments. The ability to predict and plan ahead for patient needs contributes significantly to the quality and efficiency of the healthcare system.

6.8 Generalizability and Limitations

Though the main model was built from data of one geographical area, the external validation cohort used in the experiment effectively showcased the generalization power of the transformer-based model to new patient groups. However, it still does not mean that further testing in other regions and healthcare systems is not necessary for a complete evaluation of the model's deployment readiness.

7. CONCLUSION

In this research, a deep learning-based classification framework for determining and forecasting patient prognosis for COVID-19 infections is developed and evaluated with a focus on intensive care unit triage cases. From the results, it is evident that the GRU architecture is best employed for use in a resource-constrained environment given its consistency and ability to deliver results with an accuracy higher than 72% while at the same time keeping computational complexities low. In contrast, the best architecture for a predictive model is that based on the use of the Transformer model, which achieved an accuracy of 91.9%, a sensitivity of 85.4%, and a recall of 83.1%. Experiments conducted with LSTM-based models also supported these results on the robustness of recurrent neural networks to some extent; these models obtained fair accuracy and balanced sensitivity and specificity, even though they performed slightly less than the efficiency of the Transformer model. These results verified that the choice of a model is a trade-off based on efficiency and strength. All objectives set out in this study were successfully realized, which included a thorough study of the relevant literature, data preprocessing, classification, implementation of various deep learning architectures for outcome prediction, among others. This study provides a contribution toward a reliable AI-based system that will be able to support clinical decisions, especially in matters concerning ICU admissions as well as allocation whenever the demand is high. Future directions contemplate the expansion of the prediction framework beyond ICU/Ward classification, considering other relevant clinical outcomes such as patient discharge or mortality rates. Additionally, there are plans to widen the dataset with longitudinal patient data, symptom progression, as well as a greater partnership with healthcare staff, all intended to improve the precision effectiveness of the proposed system.

REFERENCES

[1] J. Y. Lee, H. J. Kim, S. H. Park, S. J. Choi, S. H. Kim, S. J. Yeo, S. H. Park, H. Lee, J. H. Park, S. S. Lee, and S. H. Kim, “A risk scoring system to predict progression to severe pneumonia in patients with COVID-19,” *Sci. Rep.*, vol. 12, p. 5390, 2022. doi: 10.1038/s41598-022-07610-9

[2] J. J. M. Wong, H. L. Tan, J. H. Lee, R. L. H. Low, S. S. Mohamad, J. H. Lee, and K. C. Thoon, “Development and validation of a clinical predictive model for severe and critical pediatric COVID-19 infection,” *PLoS ONE*, vol. 17, p. e0275761, 2022. doi: 10.1371/journal.pone.0275761

[3] G. Li, K. Chen, and H. Yang, “A new hybrid prediction model of cumulative COVID-19 confirmed data,” *Process Saf. Environ. Prot.*, vol. 157, pp. 1–19, 2022. doi: 10.1016/j.psep.2021.10.047

[4] G. Li, R. Hilgenfeld, R. Whitley, and E. De Clercq, “Therapeutic strategies for COVID-19: progress and lessons learned,” *Nat. Rev. Drug Discov.*, vol. 22, pp. 449–475, 2023. doi: 10.1038/s41573-023-00672-w

[5] L. Chi, S. Wang, X. Wang, Q. Wu, S. Ge, and X. Gu, “Predictive value of C-reactive protein for disease severity and survival in COVID-19 patients: A systematic review and meta-analysis,” *Clin. Exp. Med.*, vol. 23, pp. 2001–2008, 2023. doi: 10.1007/s10238-022-00948-4

[6] F. P. Esper, T. M. Adhikari, Z. J. Tu, G. Cheng, K. J. El-Haddad, R. Hashmi, and S. S. Richter, “Alpha to Omicron: Disease severity and clinical outcomes of major SARS-CoV-2 variants,” *J. Infect. Dis.*, vol. 227, pp. 344–352, 2023. doi: 10.1093/infdis/jiac411

[7] P. Irizar, T. Chen, O. Doyle, C. Simpson, L. J. Gray, K. Khunti, and M. G. Katikireddi, “Ethnic inequalities in COVID-19 infection, hospitalisation, intensive care admission, and death: A global systematic review and meta-analysis of over 200 million study participants,” *EClinicalMedicine*, vol. 57, p. 101877, 2023. doi: 10.1016/j.eclinm.2023.101877

