User Identification based on Cursor Movements

Samriddh Singh samriddh20466@iiitd.ac.in
Varun Parashar varun20482@iiitd.ac.in
Shivam Kurda shivam21419@iiitd.ac.in
Dev Mann dev21382@iiitd.ac.in

***Abstract*—Our project focuses on creating an advanced user categorization system inspired by the unique patterns of cursor movements during online interactions. By collecting and analyzing cursor movement data from registered users, we aim to classify individuals based on their distinct interaction styles. While user categorization is the primary objective, our approach has broader implications. It indirectly enables the detection of unusual or suspicious activities through cursor movement analysis, significantly enhancing online security. This strategy not only improves user authentication but also fosters user confidence in the digital realm. Our innovation represents a step toward a safer and more personalized online experience in an ever-evolving digital landscape.**

I. Introduction

It is crucial to protect online interactions in the current digital world. Traditional security measures, which often rely on usernames and passwords, must deal with growing threats from AI-driven bots and cunning fraudsters. In order to improve user classification and strengthen online security, this study offers a novel strategy that makes advantage of the unique characteristics of each user's cursor movement patterns while engaging in online activities. Our concept is based on the understanding that every user interacts with digital interfaces in a different way, much like a digital fingerprint. This uniqueness is most noticeable while using software and manipulating mouse cursors to navigate webpages. Traditional user identification techniques and security precautions frequently struggle to discern between legitimate and fraudulent activities, underlining the necessity for a novel strategy.

Our strategy has potential in a number of sectors, including e-commerce, online banking, and cybersecurity. By improving user classification through cursor movement analysis, it claims to reduce the dangers associated with illegal access, identity theft, and fraudulent transactions. Additionally, we create a more natural and user-friendly digital environment by seamlessly validating people based on their interaction habits. This project aims to reshape the future of online interactions by highlighting cursor movement patterns through our investigation of cursor movement analysis. It responds to the pressing need for reliable authentication systems in the digital age while maintaining user experience and privacy as the two most important considerations.

II. Literature Survey

A. Intrusion Detection Using Mouse Dynamics

The understudied field of mouse dynamics as a behavioral biometric for intrusion detection is examined in this research. It tests the Balabit dataset for imposter identification, which is one of the few publicly accessible datasets with unfettered mouse usage data. The authors partition the data into mouse move, point and click, and drag and drop operations, extracting numerous characteristics despite the difficulty of relatively brief test sessions. Mouse data is less sensitive than keyboard dynamics, allowing negative data to be collected for two-class classifiers. Notably, a maximum AUC of 1 is attained in training with only 13 actions, with drag and drop actions proving to be very useful for imposter identification[1].

B. A Study on Mouse Movement Features to Identify User

In order to identify users for information security concerns, the study "A Study on Mouse Movement Features to Identify User" investigates the use of mouse movement data. Mouse dynamics provide an affordable and user-friendly means of identification without the need for extra devices, in contrast to conventional techniques like passwords or PINs. The study emphasizes the significance of feature selection and computation by categorizing and describing several characteristics retrieved from mouse movement data utilized by various studies. Additionally, it emphasizes how useful mouse movement attributes are for identifying users and urges their ongoing usage and research for successful security applications[2].

C. *User Activity Anomaly Detection by Mouse Movements in Web Surveys*

A method for detecting user validity in survey replies using machine learning techniques is presented in the study titled "User Activity Anomaly Detection by Mouse Movements in Web Surveys". It makes use of mouse movement data gathered during web surveys to quickly determine a survey's general validity without looking at particular responses. For this job, the study investigates expert rules-based, LSTM-based, and HMM-based techniques. The technique uses mouse behavior analysis to identify suspect user activity and distinguish it from real replies, enhancing the accuracy of survey data. A scoring method is used to identify abnormalities by taking into account various mouse movement and behavior variables. The method offers an alternative to the current methods of validation for survey replies[3].

III. Dataset

For our analysis, we utilized two distinct datasets: - **User Interaction Dataset (UID) by Chao Shen** and **Cursor Activity Dataset (MAD) by DFL**

These datasets encompass a total of 42 attributes, now named more descriptively:

Action Type: It is denoting the nature of user actions (e.g., drag drop, cursor movement and click) which have been label encoded to (0,1,3,4) in order to model it further.

Distance Traveled: It represents the total distance covered by cursor movements carried out by the user.

Curvature Metrics: It includes average curvature, standard deviation of curvature, minimum curvature, maximum curvature, the Irregularity Index and the Smoothness Index which assesses the smoothness of cursor movements during an action.

Velocity Metrics: Encompassing average velocity, standard deviation of velocity, minimum velocity, maximum velocity of cursor movements and fluidity of cursor actions.

