Behavioral Biometrics for User Authentication: Analyzing Cursor Movements

Mayank Pandey
mayank21264@iiitd.ac.in

Sunny Dhaka
sunny21429@iiitd.ac.in

Tushar Goel
tushar21431@iiitd.ac.in

Abstract

Through cursor movement patterns, we aim to create an advanced system for user categorization. In other words, capturing and studying the motion of the cursor by the registered users we want to differentiate their users according to how they perform a certain action. While user categorization is one of the main objectives, our technique is not limited only to that. Since the analysis of cursor movements can also find application in the detection and analysis of deviant activities, it aids in increasing the overall security of online activities. This approach is not only means of improving the authentication of users; it is also a booster of users' confidence in the online environment. Our development is an advance towards customization on the web while safeguarding the user in this fast changing world.
Github Link

1. Introduction

In today's digital landscape, where cyber threats evolve at an unprecedented pace, traditional authentication methods increasingly show their limitations. While passwords, two-factor authentication, and biometric systems remain prevalent, they often create friction in user experience and can still be vulnerable to sophisticated attacks. This research explores an innovative approach to digital security by focusing on an overlooked yet ubiquitous aspect of human-computer interaction: cursor movement patterns.

Every individual interacts with their computer in subtly unique ways, from the speed of their mouse movements to the trajectories they follow when navigating interfaces. These behavioral patterns, much like a digital fingerprint, offer a promising avenue for both enhanced security and improved user experience. Our research investigates how these distinctive cursor movement signatures can be leveraged to create a dynamic, unobtrusive security layer that operates continuously without requiring explicit user action.

The significance of this work extends beyond mere security enhancement. As online services become increasingly personalized, understanding and adapting to individual user behaviors represents a crucial frontier in human-computer interaction. By analyzing cursor movements, we aim to develop a system that not only identifies potential security threats but also adapts interface elements to match users' natural interaction patterns.

The findings of this research have implications for various domains, from e-commerce platforms seeking to prevent fraud to educational systems requiring continuous user verification.

2. Literature Survey

2.1. Assessing user behavior by mouse movements.

The paper by Matthiesen and Brefeld explores user identification through mouse movement analysis, taking a novel approach by treating it as a one-class problem rather than traditional multi-class classification. They utilized the Balabit Mouse Dynamic Challenge dataset, which comprises mouse movement data from ten users with 5-7 longer training sessions per user and multiple shorter test sessions, including data from out-of-sample users and simulated attackers.

The researchers evaluated two session-splitting methods: Time Difference Split (TDS) and Equal Number of Data Points Split (EDPS), comparing One-Class SVM, SVDD, Deep SVDD, and One-versus-Rest SVM approaches. The study concluded that mouse behavior is idiosyncratic, suggesting that improved user identification might be achieved by developing individualized feature representations for each user rather than using a universal approach [1].

2.2. User authentication based on mouse movement data using normalized features

This study presents a mouse dynamics-based user authentication system using normalized features. The researchers utilized the Recording User Input (RUI) logging tool to collect mouse movement. Three types of mouse ac-

tions were analyzed: Mouse Move, Point-and-Click, and Drag-and-Drop.

The system employed both Support Vector Machine (SVM) and Artificial Neural Network (ANN) classifiers. Results showed that SVM outperformed ANN, achieving a False Rejection Rate (FRR) of 1.1594 and False Acceptance Rate (FAR) of 1.9053 with a block size of 600 actions. The inclusion of the new direction-specific features significantly improved performance compared to using only traditional features, demonstrating the effectiveness of the proposed approach for behavioral biometric authentication [2].

2.3. User Activity Anomaly Detection by Mouse Movements in Web Surveys

The study "User Activity Anomaly Detection by Mouse Movements in Web Surveys" introduces a method to verify survey responses using machine learning. By analyzing mouse movement data from web surveys, it quickly assesses survey validity without examining individual answers. The study explores expert rules-based, LSTM-based, and HMM-based techniques. This approach identifies suspicious user activity and distinguishes it from genuine responses, improving survey data accuracy. A scoring system detects anomalies by considering various mouse movement factors, offering a robust alternative to traditional validation methods. This method enhances the reliability of survey data and provides deeper insights into user behavior [3].

