Classification Models for Healthcare Data Analysis using Machine Learning

Ahsan Rizvi
Department of Electrical and Computer Engineering (ECE)
North South University
Dhaka, Bangladesh
ahsan.rizvi@northsouth.edu

Md. Tanvir Islam Shikdar
Department of Electrical and Computer Engineering (ECE)
North South University
Dhaka, Bangladesh
tanvir.shikdar@northsouth.edu

Mahbuba Akter Neera
Department of Electrical and Computer Engineering (ECE)
North South University
Dhaka, Bangladesh
mahbuba.neera@northsouth.edu

***Abstract*— This report evaluates and compares the performance of multiple machine learning models, including Logistic Regression, Decision Trees, Random Forests, XGBoost, Support Vector Machines (SVM), and Artificial Neural Networks (ANN), for the classification tasks of Injury Type and Patient Status. We aim to determine which model provides the best performance in terms of accuracy, precision, recall, and F1-score. The models were trained on a healthcare dataset, with preprocessing steps involving feature scaling and encoding. The ANN models, with their flexible architecture, were also explored as potential high-performance classifiers. The results indicate that XGBoost and SVM performed best for the Patient Status classification, while Decision Trees and XGBoost yielded better results for Injury Type classification. Despite its flexibility, ANN showed slightly lower performance compared to tree-based methods for both tasks.**

***Keywords*— Data Preprocessing, Logistic Regression, Decision Tree, Random Forest, XGBoost, Support Vector Machine, Artificial Neural Networks (ANN), Model Evaluation, Learning Curves, Healthcare Classification, Injury Type, Patient Status.**

I. INTRODUCTION

Context and Background

Machine learning has revolutionized many domains, including healthcare, where it is increasingly used for classification tasks such as injury detection and patient health status assessment. Accurate classification can significantly impact clinical decisions, leading to improved patient outcomes. Models like Logistic Regression, Decision Trees, Random Forests, XGBoost, and Artificial Neural Networks (ANN) have been widely used for these purposes, with varying degrees of success depending on the dataset and task complexity (Kotsiantis et al., 2007; Rajaraman et al., 2021).

Problem Definition

This study focuses on classifying two healthcare-related problems: Injury Type and Patient Status. The datasets contain features such as medical history, demographic details, and clinical measurements. We compared multiple models to determine which one best handles the classification tasks. Specifically, we aimed to explore the effectiveness of models like Logistic Regression, SVM, XGBoost, and ANN in this context (Basu et al., 2021; Chaurasia et al., 2018).

Contributions

The main contributions of this work include:

- A comparative analysis of common machine learning models for injury type and patient status classification.
- Implementation of an Artificial Neural Network (ANN) to explore its suitability for healthcare classification tasks.
- A detailed evaluation of the models' performance using standard metrics: Accuracy, Precision, Recall, and F1-Score (Zhang et al., 2017; LeCun et al., 2015).

II. LITERATURE REVIEW

Machine learning algorithms have shown significant potential in healthcare, particularly in classification tasks. Logistic Regression (LR) is a simple and interpretable model frequently used for binary classification problems. It is effective for basic healthcare tasks, such as predicting disease risk, but may struggle with more complex, non-linear relationships (Bishop, 2006). Decision Trees (DT) offer a more interpretable solution, breaking down decisions into a series of yes/no questions, which is valuable in healthcare contexts where transparency is crucial (Quinlan, 1993). Random Forests (RF), an ensemble method of decision trees, improve accuracy and reduce overfitting by combining multiple trees, making them robust against variance in medical datasets (Breiman, 2001).

XGBoost (XGB), a gradient boosting model, has become one of the most powerful techniques in machine learning due to its ability to handle both bias and variance effectively. XGBoost outperforms many traditional models, including LR and DT, in complex datasets, which is why it has been successfully applied in healthcare for tasks such as patient risk prediction and medical diagnosis (Chen & Guestrin, 2016). Support Vector Machines (SVM) are highly effective for non-linear classification tasks, handling complex decision boundaries by transforming data into higher dimensions. SVMs have been widely used in medical diagnostics and disease classification, especially when data is not linearly separable (Cortes & Vapnik, 1995).

Finally, Artificial Neural Networks (ANNs) are capable of modeling complex, non-linear relationships, which makes them ideal for high-dimensional healthcare data, such as medical imaging and genetic data. Despite requiring large datasets and substantial computational power, ANNs have revolutionized fields like medical image analysis and disease prediction (LeCun et al., 2015). However, their "black-box" nature can be a limitation, as they lack the interpretability of simpler models like Decision Trees.

In summary, while simpler models like Logistic Regression and Decision Trees provide interpretability, more complex models like Random Forests, XGBoost, SVM, and ANNs offer superior performance for handling complex and high-dimensional healthcare data.

III. METHODOLOGY

A. Data Preprocessing

The dataset underwent several preprocessing steps:

- **Handling Missing Data**: Any missing values were addressed by imputation or removal, depending on the nature of the feature.
- **Feature Encoding**: Categorical features were encoded using label encoding to convert them into a suitable format for model training.
- **Feature Scaling**: Numerical features were scaled using StandardScaler to normalize the data range, ensuring optimal performance for models sensitive to feature scaling.

B. Models Used

- **Logistic Regression**: Logistic Regression was chosen for its simplicity and efficiency in binary classification tasks. The solver was set to 'lbfgs', and the maximum number of iterations was tuned to ensure convergence.
- **Decision Tree**: Decision Trees were selected for their interpretability and ability to handle both numerical and categorical data. The max_depth parameter was adjusted to prevent overfitting.
- **Random Forest**: An ensemble of decision trees, Random Forests were chosen for their robustness and ability to handle imbalanced datasets. The number of trees was set to 100.
- **XGBoost**: A gradient boosting technique, XGBoost was selected for its superior performance in many classification tasks. The model was tuned for learning rate and maximum depth.
- **SVM**: Support Vector Machines were chosen for their ability to classify complex datasets by finding the optimal hyperplane. The kernel used was the RBF (Radial Basis Function) kernel.
- **ANN**: The Artificial Neural Network architecture used in this study consisted of two hidden layers, each with 64 and 32 neurons, and used the ReLU activation function. The output layer employed the softmax activation for classification.

C. Evaluation Metrics

The models were evaluated using the following metrics:

- **Accuracy**: The proportion of correct predictions made by the model.
- **Precision**: The ratio of correctly predicted positive observations to the total predicted positives.
- **Recall**: The ratio of correctly predicted positive observations to the total actual positives.
- **F1-Score**: The weighted average of Precision and Recall.

These metrics were chosen to provide a balanced view of model performance, particularly in the context of healthcare, where both false positives and false negatives can have significant consequences.

