Exploring Mouse-Based User Identification

Lev Rose
Washington University in St. Louis

Mohammad Rouie Miab
Washington University in St. Louis

Nicole Lucas
Washington University in St. Louis

April 2025

Abstract

The purpose of this paper is to investigate the feasibility of using mouse movement patterns solely as a method for uniquely identifying individual users on a webpage. Specifically, this paper aims to analyze raw cursor movement and basic webpage interaction data to determine if these elements can be used as distinctive biometric identifiers via feature engineering and a random forest machine learning model. Taking raw data from the *Mouse Movement Tracking* dataset from huggingface.com, motion-based features, spatial features, and pause and idle features are used in an attempt to classify individual users. As user identification via fingerprinting has become more prevalent in many aspects of tracking, for example, cookie syncing, investigating the feasibility of cursor-based fingerprinting is an important piece of knowledge to describe further the surface of vulnerability for users' personal privacy online.

1 Introduction

In the rapidly evolving landscape of online user-tracking, unique user identification has become a central piece of many tracking goals, such as cookie-syncing, targeted advertisements, and content suggestion. Online fingerprinting has become a major vector for these types of identification tasks. Whether that be directly through cookie sharing, JavaScript, CSS, or otherwise, users' identities and personal data is being collected and categorized. Traditional fingerprinting leverages computer attributes or such as installed fonts, screen resolution, or browser plug-ins, but a newer line of research investigates whether biometric trends in human-computer interaction, specifically mouse-cursor movements, can be used to identify users. The motivation for this method is primarily the persistence of the tracking method across sites and even on tracking blocking or private browsers.

This work investigates the viability of using a classification Machine Learning model to accurately identify specific users based on their cursor movements across web pages. This work takes the *Mouse Movement Tracking* dataset and applies feature engineering to create new, highly focused metrics that will uniquely describe a user's mouse movement tendencies. These extracted features will then be used in a Random Forest model to evaluate the data and report on a classification task.

2 Background and Related Work

Significant work has been done in the field of cursor tracking, in large part related to the field of Human-Computer Interaction. Much of the work is more focused on psychological trends of user attention, focusing on color and engagement metrics. For example, Huang et al.[1] pairs cursor movement with eye-tracking to determine user focus on a given webpage. Additionally, Warnock & Lalmas [2] investigate fur-

ther into cursor-tracking's scalability in conjunction with other tracking methods and traditional web analytics. In addition to the technical work done in this area, ethical works have also been produced to discuss the advancement of tracking technologies. For example, Leiva [3] addresses the privacy concerns that come with mouse tracking and argues for more regulatory practices to protect users from exploitation and unauthorized data collection.

For our work, we also implement a Machine Learning Classification Model. The model we chose is the Random Forest Classifier. A Random Forest is an ensemble machine learning model that utilizes the autogeneration of many individual decision trees that work together to decide on a final classification among a series of options or classes. Each decision tree is a "weak classifier" and makes predictions by repeatedly asking yes/no questions about the features of the input data to try to narrow down the result. These features are randomly assigned to each decision tree, and each tree individually attempts to guess the user's identity among a set of given options. While a single decision tree can easily overfit and memorize the features it's given without learning real patterns, a Random Forest builds many trees. Each tree is then trained slightly differently, being given a random subset of the features of the training data. In the context of our problem, when a new mouse movement's feature-values are input for classification, the Random Forest passes it through each decision tree and collects votes from each tree on who they think the user is. After which, the majority vote wins and becomes the model's final prediction.

3 Methodology

3.1 Dataset

The *Mouse Movement Tracking* dataset, originally collected by Dan Petrovic (DejanSEO) and hosted on Hugging Face, provides detailed recordings of user mouse activity across web sessions. Prior work has utilized this dataset for tasks such as user behavior modeling, predictive analytics, user interface optimization, and fraud detection.

The dataset contains approximately 686,000 individual mouse events, stored in Parquet format. Each event captures both spatial and temporal characteristics of user interaction, including cursor position, timestamps, and screen attributes. The data is provided as a single training split without a predefined validation or testing partition. Mouse movements are recorded during free-form browsing behavior, encompassing both periods of active interaction and intervals of idleness.

3.2 Feature Engineering

The feature engineering process involved a multi-stage transformation of raw interaction logs into structured, informative representations. This process is essential for revealing latent behavioral patterns that are not immediately apparent in the original data but are potentially highly discriminative for differentiating users.

A. Data Cleaning and Preprocessing

Initial preprocessing steps were applied to ensure the integrity and reliability of subsequent analyses. Missing values were identified and removed to preserve consistency across user sessions. Basic noise filtering was performed: cursor records with zero movement in both distance and time were excluded, as these events likely corresponded to spurious readings rather than genuine user actions. This cleaning process provided a more robust foundation for downstream feature extraction by minimizing the influence of outliers and measurement artifacts.