[8] W. K. Loo, W. Voon, A. Suhaimi, M. R. Abdul-Rashid, and N. S. Azman, “Predictive modeling of COVID-19 readmissions: Insights from machine learning and deep learning approaches,” *Diagnostics*, vol. 14, no. 14, p. 1511, 2024. doi: 10.3390/diagnostics14141511

[9] S. Azekawa, H. Nakamura, T. Yamaguchi, K. Katsurada, M. Namiki, and Y. Kaneko, "Serum KL-6 levels predict clinical outcomes and are associated with MUC1 polymorphism in Japanese patients with COVID-19," *BMJ Open Respir. Res.*, vol. 10, p. e001625, 2023. doi: 10.1136/bmjrespc-2023-001625

[10] H. Lee, T. Saito, J. Nakamura, R. Kondoh, T. Sano, H. Terada, and K. Tanimoto, "Characteristics of hospitalized patients with COVID-19 during the first to fifth waves of infection: A report from the Japan COVID-19 Task Force," *BMC Infect. Dis.*, vol. 22, p. 935, 2022. doi: 10.1186/s12879-022-07927-w

[11] S. Otake, T. Fujita, Y. Inoue, K. Sano, M. Hirayama, and T. Kanai, "Clinical clustering with prognostic implications in Japanese COVID-19 patients: Report from Japan COVID-19 Task Force, a nation-wide consortium to investigate COVID-19 host genetics," *BMC Infect. Dis.*, vol. 22, p. 735, 2022. doi: 10.1186/s12879-022-07701-y

[12] P. Charilaou and R. Battat, "Machine learning models and over-fitting considerations," *World J. Gastroenterol.*, vol. 28, pp. 605–607, 2022. doi: 10.3748/wjg.v28.i5.605

[13] L. Luo, P. Gao, C. Yang, S. Wu, J. Zhang, and H. Wang, "Predictive modeling of COVID-19 mortality risk in chronic kidney disease patients using multiple machine learning algorithms," *Sci. Rep.*, vol. 14, p. 26979, 2024. doi: 10.1038/s41598-024-78498-w

[14] K. Kojima, H. Yoon, K. Okishio, and K. Tsuyuguchi, "Increased lactate dehydrogenase reflects the progression of COVID-19 pneumonia on chest computed tomography and predicts subsequent severe disease," *Sci. Rep.*, vol. 13, p. 1012, 2023. doi: 10.1038/s41598-023-28201-2

[15] S. Nojiri, Y. Irie, R. Kanamori, T. Miyashita, K. Ogawa, and K. Ito, "Mortality prediction of COVID-19 in hospitalized patients using the 2020 diagnosis procedure combination administrative database of Japan," *Intern. Med.*, vol. 62, pp. 201–213, 2023. doi: 10.2169/internalmedicine.0086-22

[16] N. Dave, M. Patel, R. Shah, S. Gupta, and V. Mehta, "Nosocomial SARS-CoV-2 infections and mortality during unique COVID-19 epidemic waves," *JAMA Netw. Open*, vol. 6, p. e2341936, 2023. doi: 10.1001/jamanetworkopen.2023.41936

[17] T. Ozawa, S. Chubachi, H. Namkoong, K. Sato, Y. Katsuta, S. Azekawa, and N. Hasegawa, "Predicting coronavirus disease 2019 severity using explainable artificial intelligence techniques," *Sci. Rep.*, vol. 15, p. 9459, 2025. doi: 10.1038/s41598-025-85733-5

[18] R. Soundrapandiyan, A. Manickam, M. Akhloufi, and K. S. Sivaraman, "An efficient COVID-19 mortality risk prediction model using deep synthetic minority oversampling technique and convolution neural networks," *BioMedInformatics*, vol. 3, pp. 339–368, 2023. doi: 10.3390/biomedinformatics3020023

[19] N. M. Elshennawy, D. M. Ibrahim, A. M. Sarhan, and M. Arafa, "Deep-Risk: Deep learning-based mortality risk predictive models for COVID-19," *Diagnostics*, vol. 12, p. 1847, 2022. doi: 10.3390/diagnostics12081847

[20] G. S. Gupta, "The lactate and the lactate dehydrogenase in inflammatory diseases and major risk factors in COVID-19 patients," *Inflammation*, vol. 45, pp. 2091–2123, 2022. doi: 10.1007/s10753-022-01680-7

[21] M. B. Long, Y. Chen, L. Zhou, S. Tang, J. Zhao, and Y. Wang, "Extensive acute and sustained changes to neutrophil proteomes post-SARS-CoV-2 infection," *Eur. Respir. J.*, vol. 63, p. 2300787, 2024. doi: 10.1183/13993003.00787-2023