Angular Velocity Metrics: Covering average angular velocity, standard deviation of angular velocity, minimum angular velocity, and maximum angular velocity.

Deviation: Indicating the largest deviation observed during cursor usage by the user.

Jerk Metrics: Incorporating average jerk, standard deviation of jerk, minimum jerk, and maximum jerk.

Angle Summation: Sum of angles across all cursor trajectories.

Number of Events: Representing the count of cursor events within an action.

Initial Acceleration Time: Reflecting the acceleration time at the beginning of a cursor movement.

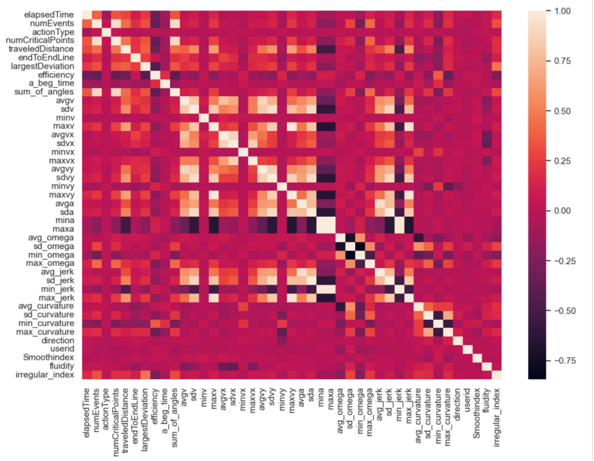


Figure 1. Heatmap

IV. Data Preprocessing

A. *Feature Selection*

Given the dataset's complexity with 42 features, we conducted a thorough evaluation of feature importance. Features with limited significance for the classification model were earmarked for potential removal.

B. *Data Shuffling*

To improve our model's efficiency, we initiated the data preprocessing by shuffling the dataset.

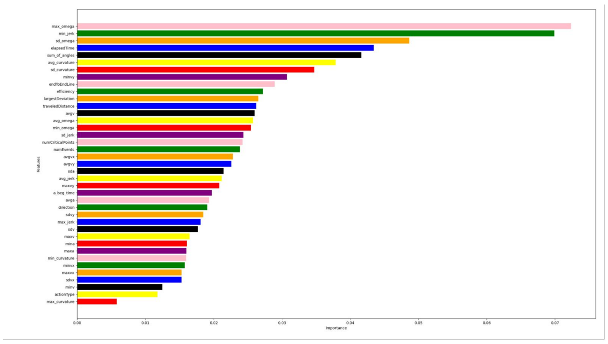


Figure 2. Feature importance

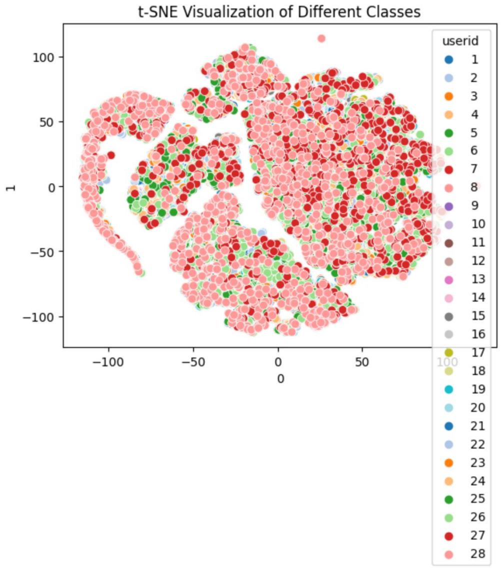
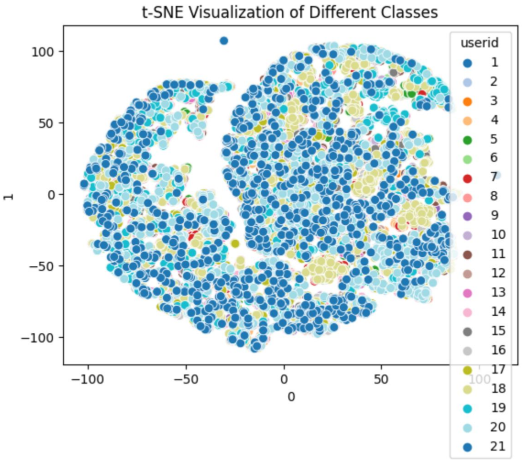


Figure 3. Chao shen t-sne



```
<Figure size 1000x800 with 0 Axes>
```

Figure 4. DFL T-SNE

Initially, the dataset contained user session records in a continuous fashion. Shuffling the data randomized the order of records, preventing the model from learning irrelevant sequential patterns.