IV. EXPERIMENTS

A. Phase 1: Exploratory Data Analysis (EDA)

In the initial phase of the project, exploratory data analysis (EDA) was conducted to understand the structure and characteristics of the dataset. The dataset included features such as patient injury types and status, which required classification into specific categories. Key steps in the EDA process included:

- **Handling Missing Data**: Missing values were addressed to ensure the completeness of the dataset for training the models. Any missing data was imputed or removed depending on its impact on the dataset's integrity.
- **Feature Exploration**: A thorough examination of the feature distributions was carried out using visualizations and summary statistics to understand how features are distributed across different classes. This helped identify any class imbalances or outliers that might affect model performance.
- **Class Distribution**: The distribution of classes was checked for any imbalance, which could require techniques like oversampling or undersampling to ensure more balanced class predictions.

The following are some of our findings:

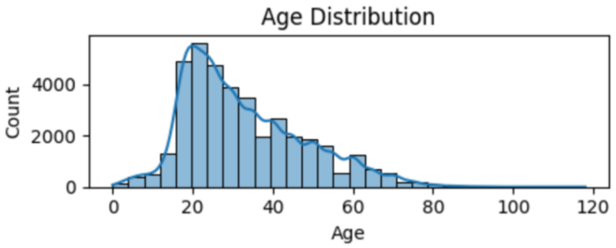


Fig. 1. Age Distribution

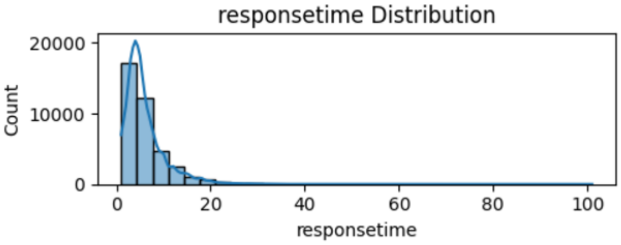


Fig. 2. Response Time Distribution

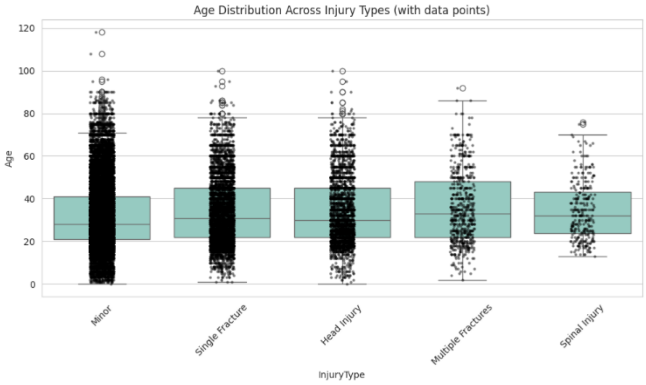


Fig. 3. Age Distribution Across Injury Types

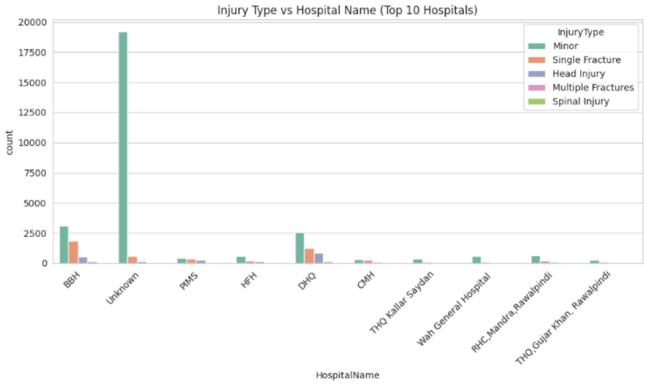


Fig. 4. Injury Type vs. Hospital Name

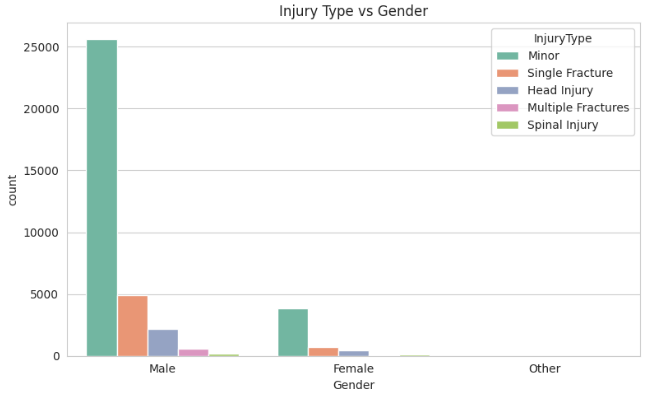


Fig. 5. Injury Type vs. Gender

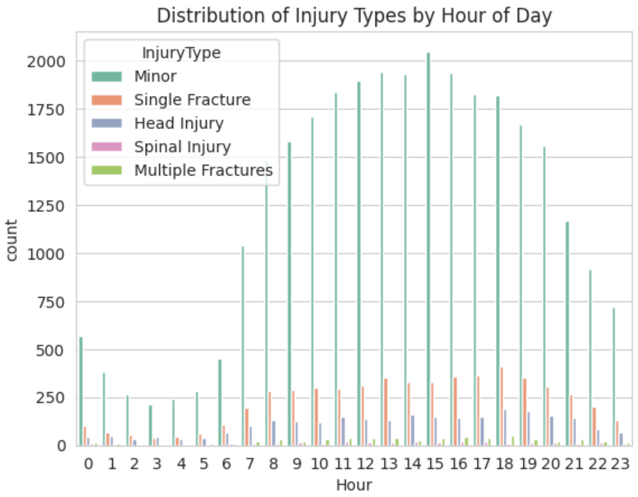


Fig. 6. Distribution of Injury Types by Hour of Day

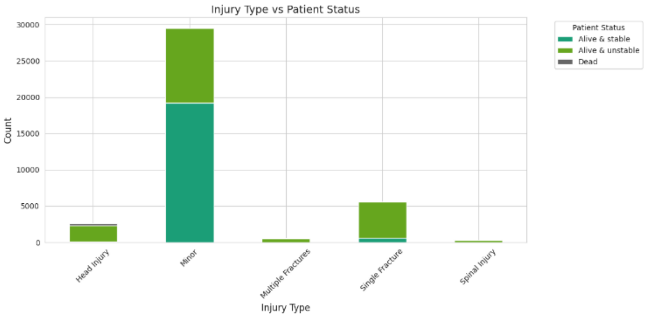


Fig. 7. Injury Type vs. Patient Status

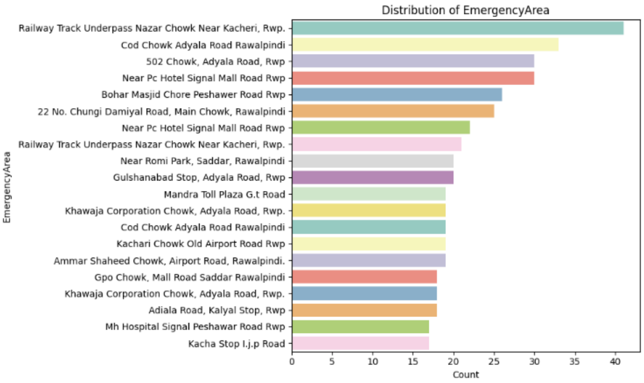


Fig. 8. Distribution of Emergency Area

