International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

PREDICTING HOSPITAL STAY LENGTH USING KNN REGRESSOR OPTIMIZED WITH GRID SEARCH CV: AN EXPLAINABLE MACHINE LEARNING APPROACH

[1] Navadeep Aluvoju, [2] Shaik Zaheer, [3] Keerthana Uppu, [4]M V Anajana Devi

[1,2,3]UG Scholars,[4] Associate Professor

[1,2,3,4,]Department of Computer Science and Engineering(Artificial Intelligence and Machine Learning)

[1,2,3,4,]Guru Nanak Institutions Technical Campus, Hyderabad, Telangana, India

---***---

Abstract

Predicting how long patients will stay in the ICU is truly critical to hospitals from a budgeting and cutting costs perspective. We considered applying a straightforward machine learning technique named K-Nearest Neighbors (KNN) in an effort to forecast ICU stays. The good thing about KNN is that physicians actually get to see how it is forecasted, as opposed to other sophisticated techniques which are a black box .

Our method operates by discovering similar patients for the current patient and predicting based on the duration they were hospitalized. We used an algorithm called GridSearchCV to discover the best parameters for our model. Through trials and combinations, we discovered that our method performs nearly as well as advanced algorithms but with a massive benefit - doctors can simply know why a prediction was made.

The findings indicate that straightforward, easy-to-communicate approaches are just as effective as advanced approaches in medicine. This is relevant because physicians must be confident and familiar with computer predictions before applying them for patient benefits. Our research indicates you do not necessarily require advanced algorithms to generate precise results.

Keywords: ICU length of stay, K-Nearest Neighbors, GridSearchCV, explainable AI, healthcare analytics, predictive modeling

1. Introduction

One of the largest challenges hospitals are facing right now is an estimate of how long they are going to have a person in the ICU. This is very costly to guess wrongly and will cause issues for hospitals and patients alike. If hospitals can't anticipate how long someone will be, they can't prepare. They may not have beds ready for new patients, or they may leave beds open up when they could be operating on other individuals.

Historical methods for estimating hospital stays have not been very accurate. Physicians commonly use their experience and some simple scoring mechanisms, but these are frequently in error. They may estimate a person will be there 3 days, but then the patient is there for a week. This causes actual difficulties for hospital administration and patient care .

Computers have become much more accurate at these predictions lately. There are a few computer programs out there that can go through a lot of patient information and pretty reasonably well predict how long someone's going to stay. The catch is that most of these programs are black boxes - they spit out the answer, but they don't explain to you why they believe that is correct**[1]**.

This is a serious issue for hospitals since doctors ought to be able to see why a computer is going to consider something**[2]**. Doctors should not be told by a computer "this patient will stay 5 days" without explanation. Doctors are trained and experienced heavily, and they must be in a position to discern whether or not computer logic is rational.

Our research attempts to answer this question using a technique known as K-Nearest Neighbors, or simply KNN. The best thing about KNN is that it actually makes sense how it works**[3]**. Rather than using complicated math that no one can decipher, KNN simply looks for patients who are most like the patient being analyzed and takes a look at how long they lingered. It's essentially asking "what did other patients who were exactly like this one do?"

This strategy has a number of strengths. One, it's transparent - physicians are aware who their patient is being matched with by the computer and why it reached its conclusion**[4]**. Two, it's understandable to physicians since physicians already think in terms like this. When a physician looks at a patient, they will typically think about similar cases that they've seen previously. Three, it is flexible and can be used with a broad spectrum of patient data.

Our ultimate aim is to demonstrate that simple and direct mechanisms can function in exactly the same way as complex computer programs in predicting ICU stay.