Feature	Description
session_id	Unique identifier for a user session
timestamp	Event occurrence time (Unix timestamp in milliseconds)
datetime	ISO-formatted human-readable timestamp
type	Event type (enter, click, leave, etc)
x,y	Cursor position coordinates on the screen
screen_width and screen_height	User's screen dimensions during the session
time_delta	Time elapsed since the previous event (milliseconds)
x_prev, y_prev	Cursor coordinates from the preceding event
dx, dy	Relative movement in the x and y directions
distance	Euclidean distance between consecutive cursor positions

Table 1: Feature descriptions for the Mouse Movement Tracking dataset.

B. Feature Creation

Upon establishing a clean dataset, feature engineering was conducted to derive a comprehensive set of descriptors targeting four principal categories: kinematic, directional, spatial, and temporal attributes. These engineered features aim to capture different facets of user behavior, enabling a more nuanced and distinctive profile for each session.

To model the dynamic properties of user interactions, we calculated the instantaneous speed of the cursor, defined as the distance traveled divided by the elapsed time between two events (speed=distance/time_delta). Building on this, we computed the acceleration as the rate of change of speed between successive points (acceleration=speed.diff()/time_delta). These kinematic features characterize the physical movement style of users, capturing traits such as smoothness, hesitancy, and abruptness, which may vary systematically across individuals.

Directionality served as another important behavioral signal. We computed the movement angle relative to the horizontal axis using the arctangent function arctan2(dy,dx). To capture variations in path sharpness, we further derived the angle difference between successive movements. Frequent or abrupt directional changes can serve as distinguishing characteristics, as users often differ in their degree of movement smoothness or hesitation. Together, these directional features provide a richer understanding of how users navigate the screen environment.

Spatial interaction patterns were represented by partitioning the screen horizontally into three zones — left, middle, and right — based on the x-coordinate values. Each cursor event was assigned to one of these zones, and session-level distributions were computed by normalizing the frequency of cursor points within each quadrant. Spatial tendencies often reflect ergonomic habits, dominant hand preference, or interaction strategies, offering another behavioral signature. Additionally, curvature was introduced as a derived spatial feature, defined as the ratio of the angle difference to the corresponding traveled distance. This feature quantifies the linearity or "curviness" of movement paths, further enriching the behavioral profile.

Given the inherently temporal nature of mouse movements, we introduced features to capture the rhythm of user interaction. An idle state was defined whenever the time delta between consecutive events exceeded a threshold value. Each event was flagged as active or idle, allowing the computation of an idle ratio for each session (i.e., the proportion of idle events). These temporal patterns provide insights into user cognitive rhythms, such as pausing to read, reflect, or navigate. We also aggregated session-level timing metrics, including total active time and the number of idle periods, to capture broader engagement trends.

C. Feature Aggregation Following the construction of point-level features, a session-level aggregation process was applied to generate a fixed-length feature vector for each user session. This transformation was essential for preparing the data for traditional machine learning workflows, which typically assume uniform input dimensions. For continuous features — including speed, acceleration, angle difference,

and curvature — we computed a series of statistical aggregations: the mean, standard deviation, maximum, and minimum values across all recorded events within a session. These statistics encapsulate the central tendency, dispersion, and range of user behaviors during interaction. For categorical features such as quadrant usage and idle state, we computed normalized frequencies or proportions to capture relative patterns independent of absolute session length. For instance, the quadrant distribution vector for each session indicates the proportion of cursor events within each screen region, while the idle ratio quantifies the proportion of idle versus active events. By applying this aggregation strategy, we effectively transformed highly variable-length sequences of raw interaction logs into standardized, compact session-level descriptors. This representation preserves essential behavioral dynamics while enabling compatibility with a wide array of supervised learning models, such as logistic regression, support vector machines, and tree-based classifiers.

Feature Justification
The selection and design of features were motivated by prior empirical research and cognitive theories of motor control, attention, and interaction behavior, with the objective of maximizing discriminative power across users. Each category of features was crafted to capture distinct aspects of user behavior that are both stable over time and uniquely individualized.

Kinematic features Features such as speed and acceleration were engineered to model the dynamic characteristics of mouse movements. Human motor behavior exhibits significant individual variability in terms of movement smoothness, hesitation, and control precision. Users differ consistently in their preferred movement velocities and their rates of acceleration or deceleration, making these metrics strong predictors of user identity. Fast, erratic cursor trajectories contrast sharply with slower, more deliberate movements, offering a reliable basis for classification.