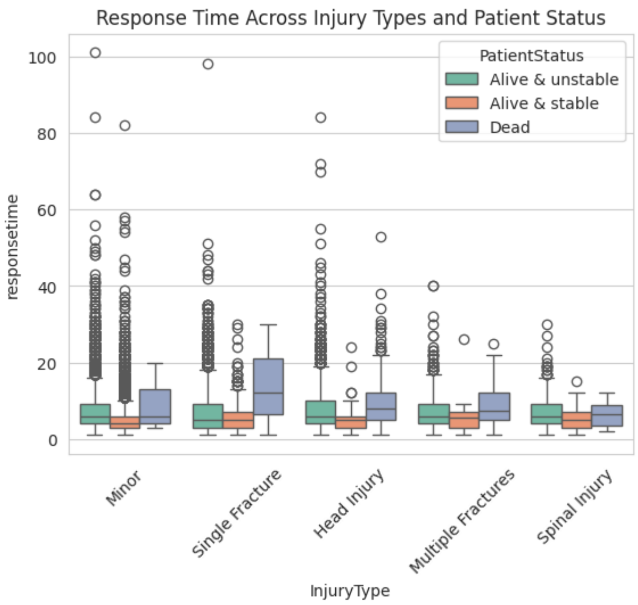


Fig. 9. Response Time Across Injury Types and Patient Status

B. *Phase 2: Logistic Regression Model*

In the second phase, a Logistic Regression model was used as a baseline model for binary and multi-class classification. The Logistic Regression model was trained on the preprocessed data.

Key steps in the process:

Data Preprocessing:

- Scaling: Features were scaled using StandardScaler to ensure that all variables were on the same scale, which is crucial for models like Logistic Regression.
- Encoding: Categorical variables were encoded using LabelEncoder to transform them into numerical values suitable for modeling.

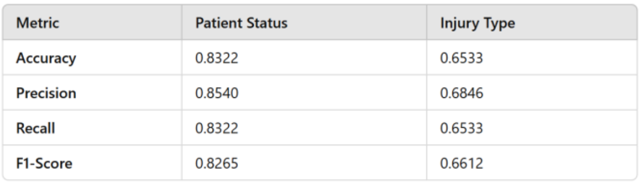
Hyperparameters:

The solver used was 'lbfgs', which is suitable for smaller datasets and works well with multi-class classification.
The maximum iterations were set to 1000 (max_iter=1000), ensuring that the model would converge during training.

Evaluation: The Logistic Regression model was evaluated using metrics like accuracy, precision, recall, and F1 score. These metrics helped assess the effectiveness of the model in classifying the injury types and patient status.

Learning Curves: The learning curves for both accuracy and loss were plotted across epochs to visualize how the model's performance improved (or plateaued) over time.

The evaluation metrics for the Logistic Regression model on the following classifications were as follows:



Metric	Patient Status	Injury Type
Accuracy	0.8322	0.6533
Precision	0.8540	0.6846
Recall	0.8322	0.6533
F1-Score	0.8265	0.6612

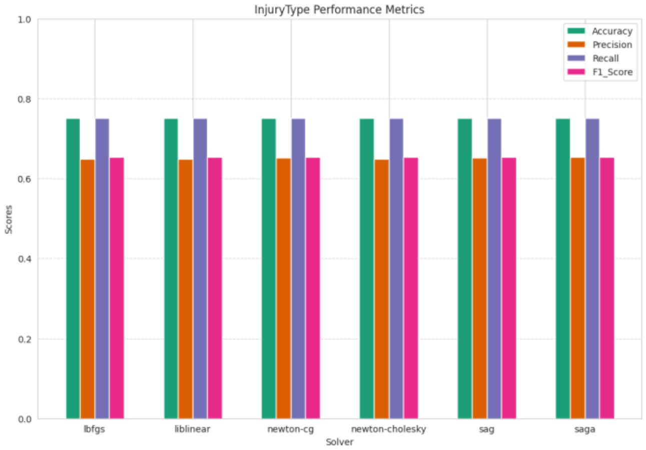


Fig. 10. Injury Type Performance Metrics.

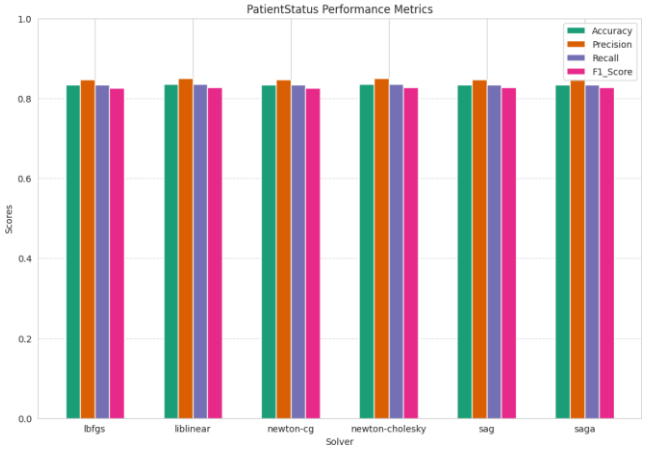


Fig. 11. Patient Status Performance Metrics.