3. Dataset

For our analysis, we used two distinct datasets: the User Interaction Dataset (UID) by Chao Shen and the Cursor Activity Dataset (MAD) by DFL. Together, these datasets consist of 42 attributes, which have been renamed for better clarity:

Action Type: Represents the nature of user actions (e.g., drag, drop, cursor movement, click), which have been label-encoded into values (0, 1, 3, 4) for further modeling.

Distance Traveled: Captures the total distance the cursor moved during user interactions.

Curvature Metrics: Includes measures such as average curvature, standard deviation of curvature, minimum curvature, maximum curvature, the Irregularity Index, and the Smoothness Index, which assess how smooth or irregular cursor movements were during an action.

Velocity Metrics: Encompasses metrics like average velocity, standard deviation of velocity, minimum and maximum velocity, and fluidity, which describe the speed and fluidity of cursor movements.

Angular Velocity Metrics: Covers average angular velocity, standard deviation of angular velocity, as well as the minimum and maximum angular velocity of cursor actions.

Deviation: Indicates the largest deviation observed in the cursor's movement trajectory during user interactions.

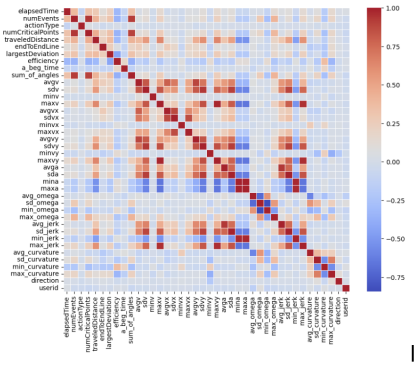

Figure 1. Heatmap

4. Data Preprocessing

4.1. Feature Selection

Given the dataset's complexity with its 42 features, we performed a comprehensive assessment of feature importance. Features deemed insignificant for the classification model were marked for potential removal.

4.2. Data Randomization

To enhance the model's performance, we began preprocessing by shuffling the dataset. Initially, the dataset consisted of user session records in a sequential order. By randomizing the data, we ensured that the model wouldn't learn irrelevant sequential patterns.

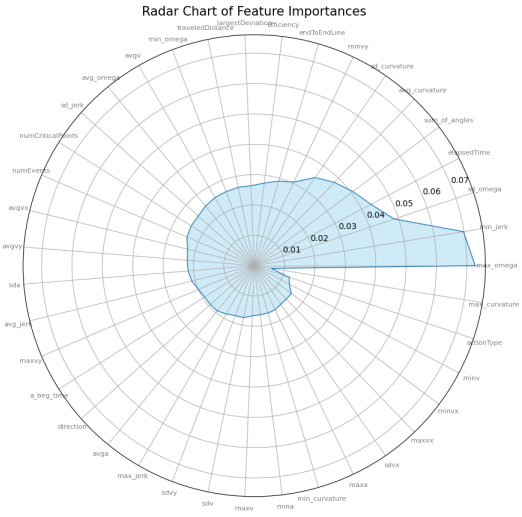

Figure 2. Radar chart

4.3. Addressing Missing Data

Interestingly, there were no missing or null values in the dataset. As a result, no imputation or removal of incomplete

records was necessary, simplifying the preprocessing stage.

4.4. Data Scaling

The dataset contained attributes with varying value ranges. To ensure uniform treatment of all attributes by the model, we applied normalization.

4.5. Anomaly Detection

Using Isolation Forest, we identified 1050 anomalies (5 percent of the data), which may represent unusual user behavior or technical issues, while the remaining 19950 points were classified as "Normal."