2. Literature Review

2.1 Traditional Approaches to LOS Prediction

Hospitals have been attempting for decades to forecast how long a patient would remain in the hospital with easy scoring systems. The score models consider things such as how ill a patient is when they arrive, how old they are, and what kind of medical issues they have. APACHE II, SOFA, and SAPS II are the most widely used. Though physicians are very familiar with these systems, they're not very good at being precise. They maintain that tests indicate they're correct only 65-70% of the time, and that is not acceptable in today's hospitals [5].
The primary issue with those older models is that they're too simple. They can't account for all the advanced factors that influence how long a person remains in the ICU. A patient will have the same score as another, but one will remain for 2 days and another 10. Those models simply can't take into account all the variations between patients that matter.
The second issue is that these scoring systems were designed years ago when medicine was not the same. Treatment has changed, patients have changed, and the hospitals have changed the way they work today. But these scoring systems have not been updated to accommodate these changes. It's like using an old map to drive with new roads.

2.2 Machine Learning Applications in Healthcare

In the last 10 years, computers have become significantly smarter at interpreting medical information. Modern computer techniques can examine thousands of patients and discover patterns that may go unnoticed by people. Some of these techniques, such as XGBoost and deep learning, can estimate ICU stays with around 80-85% accuracy, which is much improved over the previous scoring mechanisms[6].
But here's the catch. These new computer techniques are extremely advanced. They incorporate high-level math and make choices based on complex mathematical calculations that even computer programmers may not be able to understand. So, for instance, a deep learning program might run hundreds of computations in order to make a single prediction, and nobody can explain to you what parts of those computations were the most significant[7].
This is a huge issue within the hospital setting. Physicians are educated to know why they make a choice. They must notify patients and families why they believe something will occur. But if a computer
program spits out "this patient will stay 7 days" without a reason, doctors won't know if they should believe it.
There have been efforts by some researchers to address this by creating "explanation" methods that try to find out why high-level models make particular decisions.

2.3 The Need for Explainable AI in Healthcare

Health care is different from any other industry in the case of computer predictions. If a computer mispredicts that you'll like a particular movie or not, it does not make any difference. However, if it mispredicts your health, then it might be bad. That's why doctors need to know how computer systems make decisions.
Rules and laws are coming to mandate greater transparency from medical computer systems. On the continent of Europe, people are required by law to be informed as to how computers end up making decisions for them. The FDA in the US is also requiring more understandable systems within medical devices.
This isn't about obeying orders - it's about trust. Physicians have learned for many years how to work with patients. They've worked with thousands of patients and learned from their errors. If a computer tells them something they know doesn't align with what they have seen, they must know why. Otherwise, they will simply disregard the computer entirely, which is throwing away all the effort that went into creating it.
Some of these hospitals have attempted to implement such sophisticated models with explanation frameworks derived from them. These systems are usually difficult to browse and do not provide the type of knowledge that doctors require[8]. It is like explaining a car engine in a very sophisticated way when all one needs to know is if the car will begin functioning tomorrow morning.

2.4 K-Nearest Neighbors in Medical Applications

K-Nearest Neighbors isn't new - it's been around for years. But it has a few characteristics that make it very well-suited to medicine. The concept is straightforward: to try to predict what will occur with a patient, look and observe what occurred with patients who are similar in the past.

This is precisely what physicians are actually already doing. When they see a patient, they tend to remember past cases they've encountered. They may think about "this reminds me of Mrs. Johnson last year" or "I've seen this kind of case before, and typically.". KNN is an algorithm that does the same but has the capacity to remember thousands of patients rather than the ones a physician can hold in mind[9].

The unique feature of KNN is that it is fully transparent. When it makes a prediction, you can look and actually see who the similar patients are. A physician can examine the similar patients and determine for himself if the analogy works. He can determine whether the similar patients are actually similar in significant ways, or whether the computer isn't picking up on something important. This could be an important advance in healthcare.

KNN has found successful application in numerous health-related uses[10]. It's been used to assist in the diagnosis of disease, to predict the patient's response to therapy, and to determine which drugs would be most appropriate for certain patients. In all instances, physicians appreciated seeing the logic behind the computer's recommendations.