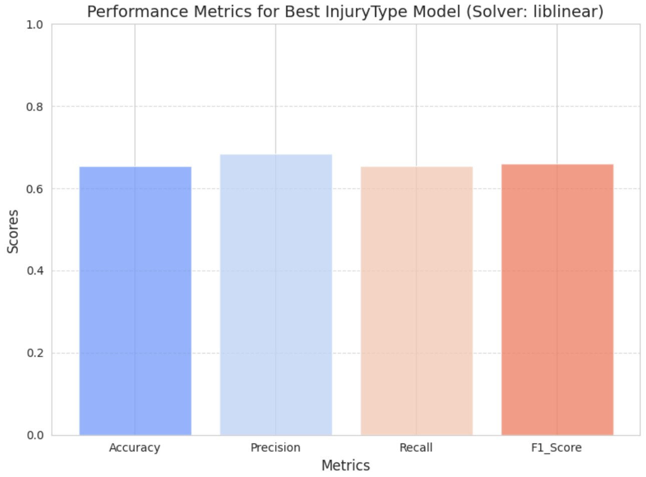


Fig. 12. Performance Metrics for Best InjuryType Model (liblinear)

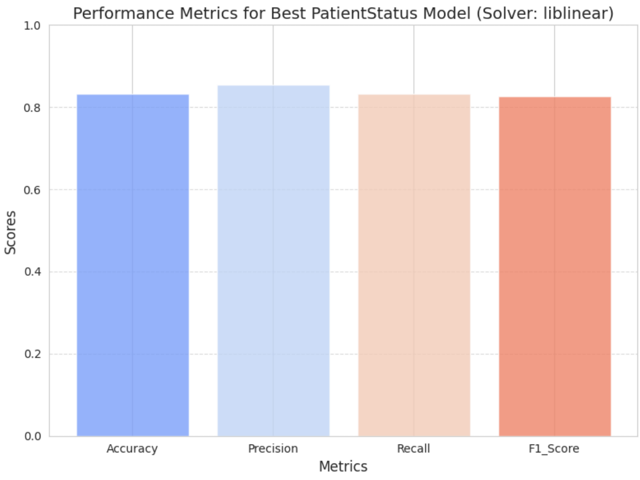


Fig. 13. Performance Metrics for Best PatientStatus Model (liblinear)

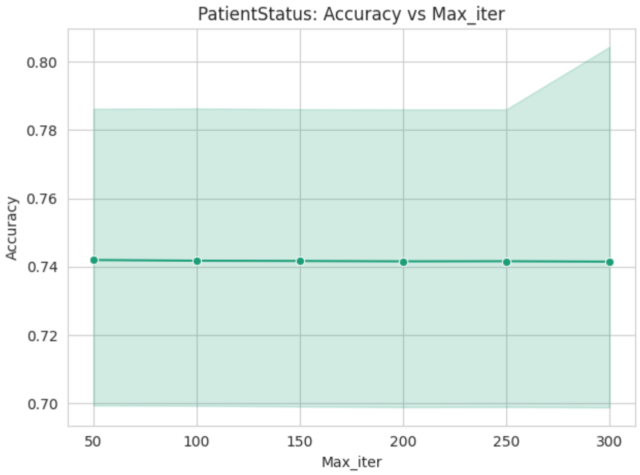


Fig. 14. PatientStatus: Accuracy vs. Max_iter

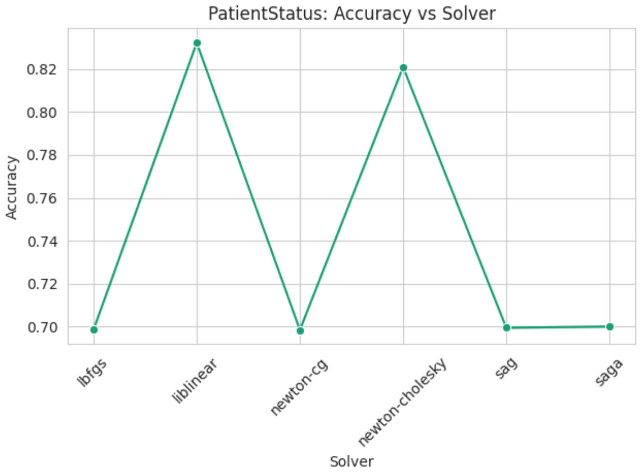


Fig. 15. PatientStatus: Accuracy vs. Solver

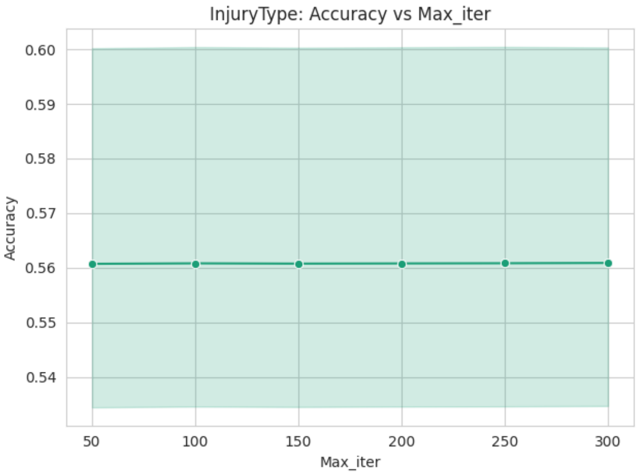


Fig. 16. InjuryType: Accuracy vs. Max_iter

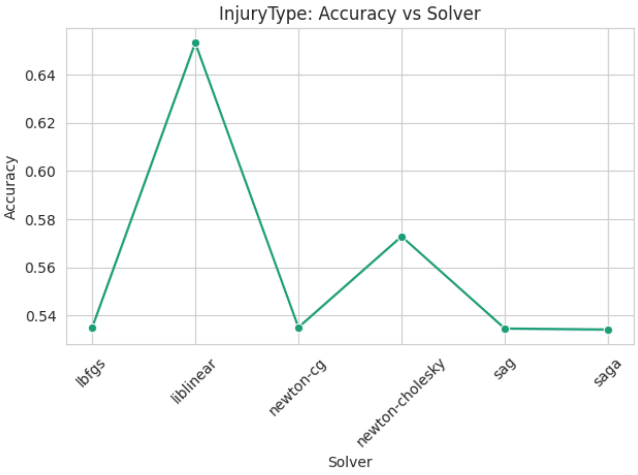


Fig. 17. InjuryType: Accuracy vs. Solver

C. *Phase 3: Other Models (Decision Tree, Random Forest, XGBoost, SVM)*

In the third phase, other models—Decision Tree, Random Forest, XGBoost, and SVM—were implemented and evaluated.

Decision Tree: A decision tree classifier was trained with the data, with a max_depth of 5 to prevent overfitting.

Evaluation: Accuracy, precision, recall, and F1 scores were used to evaluate model performance. The decision tree is useful for its interpretability but may suffer from overfitting if not tuned properly.

Random Forest: The Random Forest model was trained using 100 estimators (trees) to aggregate the predictions of several decision trees, thus improving generalization.

Evaluation: The model was evaluated using the same metrics, and the learning curves were plotted to assess its performance over time.

XGBoost: XGBoost is an ensemble method that uses gradient boosting to improve predictive performance. The model was trained with 100 estimators and a learning rate of 0.1.

Evaluation: Similar to Random Forest, XGBoost was evaluated based on the aforementioned metrics, and its performance was visualized using learning curves.