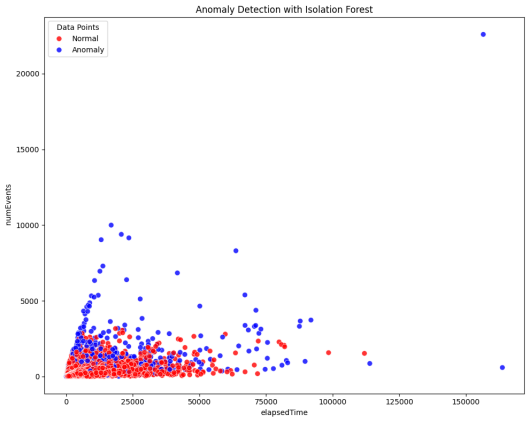

Figure 3. Anamoly

4.6. UMAP (Dimensionality Reduction)

We applied UMAP to the merged datasets to visualize their structure in 2D. UMAP revealed that the data is non-linearly separable, highlighting potential clusters of user behaviors or interaction patterns that were not apparent through linear methods.

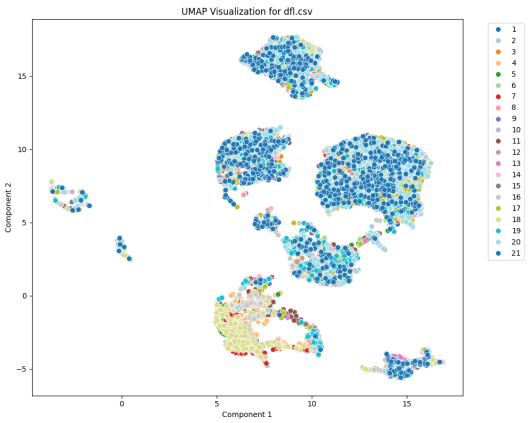

Figure 4

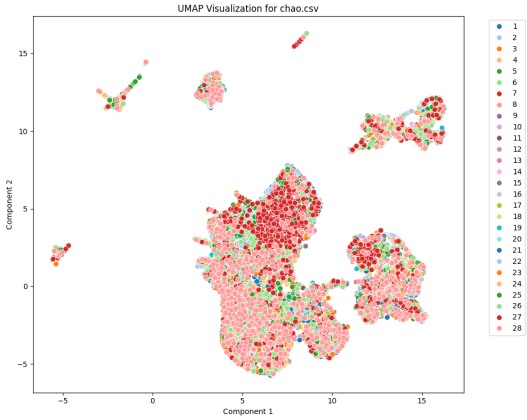

Figure 5

5. Methodology and Model Overview

5.1. Logistic Regression

Overview: Logistic Regression is a supervised machine learning model commonly used for binary classification tasks. It estimates the probability that a given input belongs to a particular class by fitting data to a logistic curve (sigmoid function). The model assumes a linear relationship between the input features and the log-odds of the binary outcome.

Hyperparameters: Important hyperparameter include C, which controls the regularization applied to the model and helps prevent overfitting and max-iter, which sets the maximum number of iterations for the optimization algorithm to converge.

5.2. Random Forest Classifier

Overview: Random Forest is an ensemble learning technique that combines multiple decision trees to make predictions. It averages the predictions from individual trees (via majority voting for classification tasks) to improve accuracy and reduce overfitting.

Hyperparameters: Important hyperparameters include n-estimators, which specifies the number of trees in the forest.

5.3. Support Vector Machine

Overview: SVM is a powerful supervised learning model used for both classification and regression tasks. It works by identifying a hyperplane that best separates the classes while maximizing the margin between them, making it particularly effective for high-dimensional data.

Hyperparameters: SVM hyperparameters include the

regularization parameter C, the kernel function (linear, polynomial, radial basis function), and kernel-specific parameters.

5.4. AdaBoost

Overview: AdaBoost is a popular boosting alogrithm. It works iteratively training. It works iteratively by training classifiers on the data and focusing on the samples that were previously misclassified, assigning them higher weights.

Hyperparameters: Hyperparameters include n-estimator, that specifies number of trees, learning rate and estimator which is usually a decision tree.

5.5. XGBoost

Overview: It is a powerful and efficient machine learning algorithm based on gradient boosting. It builds an ensemble of decision trees sequentially, where each tree corrects the errors of the previous one. It is known for speed and is popular because of preventing overfitting due to regularization.

