Automated Gait Analysis Using XGBoost and Isolation Forest for Anomaly Detection

Arssh Bajpai, Eric Mecca, Junxiang Lin
Northeastern University

August 12, 2024

Abstract

Gait data (data on how a person walks) is crucial in the detection and diagnosing of neurological and muscular diseases as well as treating and managing these conditions. However the data in its raw form is very unorganized and difficult to extrapolate and assess in an effective and accurate way. This is where the justification of our project comes in, using machine learning algorithms like XGBoost and Isolation Forests, we analyze the gait data to assist us in detecting anomalies. An anomaly in gait data typically constitutes an irregular step in terms of the distance from the previous time interval. We then take the data from these models and evaluate them using mean-squared error, mean absolute error, as well as R-squared metrics. Additionally we tune the hyperparameters (the starting values that dictate the extent and scale of learning) using GridSearchCV so that the models are then optimized.

1 Introduction

Most gait studies analyze and study the human walking, in order to diagnose conditions affecting gait such as Huntingtons or Parkinsons. Traditional methods to analyze gait are very rudimentary and often rely on a doctors own intuition and eyeballing of data. Our project aims to solve this, by analyzing and predicting metrics of gait using machine learning in order to give pin points on where a patient has irregularities, and by how much they do. Essentially we don't want to diagnose the patient with a neurological disorder instead we want to give doctors the tools and the information for them to make an accurate diagnosis.

2 Problem Statement and Methods

2.1 Problem Statement

The objective we have for this project is to create a comprehensive robust Machine Learning model that efficiently and accurately analyses gait in order to detect anomalies. To allow the processing of gait data so that it can be used efficiently in a clinical setting.

2.2 Methods

2.2.1 Data Preprocessing

The data is read from CSV files found online containing gait data from patients, we take the data from the files and put it into an array of tuples so that we can better analyze it and run it through our models.

2.2.2 Model Training

XGBoost: A learning method that is the extreme version of Gradient Boosting. We chose this because it allows for deeper learning over normal gradient boosting as well as greater speed. Additionally their are built in functions and libraries that allow us to use XGBoostRegression for our project .

Cross-Validation: The model is evaluated using cross-validation to ensure it generalizes well to unseen data. It also helps estimate the skill of our algorithm and help our model to predict data.

GridSearchCV: We use GridSearchCV so that we can tune the hyperparameters to their optimal settings for out model. That way we can make it work at optimal efficiency.

2.2.3 Anomaly Detection

Isolation Forest: We use isolation forests in order to predict and find anomalies in our residuals. We chose this because of its explicit nature is helping detection deviations from the norm, which is our explicit definition of an anomaly in our problem.

3 Algorithms and Mathematics

3.1 XGBoost

XGBoost (Extreme Gradient Boosting) is an optimized distributed gradient boosting library designed to be

highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework.

3.1.1 Gradient Boosting

Gradient Boosting is a machine learning technique for regression and classification problems that produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion and generalizes by optimizing a loss function. We use an implementation of this in XGBoost as its faster more efficient and more robust. We tested with Gradient Boost but found it too slow and not as precise for anomaly detection. with the additional hurdles of not having dedicated functions like XGBoost we abandoned it, but still use it conceptually in XGboost.

4 Objective Function

The objective function is crucial for guiding the training of our model. It helps in measuring how well the model is performing and directs the optimization process. It consists of a training loss term and a regularization term.

Training Loss The training loss evaluates how far the model's predictions are from the actual target values. The goal is to minimize this loss, which helps the model make accurate predictions. For regression tasks, the objective function commonly uses the squared error loss:

$$\text{Training Loss} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) \tag{1}$$

Here, L is a differentiable convex loss function, y_i is the actual target value, and $\hat{y}_i$ is the predicted value. The squared error loss is chosen because it penalizes larger errors more heavily, which is important when dealing with continuous data and ensures that the model focuses on reducing large prediction errors.

Regularization Term Regularization is used to prevent the model from becoming too complex and overfitting the training data. Overfitting occurs when the model learns noise in the training data rather than the underlying pattern. Regularization helps in achieving a balance between model complexity and performance:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2}$$

where γ penalizes the number of leaves in the trees, and λ penalizes large weights. Regularization is essential to ensure that the model generalizes well to unseen data and avoids overfitting.

5 Additive Training

Additive training is a method where new models are added sequentially to correct the errors made by the existing models. This technique helps improve the overall model by addressing the residual errors from previous iterations:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{3}$$

In this equation, $\hat{y}_i^{(t)}$ represents the prediction at stage t, and f_t is the new model added at stage t. This approach is crucial for enhancing the model's accuracy by incrementally correcting errors, leading to a more refined and accurate model.

6 Regularization

Regularization is vital for controlling the complexity of the model. It prevents overfitting by applying penalties to model parameters. In XGBoost, regularization is handled through:

- **Penalizing the Number of Leaves**: Trees with a large number of leaves are penalized to prevent overfitting. This helps in maintaining model simplicity and avoiding excessive complexity.
- **Penalizing Large Weights**: Large weights are penalized to prevent the model from becoming too sensitive to small changes in the data. This balance is crucial for maintaining model stability and generalization.

7 Evaluation Metrics

Evaluation metrics are used to assess the performance of the model and determine how well it predicts the target values. These metrics provide insights into the accuracy and effectiveness of the model.

Mean Squared Error (MSE) MSE measures the average squared difference between the actual values and the predicted values. It emphasizes larger errors, making it particularly useful when large errors are undesirable:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

MSE is chosen because it provides a clear measure of model accuracy by highlighting the impact of large errors, which is crucial for understanding model performance.