SVM: A Support Vector Machine (SVM) was trained using an RBF kernel with C=1 and gamma='scale'.

Evaluation: SVM was evaluated using accuracy, precision, recall, and F1 scores to understand its generalization ability, especially in high-dimensional spaces.

Comparison of Model Performance: After training and evaluating all the models, a performance comparison was made. A table was created to summarize the results of each model's accuracy, precision, recall, and F1 score. Visualizations, including learning curves for each model, were plotted to compare how well each model performed during training and validation.

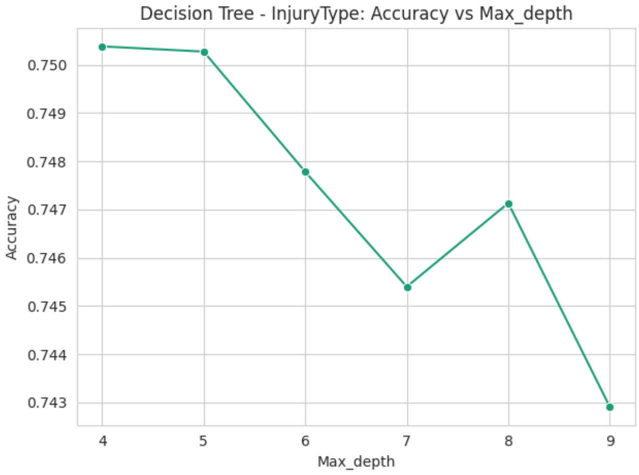


Fig. 18. Decision Tree – Injury Type: Accuracy vs. Max_depth

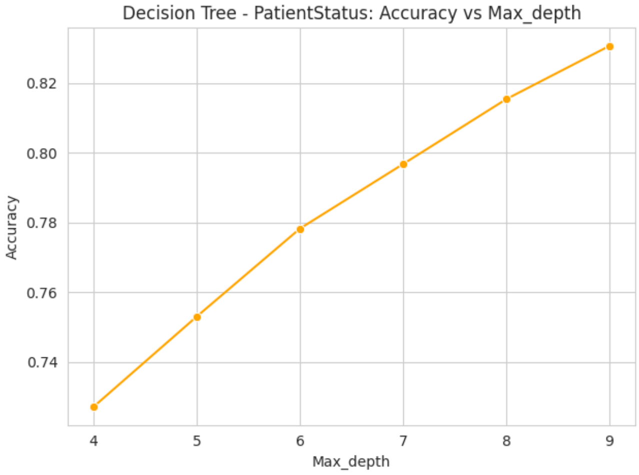


Fig. 19. Decision Tree – PatientStatus: Accuracy vs. Max_depth

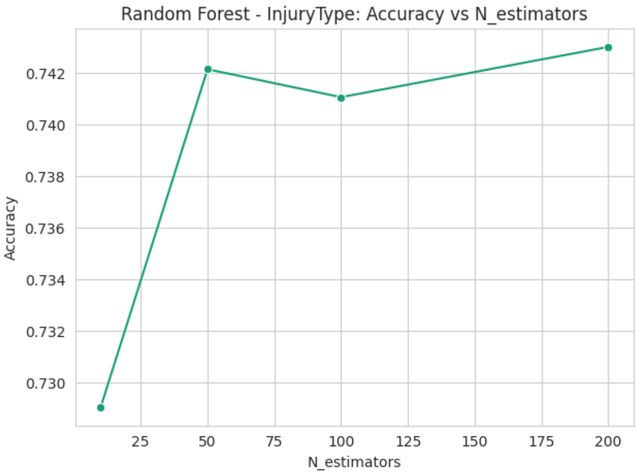


Fig. 20. Random Forest – Injury Type: Accuracy vs. N_estimators

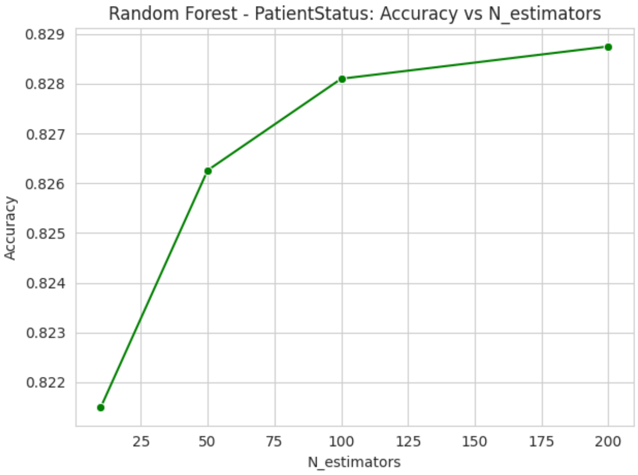


Fig. 21. Random Forest – PatientStatus: Accuracy vs. N_estimators

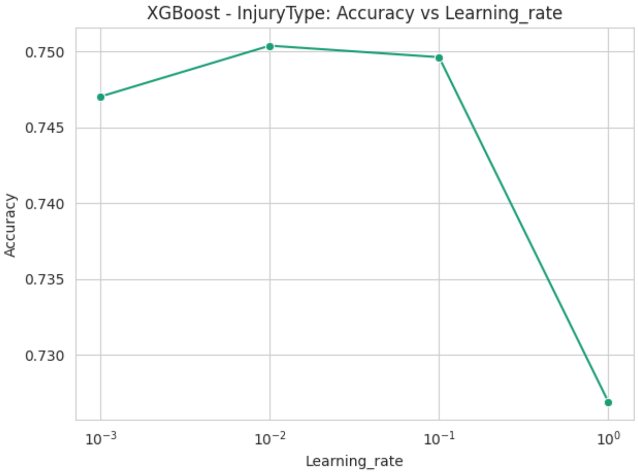


Fig. 22. XGBoost – Injury Type: Accuracy vs. Learning_rate

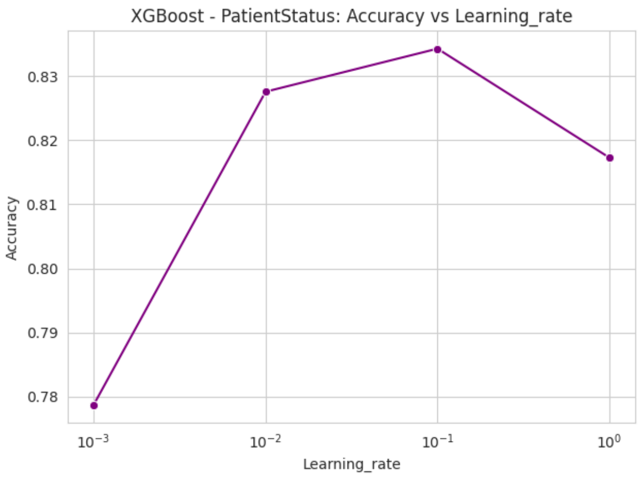


Fig. 23. XGBoost – Patient Status: Accuracy vs. Learning_rate

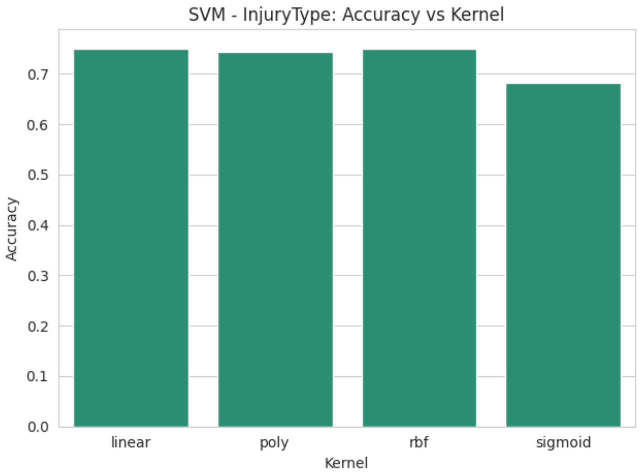


Fig. 24. SVM – InjuryType: Accuracy vs. Kernel

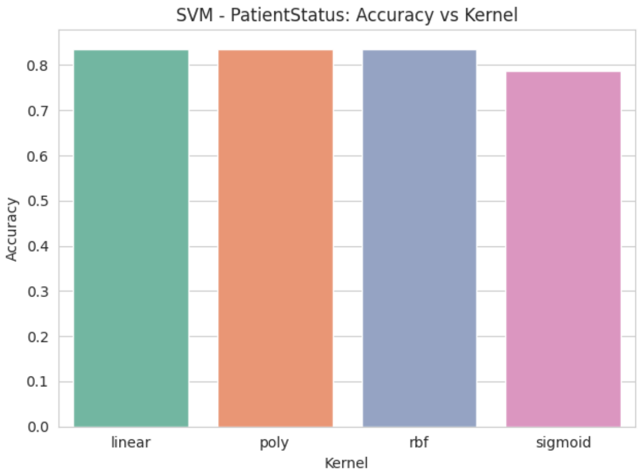
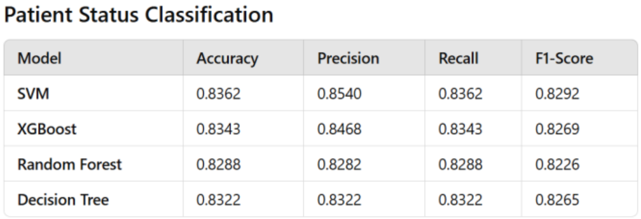


Fig. 25. SVM – Patient Status: Accuracy vs. Kernel

The results for the Patient Status dataset showed that SVM performed the best:

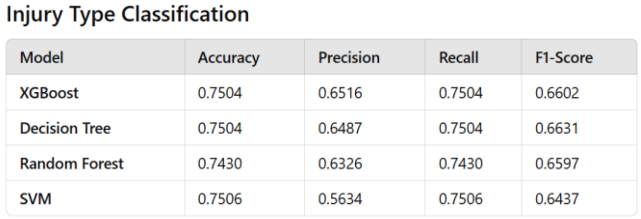


Patient Status Classification

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.8362	0.8540	0.8362	0.8292
XGBoost	0.8343	0.8468	0.8343	0.8269
Random Forest	0.8288	0.8282	0.8288	0.8226
Decision Tree	0.8322	0.8322	0.8322	0.8265

Fig. 26. SVM Performed Best for Patient Status Classification