Hyperparameters: Hyperparameters include n-estimator, learning rate, and regularization parameters.

5.6. Multi Layer Perceptron

Overview: It is a type of ANN cosisting of multiple layer of neurons, where each layer is fully connected to other layer. The hidden layers functions as feature extractors and thus learning complex patterns.

Hyperparameters: MLP hyperparameters include the hidden layer size, the kernel function (linear, polynomial, radial basis function), and kernel-specific parameters.

6. Results and Analysis

6.1. Logistic Regression

Accuracy: 17.69%
Analysis: Logistic Regression is typically more suited for binary classification tasks. However, in this multi-class scenario, the feature space might not be clearly distinguished, resulting in low performance, particularly when there is considerable overlap between the classes.

6.2. Random Forest Classifier

Accuracy: 79.4%
Analysis: By combining multiple decision trees, the Random Forest model improves robustness and reduces overfitting, leading to higher accuracy compared to a single decision tree.

6.3. Support Vector Machine

Accuracy: 20.1%
Analysis: SVM is typically effective in binary classification tasks. However, in this multi-class setting, the feature space may not be well-separated, leading to poor performance, especially with significant overlap between classes.

6.4. AdaBoost

Accuracy: 74.24%
Analysis: Since it works on weak classifers iteratively therefore a better accuracy was achieved in comparison to decision tree classifier.

6.5. XGBoost

Accuracy: 75.1%
Analysis: Since it also a boosting algorithm and incorporates regularization therefore it performed better than AdaBoost thus making it a better classifier.

6.6. Multi Layer Perceptron

Accuracy: 45.0%
Analysis: Since MLP might not have been trained for sufficient time with our chosen hyperparameters to learn complex data it might have led to a lower accuracy.

7. Conclusion

In our study on user categorization based on mouse movements, we employed six models: Logistic Regression, Random Forest, Support Vector Machines (SVM), AdaBoost, XGBoost, Multi Layer Perceptron (MLP).

Logistic Regression: Logistic Regression performed the poorest, with an accuracy of only 17%. This result may indicate that the linear boundaries established by logistic regression were insufficient to capture the complex patterns in the mouse movement data.

Random Forest Classifier: With an accuracy of 79.4%, the Random Forest model outperformed the Decision Tree. Random Forest, by leveraging multiple decision trees, is more robust and less prone to overfitting. This ensemble approach boosts accuracy by integrating the insights from several trees.

Support Vector Machine (SVM): SVM underperformed in this multi-class classification task, achieving only 20.1% accuracy. Even with hyperparameter tuning the highest accuracy achieved was 26.28%. Although SVM is effective in binary classification, it struggles in multi-class scenarios where significant interclass overlap exists, as was likely the case with this dataset derived from mouse movements.

AdaBoost: It achieved an accuracy of 74.24% as it works on weak classifers iteratively and thus results in a better accuracy.

XGBoost: It achieved an accuracy of 75.1% as it also incorporates regularization therefore it performed better than AdaBoost thus making it a better classifier.

Multi Layer Perceptron (MLP): MLP underperformed as it achieved an accuracy of 45.0% as it might not have been trained for sufficient time with our chosen hyperparameters to learn complex data which might have led to a lower accuracy.

In conclusion, a total of 6 models were used in our project. Ensemble methods like Random Forest,AdaBoost and XGBoost demonstrate promising potential for user classification tasks involving cursor movement data. However, further model optimization in MLP with hyperparameter tuning may improve accuracy, especially when dealing with complex feature overlaps. Achieving a balance between model complexity and generalization is essential for enhancing user classification performance and ensuring robust, reliable results.

References

[1] Bashira Akter Anima. User authentication based on mouse movement data using normalized features. *ResearchGate*, 2017. 1

[2] Unknown. Identifying user authentication and most frequently used region based on mouse movement data: A machine learning approach. *ResearchGate*, 2021. 2

[3] Unknown. User activity anomaly detection by mouse movements in web surveys. *Academia.edu*, 2021. 2