C. Handling Null Values

Remarkably, there were no null or missing values in the dataset. Consequently, there was no requirement for imputation or the removal of incomplete data points, simplifying the preprocessing phase.

D. Normalization

The dataset's attributes exhibited varying value ranges. To ensure equitable treatment of these attributes by our model, we carried out normalization.

E. Checking for Linear Separability

Upon employing t-SNE separately for the 2 datasets which we went on to merge we found out that the data is linearly inseparable in the both the datasets.

V. Methodology, model details

A. Decision Tree Classifier

Description: A supervised machine learning model called the Decision Tree Classifier is employed for classification tasks. Recursively dividing the dataset into subgroups according to the most important property at each tree node is how it operates. These splits are based on specific standards, usually to reduce Gini impurity or increase information gain. For this particular model we also went on to incorporate Principal Component Analysis for varying component sizes.

Hyperparameters: Hyperparameters that we used were - (criterion='entropy',splitter='best')

B. Random Forest Classifier

Description: Several decision trees are used in the Random Forest ensemble learning technique to generate predictions. By averaging the predictions of each decision tree by majority voting, it works well for classification problems and reduces overfitting.

Hyperparameters: We used Randomized-SeachCV for hyperparameter tuning (n estimators, max features, max depth, min samples split, min samples leaf, bootstrap), with the Random Forest model being trained on the best parameters which we found.

C. SVM (Support Vector Machine)

Description: SVM is a potent supervised machine learning technique that may be applied to regression and classification problems. In order for SVM to function, a hyperplane must be identified that maximizes the margin between classes while effectively dividing the data into distinct classes. For high-dimensional datasets, it works well.

Hyperparameters: Since the accuracy which we arrived upon was too low because of the data being linearly inseparable, we didn't carry out hyperparameter tuning for this model in particular.

D. ADABoost Classifier

Description: The ADABoost Classifier is a form of ensemble learning, wherein trees are interdependent, growing successively on one another rather than operating independently.

Hyperparameters: base estimator=clf, n estimators=100

E. MLP (Multi Layer Perceptron)

Description: The Multilayer Perceptron (MLP) is a versatile type of feedforward artificial neural network belonging to the broader category of ensemble learning. In an MLP, neurons in one layer are connected to those in the next layer, forming multiple hidden layers that enable the network to capture intricate patterns and relationships within the data. **Hyperparameters:** hidden layer sizes=(300, 200, 100), max iter=300, random state=1

F. XGBoost

Description: XGBoost, an advanced gradient boosting implementation, optimizes by sequentially building trees to minimize the overall loss function with customizable weak learners, often decision trees. Its flexible architecture allows for handling deeper trees and employs regularization techniques to control overfitting. Assigning weights based on second-order partial derivatives of the loss function, XGBoost effectively manages overfitting and exhibits robustness against outliers. **Hyperparameters:** n estimators=100

VI. Results and analysis

A. Decision Tree Classifier

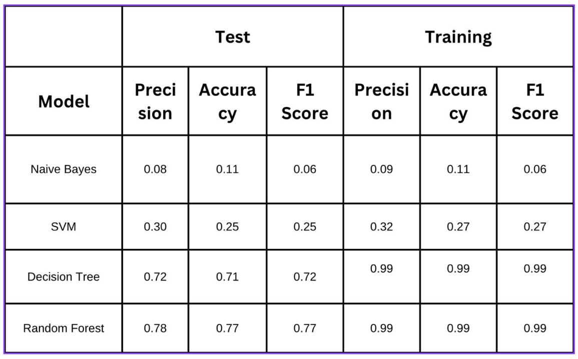
Accuracy: 71.97 percent

Analysis: It's possible that decision trees overfit the training set, which reduced their ability to generalize the test set.

B. Random Forest Classifier

Accuracy: 76.92 percent

Analysis: An ensemble of many decision trees called a random forest can increase resilience and decrease overfitting, therefore enhancing accuracy.



	Test			Training		
Model	Precision	Accuracy	F1 Score	Precision	Accuracy	F1 Score
Naive Bayes	0.08	0.11	0.06	0.09	0.11	0.06
SVM	0.30	0.25	0.25	0.32	0.27	0.27
Decision Tree	0.72	0.71	0.72	0.99	0.99	0.99
Random Forest	0.78	0.77	0.77	0.99	0.99	0.99

Figure 5. Model Results

C. SVM (Support Vector Machine)

Accuracy: 25.33 percent

Analysis: SVM is well-known for its efficiency in binary classification tasks, but because the feature space may not be well-separated(it might be the case in classes with larger sizes), it may not perform well in multi-class classification when there is a significant amount of overlap across classes.