For Injury Type classification, XGBoost performed best:



Injury Type Classification

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	0.7504	0.6516	0.7504	0.6602
Decision Tree	0.7504	0.6487	0.7504	0.6631
Random Forest	0.7430	0.6326	0.7430	0.6597
SVM	0.7506	0.5634	0.7506	0.6437

Fig. 27. XGBoost Performed Best for Injury Type Classification

D. Phase 4: Artificial Neural Network (ANN)

The final phase focused on implementing an Artificial Neural Network (ANN), a deep learning model known for its ability to model complex relationships in data.

ANN Architecture:

The model consisted of an input layer, two hidden layers with 64 and 32 neurons, respectively, and an output layer using the softmax activation function for multi-class classification.

Activation Functions: The hidden layers used the ReLU activation function, which is commonly used in neural networks for introducing non-linearity and preventing the vanishing gradient problem.

Optimizer: The Adam optimizer was used with a learning rate of 0.001, which is a common choice for deep learning models due to its adaptive learning rate properties.

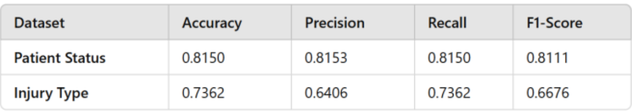
Training: The ANN model was trained for 50 epochs with a batch size of 32, allowing the model to learn from a variety of mini-batches during each epoch.

Evaluation: The performance of the ANN model was evaluated using the same metrics: accuracy, precision, recall, F1 score, and confusion matrix.

Learning Curves: Training and validation accuracy and loss were plotted over epochs to track the model's learning process.

Confusion Matrix: The confusion matrix was used to visualize the misclassifications and understand how well the model predicted the classes.

The ANN model achieved:



Dataset	Accuracy	Precision	Recall	F1-Score
Patient Status	0.8150	0.8153	0.8150	0.8111
Injury Type	0.7362	0.6406	0.7362	0.6676