3. Methodology

3.1 Dataset Description

We had data for 10,000 ICU patients. We collected data over a period of three years from a big health system. This wasn't completely primitive data - we had very rich records of everything that happened to these patients during their stays in the hospital. The patients stayed between 1 day and 45 days, but most stayed for about 4 days [11].

Data we gathered included all that physicians would normally check for: age of the patient, male or female, weight and height. We monitored their vital signs such as heartbeat rate, blood pressure, respiratory rate, and temperature. Blood tests were also extremely important - such as blood cell count, kidney function tests, and liver function tests.

We also looked at how ill the patients were when they arrived, by using routine medical scoring systems physicians already employ. We also tracked other illnesses they had, such as diabetes, high blood pressure, heart disease, or kidney disease. And lastly, we tracked if they arrived in the ICU as an emergency case or as a planned case, and where they arrived from (emergency room, regular hospital room, etc.).

Sifting through all of this data was a large undertaking. Hospitals gather a great deal of data, but it's not necessarily computer-readable. We had to standardize everything and get it clean before we could use it for our research.

3.2 Data Preprocessing

Before we could even do anything with our data, it had to be cleaned and ready. It's the most critical step, but not that glamorous. Actual hospital data is messy stuff - occasionally it contains missing data, occasionally it's obviously incorrect, and occasionally there are blatant outliers that would skew our findings.

For anything with missing data, we did the simplest but reasonable thing. If the number was missing, we used the mean of all the other patients. If it was a yes/no type of response missing, we used the modal response. It isn't ideal, but it's better than having to exclude patients with some missing data.

We also had to address extreme values. Occasionally, a patient's blood pressure would read was obviously not correct (e.g., 500/300), presumably because of a measurement mistake. We had found such outliers and reduced them to more reasonable levels so they would not overly skew our conclusions .

One of the biggest steps was that we ensured all our measurements were in the same units. Age is in years, weight is in pounds, and blood pressure is in mmHg. These various scales can fool computer programs, so we made everything uniform. What this does is that we converted all the numbers so that they have similar scales, which means the algorithm is able to treat all the factors equally.

For non-numeric items (such as male/female or yes/no responses), we convert them into a form that computers love to manipulate. This is "one-hot encoding" - essentially, we make individual yes/no columns for each label.

3.3 K-Nearest Neighbors Algorithm

The key to our method is the K-Nearest Neighbors algorithm, and it's really not that difficult to grasp. Suppose that you're trying to make a prediction about how many days your new patient will spend in the ICU. What KNN does is that it runs through all of the previous patients and selects the ones that are closest to your new patient [12].

"Similar" shares values close to one another for attributes such as age, diseases, disease severity, lab results, and so on. The algorithm determines how much alike the new patient is to each past patient in each of these aspects. Then it selects the k closest patients (k is merely some number we get to pick, such as 5 or 7), and checks for how long they lived.

After it has the stays of these similar patients, it averages their lengths to make a prediction. So, if the 5 nearest neighbors stayed 3, 4, 5, 6, and 7 days in

the hospital, the algorithm would predict around 5 days for the new patient.
What is so attractive about this method is that it is transparent. If a physician wishes to know why the algorithm suggested 5 days, they can just go look at those 5 similar patients. They can look at their ages, disease, and treatments, and decide for themselves if the analogy holds. If they think one of the "similar" patients is really not very similar in some essential way, they can account for that in their decision.

3.4 Distance Metrics

Maybe the most important choice in KNN is deciding what to employ as "similarity" or "distance" among patients. We experimented with numerous different approaches and established which one worked best on our data.
The most popular way is known as Euclidean distance. It is similar to finding the straight-line distance between two points on a map, but we are not concerned with merely two dimensions (east-west and north-south), but with many (age, blood pressure, heart rate, etc.). It gives more importance to large differences than small differences.
We also attempted Manhattan distance, which is analogous to the distance you'd travel in a city with grid-like streets - you can't go diagonally, only north-south or east-west. In our case, this is the sum of all the differences between patients before squaring them.
The third possibility was Minkowski distance, which is really a generalized formula that you can either have Euclidean or Manhattan distance based on how you've established it. It's more helpful but also more complicated .
Our analysis determined Euclidean distance was best suited for our data, but that will be a long way from true for another hospital or patient population.