Directional features Directional attributes, including movement angles and angular changes between consecutive movements, capture spatial navigation tendencies. Some users exhibit highly linear movement patterns characterized by small angular deviations, whereas others demonstrate more meandering paths with frequent sharp turns. These patterns reflect underlying strategies in visual search, ergonomic positioning, and interaction planning, all of which are known to vary idiosyncratically across individuals.

Spatial features Spatial interaction tendencies were quantified through quadrant usage distributions and curvature measurements. A user's preference for specific regions of the screen — whether gravitating towards the left, center, or right — can be influenced by dominant hand usage, visual scanning strategies, and even physical screen setup. Curvature, as a measure of movement path "curviness," further distinguishes users who prefer direct trajectories from those whose movements are more circuitous. Both features collectively provide spatial signatures that are difficult to consciously imitate or disguise.

Temporal features Temporal dynamics, including idle ratios and inter-event time intervals, were incorporated to capture differences in cognitive processing and decision-making styles. Periods of cursor inactivity may correspond to reading, reflection, or hesitation, whereas uninterrupted active movement indicates a more impulsive or goal-driven interaction style. These temporal rhythms are highly individualized and often stable across different browsing or task contexts, making them particularly valuable for behavior-based user identification.

3.3 Model Design

To achieve the monumental task of session-level user identification based on mouse movement and clicking behavior alone, we implemented a Random Forest Classifier machine learning model. The reason we decided to choose this type of machine learning model was that Random Forests are notoriously resilient to noise, scalable to a high number of features, and ro-

bust against overlapping and irrelevant features. In addition, it lacks the significant dataset size demands and ultra-long training times associated with deep neural networks. These are all practical necessities that fit well with our circumstances, given the lack of publicly available large-scale clean datasets. To the contrary, we do acknowledge the limitations of this approach, especially given that this approach may not fully preserve the time-series element of the data. To overcome this, many of our features aimed to preserve relevant time-series elements of the data by capturing a variety of trends, patterns, and temporal tendencies rather than relying solely on averaging and simple summarization. Given that our original dataset contained only one session per user, data augmentation was necessary to create usable train/test set splits. To do this, we split each original session into five smaller sub-sessions, which ensures our data includes sufficient per-user data for training and evaluation. When it comes to our model's hyperparameters and tuning, we adjusted the number of trees (`n_estimators`) and the maximum depth of the trees (`max_depth`). These were automatically tuned using GridSearchCV with 3-fold cross-validation, and tuning was set to prioritize and maximize ROC AUC scoring. As is typical, training and evaluation were performed using an 80/20 train/test split, with stratified sampling to preserve the class balance across a total of 566 individual users.

4 Evaluation

4.1 Evaluation Metrics

In the evaluation of our model's performance on our high-class-degree, 566-user classification task, we utilized multiple standard metrics:

- Top-1 Accuracy
- Top-k Accuracy
- Macro Precision, Recall, and F1 Score
- ROC AUC (One-vs-Rest)

Top-1 accuracy measures the proportion of test sessions where the model's first guess was correct. Top-k measures the proportion of sessions where the true user was among the model's top k guesses. Macro precision, recall, and F1 are still the standard confusion matrix performance metrics averaged equally among all users, but utilizing them in our context gets a bit tricky and requires nuance. Traditionally, precision, recall, and F1 are invented for binary classifications or small-class problems such as simple spam email classification. This is because binary settings fit naturally into the simple 2x2 confusion matrix structure of true positives, true negatives, false positives, and false negatives. However, our case is not just non-binary and multi-class, it's an exceedingly high class-degree of over 500. When you have a defined precision and recall in this context, each class has to be scaled down to a binary analysis, conceptually, comparing itself with all other classes, also known as One-vs-Rest (OvR). As a result, precision and recall scores tend to drop substantially as the number of classes increases to high numbers like 566. At this level, the interpretation of precision and recall scores becomes harder and requires other complementary metrics and additional analysis, and is often compared with random chance prediction of 0.17% ($= \frac{1}{566}$). This is why, in addition to all these metrics, we also used Macro ROC AUC (OvR) (Receiver Operating Characteristic curve Area Under the Curve One-vs-Rest). ROC AUC (OvR) measures how well the model ranks the correct user higher than incorrect ones, averaged across all classes. More sensibly, this is a measure of how well the multi-class classifier separates each single individual class from the rest (One-vs-Rest).

4.2 Results

Normally, for classifiers, a top-1 accuracy of 28% is low, but it's important to note that random guessing across 566 users would yield only a 0.17% accuracy. As a result, this model's performance is a significant improvement, over 160 times better than random chance, despite the small dataset and inherent high complexity of the problem. The achievement of a top-5 accuracy of 45% indicates that, while the

Metric	Score
Accuracy	28%
Top-3 Accuracy	39%
Top-5 Accuracy	45%
Macro Precision	21%
Macro Recall	28%
Macro F1 Score	23%
ROC AUC (OvR)	0.932

Table 2: 566-class Random Forest Classifier Performance

model's first guess was incorrect, it ranked the correct answer among its top guesses in nearly half the cases. Furthermore, the ROC AUC (OvR) result of 0.932 indicates that the model successfully learned meaningful patterns that could effectively distinguish users, even while in the presence of high noise and cross-session variability.