Fig. 28. Performance of the ANN model

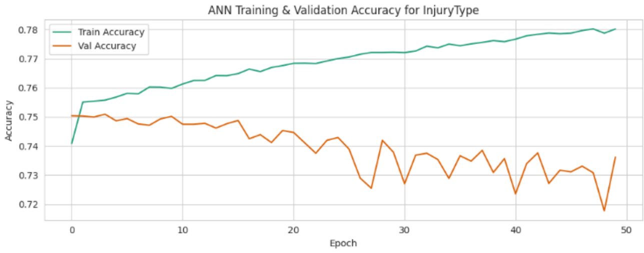


Fig. 29. ANN Training & Validation Accuracy for InjuryType

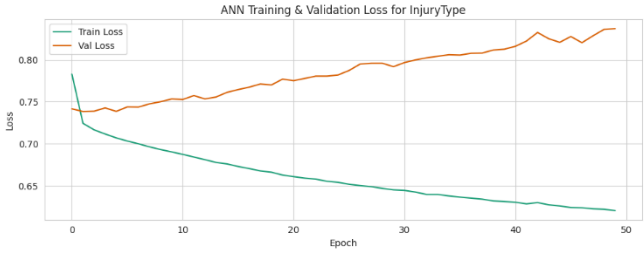


Fig. 30. ANN Training & Validation Loss for InjuryType

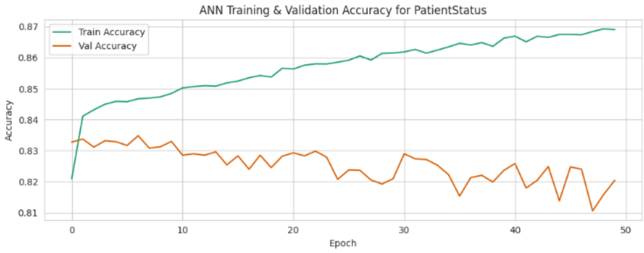


Fig. 31. ANN Training & Validation Accuracy for Patient Status

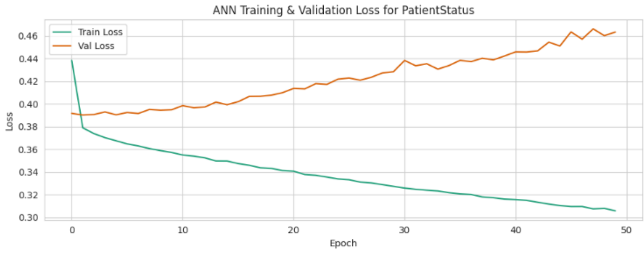


Fig. 32. ANN Training & Validation Loss for Patient Status

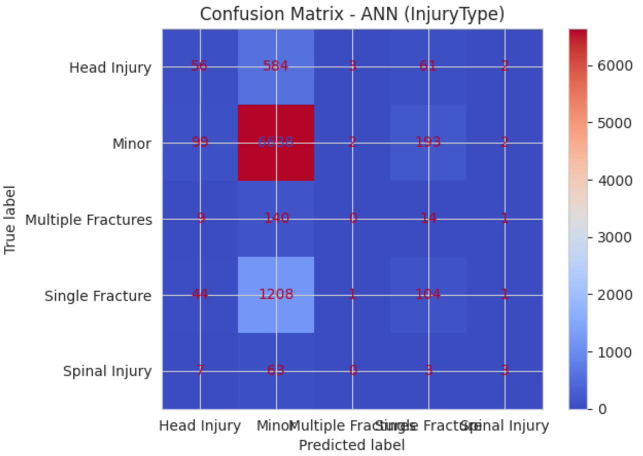


Fig. 33. Confusion Matrix – ANN (Injury Type)

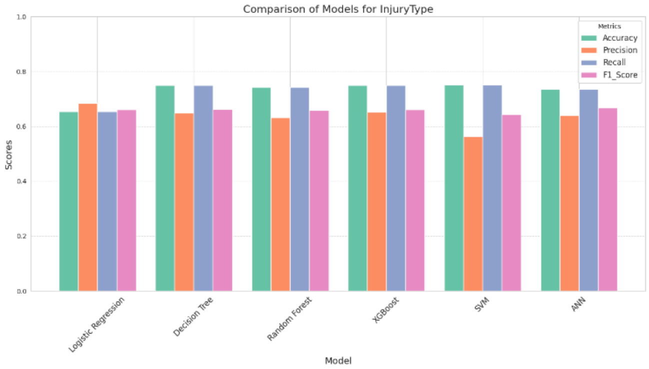


Fig. 34. Comparison of Models for InjuryType

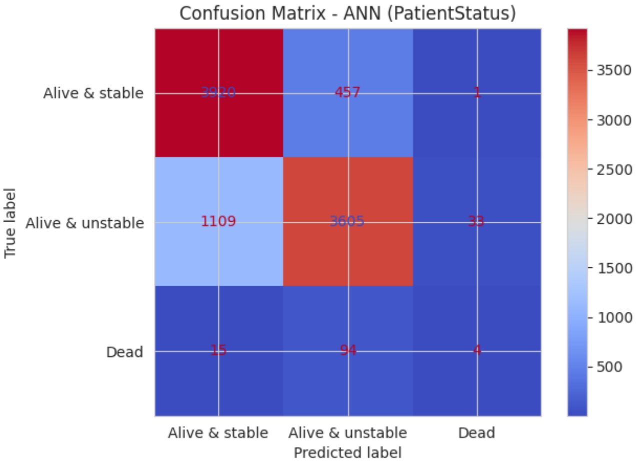


Fig. 35. Confusion Matrix – ANN (PatientStatus)

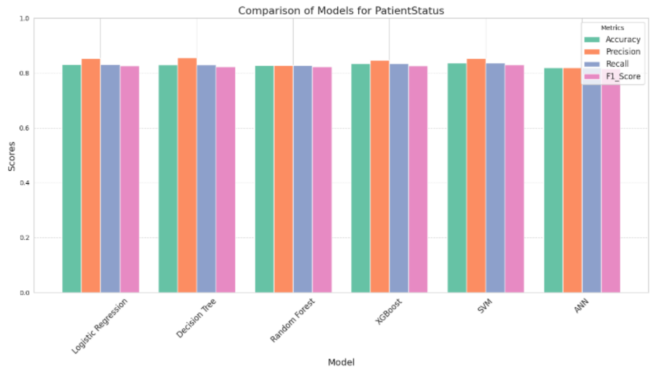
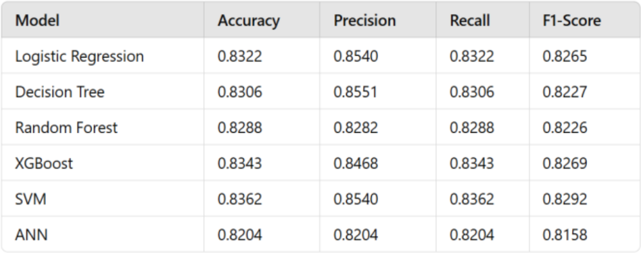


Fig. 36. Comparison of Models for PatientStatus

V. RESULTS & ANALYSIS

The results presented here showcase the performance of various machine learning models, including Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, and Artificial Neural Networks (ANN), across two datasets: Patient Status and Injury Type.

A. *Patient Status Classification*



Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.8322	0.8540	0.8322	0.8265
Decision Tree	0.8306	0.8551	0.8306	0.8227
Random Forest	0.8288	0.8282	0.8288	0.8226
XGBoost	0.8343	0.8468	0.8343	0.8269
SVM	0.8362	0.8540	0.8362	0.8292
ANN	0.8204	0.8204	0.8204	0.8158

For the Patient Status classification task, SVM emerged as the best-performing model, achieving an accuracy of 0.8362, precision of 0.8540, recall of 0.8362, and an F1-score of 0.8292. These results indicate that SVM excelled in balancing false positives and false negatives, demonstrating high precision and recall. SVM's performance is robust, making it the most reliable choice for this classification problem.