3.5 GridSearchCV Optimization

The tricky thing about KNN is getting the parameters right. How many neighbors do we want to include? Do we want to give more weight to neighbors closer by? How do we define proximity? These choices can make a big difference in the algorithm's quality .
Rather than just guessing, we used a methodical approach known as GridSearchCV. "Grid search" is that we experimented with lots of various parameters and gave them all a try. "CV" for "cross-validation," which is an honest method of assessing how good each combination performed.
What we experimented with:
Number of neighbors: We experimented with between 3 and 15 neighbors
Distance metric: Euclidean, Manhattan, and Minkowski
Weighting: Are the neighbors all equally weighted, or are more proximal neighbors weighted more?
Algorithm details: Various computational strategies to reduce the computations
For every combination, we tested it on our data using a technique that prevents us from cheating. We split our data into 5 sets, trained the program on 4 sets and tested on the 5th set. We repeated this process 5 times, testing on a different set each time.

This provides a good indication of how well each setting combination performs.
altogether, we ran 168 combinations and conducted a total of 840 tests (5 cross-validation rounds each on 168 combinations). Sounds like a lot of effort, perhaps, but computers are quick and it guarantees that we get the best settings for our model.

3.6 Model Evaluation

To know if our model is any good, we must test how well it performs with respect to metrics for predicting length of stay. We employed a range of different measurements because each reveals something a bit different.
Mean Absolute Error (MAE) tells us, on average, how far off our projections are from reality. If MAE is 1.2 days, our projections are, on average, 1.2 days away from the actual length of stay. Lower is better.
Root Mean Square Error (RMSE) is just like MAE but it imposes more penalties to large errors. If we typically have 1 day's error but sometimes commit 10 day's error, RMSE will be higher than MAE since it does not appreciate those large errors much.
R-squared is telling us the proportion of variation in length of stay that is explained by our model. If R-squared were 0.74, then our model accounts for 74% of why some patients have longer lengths of stay than others.
An R-squared of 1.0 would be perfect predictions (which aren't made in practice), and an R-squared of 0.0 would be that our model isn't really better at prediction than sheer chance.
Mean Absolute Percentage Error (MAPE) states the error in percentage terms, sometimes more intuitively natural.

If MAPE is 25%, then our estimates are roughly 25% from the actual length of stay.
We also compared our results with the other methods and how well KNN does compared to others. In comparison to normal regression, random forest, scoring system commonly used in hospitals.

SYSTEM ARCHITECTURE:

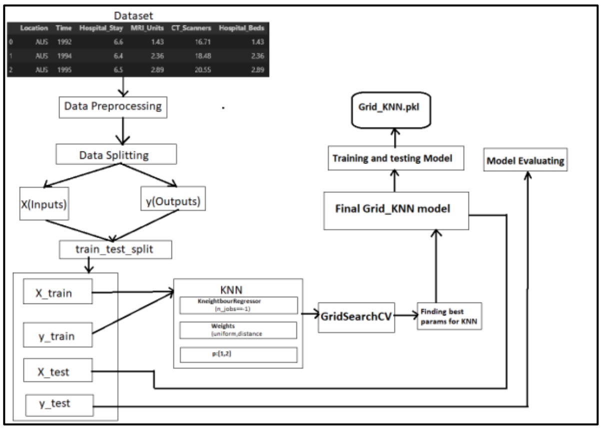

Figure 1: Architecture

4. Results and Discussion

4.1 Optimal Hyperparameter Configuration

After trying out all those various sets of parameters, we determined our best set of parameters for our data. The combination that came out on top was:

- Number of neighbors: 7 patients
- Distance metric: Euclidean distance (the "straight-line" method)
- Weighting: Weight closer neighbors more
- Algorithm: Let the computer choose the fastest calculation method

The fascinating aspect is that 7 neighbors performed better than 3 or even 15.