Mean Absolute Error (MAE) MAE measures the average magnitude of errors without considering their direction. It treats all errors equally, providing a straightforward measure of prediction accuracy:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

MAE is used to assess the average size of prediction errors and is chosen because it provides a clear and interpretable measure of accuracy, especially when errors of all sizes are equally important.

R-squared ($\mathbf{R^2}$) R-squared represents the proportion of the variance in the dependent variable that is explained by the independent variables. It provides an overall measure of model fit:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6}$$

R-squared is chosen because it indicates how well the model explains the variability in the data, providing a comprehensive measure of model performance.

8 Isolation Forest

Isolation Forest is a powerful algorithm used to detect anomalies, which are unusual or rare data points that deviate significantly from the norm. This method is particularly useful in various fields, including fraud detection, network security, and medical diagnostics.

Isolation The core idea behind Isolation Forest is based on the notion that anomalies are rare and distinct from normal observations. To identify these anomalies, the algorithm works by creating a series of decision trees, each of which splits the data randomly along different features and values. The key concept here is isolation: anomalies, being few and different, are more easily separated from the rest of the data compared to normal observations.

In practice, Isolation Forest builds multiple trees where each tree recursively partitions the data. The number of splits required to isolate a particular data point, or the path length from the root to the leaf, is crucial. Shorter path lengths indicate that a data point is more easily isolated, suggesting it is an anomaly. This method is efficient because it directly targets the process of isolation, which is simple and effective for anomaly detection.

Anomaly Score The anomaly score quantifies how much of an anomaly a data point is. It is derived from the average path length required to isolate the data point within the decision trees. If a data point is easily isolated, its path length will be short, and it will receive a higher anomaly score. Conversely, if it requires many splits to be isolated, it will have a lower score.

The anomaly score is calculated using the formula:

$$\text{anomaly\_score}(x) = 2^{-\frac{E(h(x))}{c(n)}} \tag{7}$$

In this formula:

- $E(h(x))$ represents the expected path length for isolating the data point x.
- $c(n)$ is a normalizing constant based on the average path length in a Binary Search Tree for a dataset of size n.

A lower score indicates a higher likelihood of the data point being an anomaly. This method of scoring is intuitive because it directly links the ease of isolation to the anomaly score, making it straightforward to interpret.

9 Experiments and Results

9.1 Experimental Setup

The experiments were designed to evaluate the effectiveness of the XGBoost model and the Isolation Forest algorithm on gait data.

Data We used gait data obtained from a CSV file, which contains measurements of various gait features for patients. This data forms the basis for training and testing our models.

Splitting To assess model performance reliably, the data was split into two sets: a training set and a testing set. The split was done with an 80-20 ratio, meaning that 80% of the data was used for training the models, and the remaining 20% was reserved for testing. This approach ensures that the model is trained on a substantial portion of the data while being evaluated on unseen data to gauge its generalization capability.

Metrics Several metrics were employed to evaluate the model's performance:

- **Mean Squared Error (MSE)**: Measures the average squared difference between actual and predicted values. It is chosen because it highlights large errors, which is crucial when precise predictions are necessary.
- **Mean Absolute Error (MAE)**: Measures the average magnitude of errors without considering their direction. It is chosen for its simplicity and interpretability.
- **R-squared ($\mathbf{R^2}$)**: Represents the proportion of variance in the target variable that is explained by the independent variables. It provides a comprehensive measure of model fit.

9.2 Results

Cross-Validation MSE The cross-validation process yielded a mean squared error (MSE) of 0.00040476. This value reflects how well the model performs in generalizing to new data by averaging the errors across multiple folds of the data.

Model Performance The model's performance metrics are as follows:

- **Mean Squared Error (MSE)**: 0.00047361. This value indicates the average squared difference between the model's predictions and the actual values. A lower MSE means better model accuracy.
- **Mean Absolute Error (MAE)**: 0.00689030. This metric shows the average magnitude of the prediction errors. It complements MSE by providing a direct measure of error size.
- **R-squared (R^2)**: 0.887608. This high value indicates that the model explains approximately 88.76% of the variance in the gait data, demonstrating a strong fit.

Anomaly Detection The Isolation Forest algorithm successfully identified several anomalies in the gait data. These anomalies might indicate unusual patterns in gait, potentially signaling health issues or conditions that require further investigation.

9.3 Visualization

The results were visualized to provide insights into model performance and anomaly detection:

- **Residuals and Anomalies**: A scatter plot was created to visualize the residuals, with anomalies highlighted in red. This plot helps in understanding how well the model's predictions align with the actual values and where anomalies occur.
- **Predicted vs Actual Values**: A line plot was used to compare predicted gait values with the actual measurements. This visualization shows how closely the model's predictions match the real data over time.

10 Discussion and Conclusion

The XGBoost model demonstrated a high level of accuracy in predicting gait data, as evidenced by an R^2 value of 0.8876. This indicates that the model effectively captures the underlying patterns in the data. The Isolation Forest algorithm also performed well, successfully identifying anomalies that could suggest potential health issues.

This automated approach to gait analysis is promising for clinical diagnostics. It offers a fast and reliable tool for healthcare professionals to assess gait patterns and identify anomalies, which can lead to early detection of possible health concerns, allowing for preventative metrics and procedures to help stop its spread.

10.1 Limitations

- **Data Quality and Quantity**: The performance of the model depends heavily on the quality and quantity of the data used. We also use a more rudimentary gait data for our analyses, rather than the comprehensive files that include movement seperated by limb and pressure sensitivity, due to data unavailability.
- **Clinical Validation**: We were able to identify possible points of anomalies and detected errors in the patients gait, but the use in a clinical setting is meant to be a starting point for analysis and diagnoses and we believe with our high level of accuracy in anomaly detection we have created that. However the usefulness cannot be determined by us as untrained physicians with no experience in diagnostics.