4.3 Discussion and Interpretation

When interpreting the results of this project, it's important to consider the nuances in its scope and limitations. In particular, several challenges arose throughout this project:

- **Overall Task Complexity**
  Generally, mouse movement data is highly complex and inherently noisy, making fingerprinting a difficult task. In addition, human behavior can vary somewhat between sessions, especially given the potential inherent variances in emotional state, task type, and hardware.

- **Data Constraints**
  Most of the problems we ran into were associated with the inherent constraints of the dataset we used. Firstly, there's only one user per session, which requires us to augment and split each user into multiple sub-sessions, reducing the integrity of the data and amplifying any existing noise. In addition, most of the sessions had holes in the data points, bots instead of real users, or just simply not enough data points to augment. Finally, each session is extracted from the user performing a random series of tasks of random length, which adds to the inherent challenge of extracting real habitual user tendencies.

- **Model Limitations**
  Random Forests work incredibly well for clear, structured-feature contexts, but they struggle to learn new details and representative attributes from raw sequences, especially when compared with neural networks. In addition, with our approach, much of the time-series data may be overlooked, given that direct time-series information is not incorporated into the training. Finally, in most fingerprinting contexts, it's necessary to add a new user in a low-cost method, and a Random Forest requires retraining every time a new user is added.

Given these challenges, the actual achieved performance is seen more as a proof-of-concept than a real-world-feasible product. Our results demonstrate a foundation for future exploration of things like more sophisticated neural networks, particularly ones that incorporate time-series directly and embed mouse movement trajectories into vector representations. For example, deep neural network (DNN) models trained using triplet loss or contrastive learning could learn, more directly, user-specific embeddings that capture the more nuanced and generalizable, individually unique styles hidden in movement data.

5 Ethics

The ethical discussion around tracking has been a long and intricate one. In the making of this paper, we acknowledge the privacy issues that stem from user tracking broadly, and the greater danger that mouse tracking might pose. We hope that through the research done in the area of mouse tracking and subsequent user identification, others will make efforts to protect users from this attack. While this paper only addresses the viability of this tracking vector, as new methods continually emerge to undermine the privacy of users, we are hopeful that the privacy community will stay a step ahead of these new methods by conducting research to evaluate these methods

and proactively protect against them.

6 Future Work

In this paper, we have explored mouse-based identification, and we feel that there are many ways future work could be taken to learn more deeply about this topic. The first testing that should be done is real-world testing with a significant sample size and an appropriate testing situation. Although time did not allow us to explore this testing first-hand, we hope that designing a multi-site testing tool might allow more benefits and gaps of the mouse-based identification methodology we explored. Additionally, other methods of biometric tracking may be viable for exploitation, such as typing habits, vocabulary habits, and others that could be explored similarly.

7 Conclusion

This work explored the feasibility of using mouse movement patterns as a means of uniquely identifying users through behavioral fingerprinting. By applying targeted feature engineering techniques to the Mouse Movement Tracking dataset and training a Random Forest classifier, we demonstrated that mouse-based user identification is surprisingly effective, pointing to the plausibility of implementation. The model achieved a top-1 accuracy of 28% and a ROC AUC (OvR) of 0.932 across a 566-user classification task. This represents a substantial improvement over random guessing and highlights the presence of highly individualized patterns embedded in user interaction behavior. While the results validate the concept of cursor-based fingerprinting, several limitations, including dataset size, session variability, and the lack of sequential modeling, still make implementation costly and difficult. Nevertheless, the findings raise important concerns about the expanding attack surface against users' online privacy. As traditional tracking methods become increasingly regulated, behavior-based identification techniques like mouse tracking may offer new vectors for user profiling that are harder to detect and prevent.

References

1. Huang, Weidong. (2007). Using eye tracking to investigate graph layout effects. Asia-Pacific Symposium on Visualisation 2007, APVIS 2007, Proceedings. 97-100. 10.1109/APVIS.2007.329282.

2. Warnock, David, & Lalmas, Mounia. (2015). An exploration of cursor tracking data. arXiv:1502.00317 [cs.HC]. https://arxiv.org/abs/1502.00317

3. Leiva, L. A., Arapakis, I., & Iordanou, C. (2021). My mouse, my rules: privacy issues of behavioral user profiling via mouse tracking. https://doi.org/10.48550/arxiv.2101.09087

8 Github Link

`https://github.com/nicoleclucas/mouse_data_notebook`