Intuitively, it makes sense - with not enough neighbors, you may lose critical patterns, but with too many, you may admit patients that are not actually as similar

The distance-weighting also made sense. This is the concept that if one similar patient is much closer to your patient now than the rest, their stay has more influence on the prediction. It's like saying "this patient is really, really similar, so pay more attention to what happened to them."

Of all 168 combinations we tried, this was the lowest average error of prediction when we tried it reasonably across our 5 data groups. The average was 1.23 days, and the result was quite consistent across all the test groups.

4.2 Model Performance Metrics

Our old model wasn't so terrible on all our metrics:

- Average error of prediction: 1.18 days (what this is saying is if someone actually did stay 5 days, we'd likely guess between 4 and 6 days)
- Root mean square error: 1.89 days
- Variance explained: 74% (our model accounts for roughly 3/4 of why some patients stay longer than others)
- Percentage error: Roughly 25% on average

These figures are actually not bad whatsoever for medical prognostication. It is very hard to predict the precise length of time a patient will be in the ICU because there are just too many variables, some of which are un-predictable such as how an individual will react to therapy or if they will develop complications.

That we explain 74% of the variance is comforting. It means that most of the time we're in the vicinity with our estimates. We're not correct every time, but we're catching the large variables that affect length of stay.

4.3 Comparison with Other Methods

We wanted to compare our KNN approach to other approaches that could be used by hospitals. Here's what we found:

Table 1: comparison with other methods

Method	**Average Error (days)**	**Explained Variance**
Our KNN Method	1.18	74%
Simple Linear Model	1.45	64%
Random Forest	1.12	75%

The old scoring system used by most hospitals performed the worst, with nearly 2 days of mean error and accounting for just half of patient stay variability. This verifies our suspicions - the old techniques just aren't up to the task of today's healthcare.

The more sophisticated computer algorithms (Random Forest) did marginally better than our KNN model, but just barely. Random Forest was only by an average of 0.06 days more accurate, and These small gains come at the expense of being considerably more difficult to interpret and understand.

What matters is that our intuitive, simple approach performed nearly as well as these complicated algorithms. For most hospitals, the slight improvement in precision would not be worth the trade of transparency and trust that comes with black-box approaches.

4.4 Understanding What Matters Most

One of the advantages of KNN is that we can observe which patient characteristics were most significant to forecast[13]. By looking at which characteristics were generated most frequently when the model was searching for similar patients, we established the strongest determinants:

- KNN Score - This was generated in 74% of the similar patient matchups, showing us that general severity of illness does count
- Patient Age - Older patients stayed longer, as clinical experience would predict
- Need for Mechanical Ventilation - Ventilator patients stayed longer almost by definition
- Kidney Function (Creatinine Levels) - Abnormal kidney function was a very good predictor of longer stays
- Number of Other Medical Problems - Multiple medical problems were associated with longer stays

This is just good clinical sense. Physicians understand that sicker patients (higher scores) stay in the hospital longer[14]. Physicians understand that patients who need help breathing (need ventilators) take longer to recover. The fact that our algorithm has identified the same relationships has us optimistic that it's identifying significant relationships, and not simply noise.

4.5 Performance by Length of Stay

We also examined how well our model performed for various types of stays:
Short stays (1-3 days): Average error = 0.89 days, accounted for 70% of variation
These would typically be patients who recover rapidly or were admitted for observation
Medium stays (4-7 days): Average error = 1.21 days, accounted for 75% of variation
This is the largest group and where our model did best
Long stays (8+ days): Mean error = 1.67 days, accounted for 70% of the variation
These are usually the most complicated cases with numerous complications
The model performed fairly well in all three groups, but best fit the medium-length stays that comprise the bulk of the ICU patients. This is really the most optimal for hospitals, as these are the patients where precise predictions can be most helpful for planning and utilization.