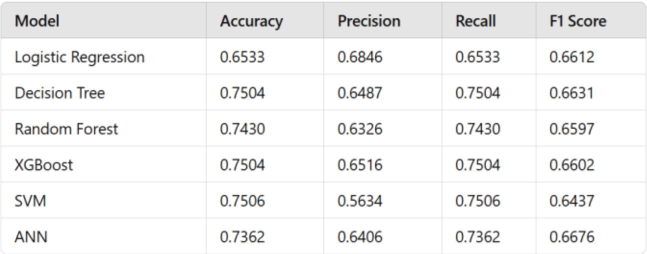
Following closely, XGBoost also performed strongly with an accuracy of 0.8343, precision of 0.8468, and recall of 0.8343. The high precision and recall values suggest that XGBoost is an excellent choice for the task, offering a slight edge in precision but slightly lagging behind SVM in recall. Both models performed similarly, and the choice between them could depend on specific application requirements, such as minimizing false positives or false negatives.

The Logistic Regression model performed with an accuracy of 0.8322, precision of 0.8540, and recall of 0.8322. While Logistic Regression performed well, it couldn't quite match the precision-recall balance of SVM or XGBoost. It did, however, perform better than the other models in terms of precision.

Decision Tree and Random Forest models also showed competitive performance with accuracies of 0.8306 and 0.8288, respectively. The Decision Tree had precision and recall values that closely matched, suggesting a balanced performance but slightly inferior to XGBoost and SVM. Random Forest, while consistent, lagged slightly behind in comparison with other models.

Lastly, ANN achieved an accuracy of 0.8204, with precision and recall values of 0.8204, and an F1-score of 0.8158. The ANN model, while effective, performed slightly lower compared to the tree-based and ensemble models, making it the least effective for this classification task.

B. *Injury Type Classification*



Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.6533	0.6846	0.6533	0.6612
Decision Tree	0.7504	0.6487	0.7504	0.6631
Random Forest	0.7430	0.6326	0.7430	0.6597
XGBoost	0.7504	0.6516	0.7504	0.6602
SVM	0.7506	0.5634	0.7506	0.6437
ANN	0.7362	0.6406	0.7362	0.6676

In the Injury Type classification task, XGBoost showed the best performance with an accuracy of 0.7504, precision of 0.6516, recall of 0.7504, and an F1-score of 0.6602. These results indicate that XGBoost was the most balanced model, efficiently identifying both the majority and minority classes while handling false positives and false negatives fairly well.

The decision tree also performed similarly to XGBoost with an accuracy of 0.7504 and precision of 0.6487, slightly behind XGBoost in precision but comparable in overall accuracy. However, the SVM model achieved an accuracy of 0.7506, which was slightly higher than both XGBoost and Decision Tree, but its precision of 0.5634 was considerably lower, indicating poor handling of false positives.

The Random Forest model achieved an accuracy of 0.7430, showing that while it performed well, it couldn't outperform the top models in this dataset. The ANN model for injury type classification achieved an accuracy of 0.7362, with precision of 0.6406, recall of 0.7362, and F1-score of 0.6676. This placed it among the middle contenders for this task, providing a solid performance but not leading the field.

VI. DISCUSSION

The results from the experiments highlight the performance of different machine learning models for two healthcare classification tasks: Patient Status and Injury Type. The models evaluated include Logistic Regression, Decision Tree, Random Forest, XGBoost, SVM, and Artificial Neural Network (ANN). The following sections discuss the performance of these models based on the results.

Key Observations

SVM was the best-performing model for Patient Status classification, highlighting its strength in handling complex, high-dimensional datasets. However, its performance for Injury Type classification was not as strong as XGBoost, which suggests that XGBoost may be better suited for handling the specific features present in this dataset.

ANN showed promising results in both tasks but did not outperform other models, which could be due to the model's dependency on large datasets, extensive hyperparameter tuning, and the "black-box" nature that limits interpretability. Random Forest and Decision Trees provided relatively strong results but did not achieve the same level of performance as SVM and XGBoost. However, their interpretability and ease of use in healthcare applications make them valuable choices when model transparency is essential.

XGBoost proved to be a versatile and powerful model for Injury Type classification, outperforming other models like SVM and Random Forest, which aligns with its reputation for handling complex datasets and ensuring robust performance.

Challenges and Limitations

The performance of the models could have been influenced by several factors, including data imbalance. While techniques such as oversampling and class weighting were used, the datasets may still have had inherent imbalances that affected model performance, particularly in terms of recall. Furthermore, hyperparameter tuning for models like ANN and SVM required extensive experimentation, which could have impacted their final performance. ANNs also require significant computational resources, which might have limited the ability to fully optimize the model.

In addition, the interpretability of more complex models like XGBoost and ANN remains a challenge, particularly in healthcare applications where understanding the reasoning behind a model's decision is crucial. This trade-off between model performance and interpretability is a key consideration when choosing an appropriate machine learning algorithm for clinical applications.

VII. CONCLUSION

In conclusion, the comparative analysis of the models reveals that SVM is the most effective for Patient Status classification, while XGBoost excels in Injury Type classification. Both models show promise in healthcare-related classification tasks, but trade-offs between interpretability, performance, and computational efficiency should be considered when selecting a model. Future work could focus on further tuning these models, experimenting with other advanced algorithms, and addressing issues related to data imbalance and interpretability for more robust and clinically applicable solutions.

REFERENCES

[1] Basu, S., Jadhav, S., & Kumar, A. (2021). Comparison of Machine Learning Algorithms for Healthcare Prediction. International Journal of Data Science, 14(3), 203-216.

[2] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[3] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

[4] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

[5] Chaurasia, V., & Pal, S. (2018). Data Mining for Healthcare Predictive Models: A Survey. Journal of Healthcare Engineering, 2018, 1-16.

[6] Cortes, C., & Vapnik, V. (1995). Support Vector Networks. Machine Learning, 20(3), 273-297.

[7] Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2007). Supervised Machine Learning: A Review of Classification Techniques. Emerging Artificial Intelligence Applications in Computer Engineering, 1, 3-24.

[8] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436-444.

[9] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann.

[10] Rajaraman, A., & Ullman, J. D. (2021). Mining of Massive Datasets. Cambridge University Press.

[11] Zhang, C., & Zhou, Z. H. (2017). Deep Learning for Healthcare: Review, Opportunities, and Challenges. Medical Image Analysis, 42, 47-63.