D. ADABoost Classifier

Accuracy: 71.26 percent

Analysis: ADABoost operates on a set of weak classifiers that learn from each other, enhancing overall performance through collaborative learning.

E. MLP (Multi Layer Perceptron)

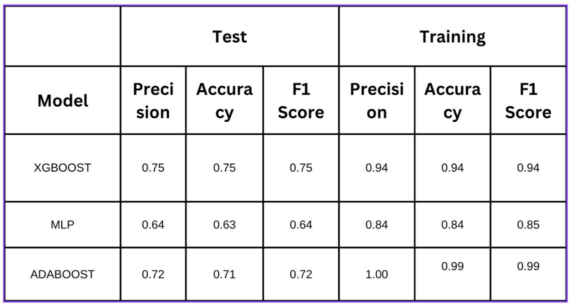
Accuracy: 66.54 percent

Analysis: The trade off between model complexity and running time is often a key consideration in practice. It involves choosing a model architecture that is complex enough to capture essential patterns in the data but not so complex that it leads to overfitting or impractical training times which is why we went with the chosen hyperparameters.

F. XGBoost Classifier

Accuracy: 74.57 percent

Analysis: XGBoost operates by employing regularization techniques along with the sequential building of trees thereby making it a robust classifier as is reflected by the Accuracy and the overall classification report.



	Test			Training		
Model	Precision	Accuracy	F1 Score	Precision	Accuracy	F1 Score
XGBOOST	0.75	0.75	0.75	0.94	0.94	0.94
MLP	0.64	0.63	0.64	0.84	0.84	0.85
ADABOOST	0.72	0.71	0.72	1.00	0.99	0.99

Figure 6. Model results

VII. Conclusion

A total of 6 classification models—Decision Tree, Random Forest, Support Vector Machines, ADABoost, MLP and XGBoost — were used in our user categorization study based on cursor movements (SVM).

The accuracy produced by the Decision Tree Classifier was 71.97 percent. Because decision trees might be prone to collecting noise in the training data, which limits their generalization to the test set, high training set accuracy may instead be the result of overfitting.

With an accuracy of 76.92 percent, the Random Forest Classifier fared better than the Decision Tree. Multiple decision trees combine to create random forests, which are often more resilient and less prone to overfitting. By utilizing the combined knowledge of several trees, this ensemble method improves accuracy.

The SVM performed poorly in this multi-class classification test, with an accuracy of only 25.33 percent. Although support vector machines (SVMs) are widely recognized for their efficacy in binary classification, they may encounter difficulties when interacting with feature spaces that have notable interclass overlap. It's possible that in this instance, the feature space created by the cursor movements wasn't enough separated for SVM to function at its best.

The ADABoost Classifier achieved an accuracy of 71.26 percent, leveraging collaborative learning among weak classifiers. This approach proved effective in enhancing overall predictive performance.

The Multi-Layer Perceptron (MLP) exhibited an accuracy of 66.54 percent, emphasizing the trade-off between model complexity and running time. Careful consideration of hyperparameters was essential to strike a balance, capturing essential data patterns without compromising computational efficiency.

XGBoost, with an accuracy of 74.57 percent, demonstrated its prowess in handling diverse datasets. By employing regularization techniques and sequentially building trees, XGBoost exhibited robust classification capabilities, providing a reliable framework for our user categorization study.

Our investigation concludes that ensemble techniques like as Random Forest as well as Neural Network Architectures have potential applications in the domain of user classification using cursor movements if given enough time to run by increasing the model complexity. To get more accurate user classification, balancing model complexity and generalization is still essential.

Thus, the best classifier which we arrived upon was the Random Forest Classifier with close to 77 percent accuracy which may further be improved by employing more hidden layers with more neurons in the MLP model which we trained. Now, the models mentioned above can be used to test whether a user has an authenticated profile or not.

References

[1] M. Antal and E. Egyed-Zsigmond, "Intrusion detection using mouse dynamics," *Journal Name*, vol. XX, no. XX, pp. XX–XX, XXXX, submitted to IET Biometrics on 23 May 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1810.04668

[2] M. Karim and M. Hasanuzzaman, "A study on mouse movement features to identify user," *Journal Name*, vol. XX, no. XX, pp. XX–XX, XXXX.

[3] A. Mastrotto, A. Nelson, D. Sharma, E. Muca, K. Liapchin, L. Losada, M. Bansal, and R. S. Samarev, "User activity anomaly detection by mouse movements in web surveys?" *Journal Name*, vol. XX, no. XX, pp. XX–XX, XXXX. [Online]. Available: https://www.columbia.edu/