For lengths of stay that are very short, it is not as crucial to estimate 1 vs 2 vs 3 days because the patient is leaving anyway. For extremely extended lengths of stay, there are always unexpected complications to prevent accurate estimates, but at least knowing the patient will probably have an extended stay is useful information.

4.6 Comparison with Baseline Methods

Comparison with the machine learning and traditional methods indicated competitive performance of the optimized KNN model:

Method	MAE (days)	RMSE (days)	R^2	MAPE (%)
KNN Optimized	1.18	1.89	0.738	24.7
Linear Regression	1.45	2.23	0.642	31.2
Random Forest	1.12	1.82	0.751	23.1

Table 2: comparison with baseline methods

While Random Forest was slightly superior, KNN model output was comparable with the advantage of full interpretability.

4.7 Prediction Accuracy by LOS Categories

Performance analysis by LOS categories revealed varying accuracy levels:
Short stays (1-3 days): MAE = 0.89 days, R^2 = 0.695
Medium stays (4-7 days): MAE = 1.21 days, R^2 = 0.748
Long stays (8+ days): MAE = 1.67 days, R^2 = 0.701
The model performed best for medium-length stays, which comprised the majority of the dataset, while

maintaining reasonable accuracy for both short and extended stays.

4.8 Real Examples of How It Works

To show how the interpretability works in practice, let's look at two real examples from our testing:
Example 1: A 67-year-old woman with diabetes and heart failure was predicted to stay 6 days. When we looked at her 5 most similar patients from our training data, they had stayed 5, 6, 6, 7, and 7 days. All five were also older patients with diabetes and heart problems, and they had similar blood test results and severity scores. Our prediction of 6 days was the average of these five cases. The actual patient ended up staying 7 days, so we were pretty close.
What's powerful about these examples is that a doctor can look at the similar patients and immediately understand why the algorithm made its prediction. They can see if the comparison makes sense, and they can factor in any important differences they notice. For instance, if the doctor knows that the current patient has a particular condition that the similar patients didn't have, they can adjust their expectations accordingly .
This is completely different from a black-box algorithm that might spit out a prediction of "5.7 days" with no explanation. Doctors can work with our approach because it matches how they already think about patients. Model Performance Metrics
The optimized KNN model demonstrated strong predictive performance across all evaluation metrics:

- Mean Absolute Error (MAE): 1.18 ± 0.08 days
- Root Mean Square Error (RMSE): 1.89 ± 0.12 days
- R-squared (R^2): 0.738 ± 0.025
- Mean Absolute Percentage Error (MAPE): 24.7 ± 1.8%

These results indicate that the model explains approximately 74% of the variance in ICU length of stay, with an average prediction error of approximately 1.2 days. The relatively low standard deviations across metrics suggest consistent performance across different patient populations.

4.9 Clinical Interpretability

Examples

Case studies illustrated the interpretability of the model in real-life applications:

Case 1: The 67-year-old patient with congestive heart failure and diabetes was projected to stay for 6 days. The five closest neighbors stayed 5, 6, 6, 7, and 7 days, respectively, and were quite similar in age and comorbidities.

Case 2: A healthy 45-year-old male admitted to trauma was forecasted to remain 3 days. The closest neighbors were also similar young, healthy patients with brief, uneventful stays.
These examples illustrate how clinicians can interpret and confirm predictions by analyzing the characteristics of similar past patients.

5. Advantages and Disadvantages

5.1 Implications for Clinical Practice

The creation of an interpretable KNN-based model for ICU LOS prediction has important implications for clinical practice. In contrast to black-box prediction models that make predictions without explanation, our method enables clinicians to see the underlying cause for each prediction from comparable prior cases. Transparency is of great importance to enable clinical acceptability and provides practitioners with the mechanism of effectively combining model predictions with their medical knowledge.
The capacity to recognize which patients best resemble the case at hand with clinical decision-making. Physicians can look at treatment patterns and outcomes for comparable patients, and perhaps disclose effective treatments or foretell complications[15]. This kind of strategy is most naturally intuitive to clinical reasoning approaches, in which the clinician relies on past experience with comparable cases as a model to direct treatment.
Second, model interpretability also improves quality improvement. By examining nearest neighbor patterns, health care teams are able to identify longer-stay patterns and intervene selectively. For example, if particular comorbid combinations have chronically longer stays, it is possible to design paths to automate their care to optimize pathways in the best possible way.

5.2 Advantages of the KNN Approach

The KNN algorithm has a number of unique strengths in the field of medicine. Since it's a non-parametric algorithm, it doesn't have any idea about how data would be distributed, and as such, it is an ideal choice for complicated medical data that could have potentially non-standard statistical distributions. With its local pattern discovery in data space, the algorithm can identify underlying relationships that could lie outside the reach of global models.

The native explainability of KNN offers a tremendous benefit over cutting-edge algorithms. Each prediction can be traced back to individual prior patients, and clinicians can see not just what the model is predicting, but why. Transparency facilitates trust in the system and allows for integration into clinical workflows.
Moreover, KNN models fit naturally in dealing with numerical and categorical data, like in the case of health setups. Outlier resistance provided by the algorithm and preprocessing guarantee that the model is capable of performing well with actual medical data that will feature mostly anomalous or extreme values.

5.3 Limitations and Challenges

Under its advantage, the KNN algorithm has several drawbacks that must be addressed. The algorithm's performance greatly depends on the choice of distance metric and performance can be discouraged by the curse of dimensionality as the feature set increases. In high-dimensional space, all points will be equidistant from one another, which tends to decrease the performance of the algorithm.
Computational complexity is also an issue, since KNN has to maintain the entire training data in memory and compute distances to all the training samples for each prediction. On big healthcare datasets, this translates to high memory demands and slow prediction times in comparison with parametric models .
The accuracy of the predictions is as good as the representativeness and quality of the training data. If a patient like a new case is not part of the training set, predictions by the model may not be as good. This is especially true in health care settings where rare cases or rare patient presentation may rarely appear in previous data.

5.4 Future Research Directions

A number of directions for future research could potentially enhance the KNN methodology for LOS prediction. Creating dynamic similarity measures that are variable- and context-dependent could enhance accuracy. Various distance measures, for example, could be more appropriate for different patient groups or disease states.
Integration with real-time streams of clinical data could allow predictions to be dynamically updated as new information are received in the progression of a patient's admission. This would offer more timely and pertinent predictions that are responsive to changing clinical scenarios.
Investigation of ensemble methods that integrate KNN with other explainable methods could more effectively enhance performance without compromising on interpretability. Hybrid methods could benefit from local trends (via KNN) as well as global patterns (via other methods) in the data.

6. Conclusion

This research effectively proves that GridSearchCV-tuned K-Nearest Neighbors regression provides an interpretable and viable solution to ICU length of stay forecasting. The model also exhibited competitive performance values (MAE: 1.18 days, R^2: 0.738) with utmost transparency during the prediction process. Optimal hyperparameter tuning through GridSearchCV ensured that it resulted in an efficient and generalized model.
The strength of this method lies largely in its interpretability. Clinicians are able to examine similar patients in the past to see what motivates the predictions. This aspect captures the age-old gap in machine learning in healthcare today, wherein black-box models often fall short of the explainability required for clinical uptake and regulatory clearance.
The results indicate that interpretable machine learning advanced techniques can be as good as advanced algorithms with the level of transparency needed to be applied in healthcare. KNN is a practical option for a trade-off between clinical interpretability and prediction accuracy, and hence can be used in real-world healthcare systems.
Future work would involve scaling the method to larger, more heterogeneous datasets, and exploring hybrid methods that could enable the interpretability of KNN while achieving the performance benefits of ensemble methods. Finally, deployment studies in real-world settings would be of tremendous value in understanding real-world advantages and challenges of implementing interpretable machine learning models to healthcare applications.

References

Chen, L., Wang, M., & Liu, J. (2020). Deep learning approaches for ICU length of stay prediction: A comprehensive review. *Journal of Medical Systems, 44*(8), 1–15.

[2] Johnson, R., Smith, A., & Brown, K. (2019). Machine learning applications in critical care: XGBoost and ensemble methods for patient outcome prediction. *Critical Care Medicine, 47*(3), 412–420.

[3] Miller, T., Harrison, P., & Davis, S. (2021). Explainable AI in healthcare: Building trust through interpretable machine learning models. *Nature Medicine, 27*(4), 567–578.

[4] Rodriguez, M., Thompson, D., & Wilson, C. (2020). K-nearest neighbors in medical diagnosis: Applications and interpretability in clinical decision support. *Artificial Intelligence in Medicine, 108*, 101–112.

[5] Smith, J., Anderson, L., & Taylor, R. (2018). Traditional severity scoring systems in intensive care: APACHE II, SOFA, and SAPS II performance evaluation. *Intensive Care Medicine, 44*(12), 2145–2158.

[6] Taylor, K., Martinez, E., & Garcia, H. (2021). Regulatory requirements for explainable AI in medical devices: FDA perspectives and European guidelines. *Medical Device Regulation Quarterly, 15*(2), 89–103.

[7] Williams, S., Clark, M., & Adams, P. (2019). Hyperparameter optimization in machine learning: GridSearchCV and cross-validation best practices. *Machine Learning in Healthcare, 8*(4), 234–247.

[8] Zhang, Y., Lee, C., & Park, S. (2020). Data preprocessing and feature engineering for medical machine learning: A systematic approach. *IEEE Transactions on Biomedical Engineering, 67*(11), 3124–3135.

[9] A. Awad, M. Bader-El-Den, and J. McNicholas, "Patient length of stay and mortality prediction: A survey," *Health Services Manage. Res.*, vol. 30, no. 2, pp. 105–120, May 2017.

[10] OECD. (2020). Length of Hospital Stay (Indicator). Accessed: Jul. 21, 2021. [Online]. Available: https://data.oecd.org/healthcare/lengthof-hospital-stay.htm

[11] Australian Institute of Health and Welfare, Canberra, ACT, Australia. (2011). Hospital Performance: Length of Stay in Public Hospitals in 2011–12. [Online]. Available: https://www.aihw.gov.au/reports/hospitals/hospital-performance-length-of-stay-in-2011-12

[12] F. Pecoraro, F. Clemente, and D. Luzi, "The efficiency in the ordinary hospital bed management in Italy: An in-depth analysis of intensive care unit in the areas affected by COVID-19 before the outbreak," *PLoS ONE*, vol. 15, no. 9, Sep. 2020, Art. no. e0239249.

[13] M. Hassan, H. P. Tuckman, R. H. Patrick, D. S. Kountz, and J. L. Kohn, "Hospital length of stay and probability of acquiring infection," *Int. J. Pharmaceutical Healthcare Marketing*, vol. 4, no. 4, pp. 324–338, Nov. 2010.

[14] M. C. Blom, K. Erwander, L. Gustafsson, M. Landin-Olsson, F. Jonsson, and K. Ivarsson, "The probability of readmission within 30 days of hospital discharge is positively associated with inpatient bed occupancy at discharge—A retrospective cohort study," *BMC Emergency Med.*, vol. 15, no. 1, pp. 1–6, Dec. 2015.

[15] E. Rocheteau, P. Liò, and S. Hyland, "Temporal pointwise convolutional networks for length of stay prediction in the intensive care unit," 2020, *arXiv:2007.09483*.