

ORIGINAL RESEARCH

Open access

# Continuous User Authentication via Mouse Dynamics Using Machine and Deep Learning Models

Omar Khalid<sup>1\*</sup>, Sara Nadeem<sup>1</sup>, Bilal Farooq<sup>2</sup>, Hina Saeed<sup>1</sup>

## Abstract

Conventional authentication mechanisms, such as passwords, are becoming less effective due to technological progress and evolving attack methodologies. To address this vulnerability, continuous authentication has emerged as a viable approach, wherein a user's identity is perpetually monitored after initial login to ensure an unauthorized individual has not hijacked the session. Mouse dynamics, which quantifies the behavioral patterns of a user's cursor movements, represents a highly promising biometric indicator for these ongoing verification models. This paper extends our prior research by benchmarking a 40-user dataset across three machine learning and three deep learning models. Our investigation covers two distinct evaluation frameworks: binary classification for individual authentication—where a 1-dimensional convolutional neural network (1D-CNN) yielded the highest performance, achieving a peak average test accuracy of 85.73% for the top-10 users—and multi-class classification utilizing an artificial neural network (ANN), which achieved an unprecedented peak accuracy of 92.48%, establishing a new performance benchmark for this specific dataset.

**Keywords** Machine learning, Deep learning, Mouse dynamics, Continuous user authentication, Multi-class classification

\*Correspondence:

Omar Khalid  
omar.khalid@kfupm.edu.sa

<sup>1</sup> Department of Continuous Authentication and Mouse Dynamics, Faculty of Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

<sup>2</sup> Department of Machine and Deep Learning for Authentication, Faculty of Engineering, Qatar University, Doha, Qatar

## Introduction

An ideal user authentication framework needs to be versatile, computationally lightweight, capable of near real-time execution, and, above all, highly accurate. Among the various authentication paradigms previously explored, biometric systems have emerged as the leading candidate for next-generation security architectures. Biometric authentication is generally categorized into two primary paradigms: physical biometrics and behavioral biometrics [1]. Physical biometrics utilizes distinctive anatomical traits for identity verification, leveraging technologies such as fingerprints [2], iris patterns [3], and voice recognition [4]. Conversely, behavioral biometrics leverages the premise that individual human action patterns during specific activities are sufficiently unique to serve as identifiers, exemplified by touch dynamics [5], keystroke dynamics [6, 7], and the core subject of this investigation: mouse dynamics. When compared with physical alternatives, behavioral biometrics has drawn substantial interest because of its broader, large-scale utility, minimized user friction, and independence from specialized external sensors. Furthermore, mouse dynamics has established itself as an ongoing, low-overhead, and seamless mechanism for dynamic identity verification [8-11].

Additionally, the extensive data volumes generated within the context of mouse dynamics facilitate the deployment of advanced machine and deep learning architectures. Domains characterized by rich datasets have experienced remarkable performance gains upon integrating machine and deep learning methodologies. Machine learning uncovers hidden patterns within massive datasets that elude human detection, offering robust adaptability across diverse domains; nevertheless, it depends heavily on manual feature engineering. Consequently, discovering the ideal alignment of hyperparameters, engineered features, and data preprocessing techniques for a specific dataset remains a complex challenge. In contrast, deep learning frameworks employ a multi-layered hierarchical architecture to automatically construct abstract representations from raw inputs. This eliminates the requirement for hand-crafted features, empowering the model to form a comprehensive interpretation of the underlying data rather than being constrained by predefined feature sets. While deep learning offers enhanced algorithmic autonomy, it introduces distinct challenges, including vanishing or exploding gradients, optimization convergence issues, and reduced transparency—frequently characterized as the “black box” dilemma of deep learning.

Because mouse dynamics research is still developing and public data remains limited, the true efficacy, full systemic impact, and inherent limitations of these algorithms within this domain are not yet completely understood. In particular, mouse dynamics has traditionally been evaluated as an authentication tool in low-stress, routine computing environments. Although encouraging outcomes have been documented under these conditions, the comprehensive utility of mouse dynamics demands rigorous testing across more diverse and challenging deployment frameworks. When combined with machine and deep learning, mouse dynamics holds the potential to act as an affordable, seamless behavioral biometric for user verification. Consequently, this study focuses on benchmarking three machine learning and three deep learning paradigms utilizing an expanded, unique dataset originally presented in our baseline research [12]. This specific dataset captures the mouse telemetry of 40 individuals participating in a high-intensity computing activity for a duration of 20 min. Existing open-source mouse dynamics repositories, including the Balabit [13] and TWOS [14] benchmarks, are characterized by limited cohorts and routine, low-engagement administrative tasks. Assessing these six models against data harvested from a highly dynamic and intense environment offers a more critical analysis of how machine and deep learning interface with mouse dynamics, while shedding light on prospective research avenues needed to advance the field. Our findings indicate that the 1D-CNN excelled in binary verification, whereas the ANN demonstrated superior performance in multi-class classification, with comprehensive results. The primary contributions of this research are summarized below:

- Improve upon our previous work by introducing and analyzing a larger dataset;
- In addition to random forests, two new machine learning models and three new deep learning models are evaluated on the dataset;
- Introduce an artificial neural network for multi-class mouse dynamics user classification.

## Background

This investigation expands upon the conclusions of [12], where it was posited that a data collection setting offering users higher degrees of freedom and behavioral flexibility would inherently manifest more identifiable features within individual interaction profiles. This was initially executed by gathering telemetry from 10 subjects during independent 20-min sessions of playing Minecraft on a standardized laboratory workstation, followed by training binary random forest estimators for identity verification. For every discrete mouse event (specifically, cursor movements or clicks), a dedicated background monitoring script recorded the associated UNIX timestamp, the X and Y coordinates of the cursor, and the corresponding Subject ID. Raw telemetry metrics, illustrated in Figure 1, lack the granular detail required for machine learning models to execute reliable authentication, which necessitates the introduction of the previously discussed data filtering and feature extraction phases. Clusters of 10 individual mouse events were aggregated into a distinct structural unit designated as a “mouse action,” serving as the primary input for the random forest classifier. While alternative frameworks have utilized flexible mouse action lengths—frequently scaling to orders of magnitude significantly larger than

10—demonstrating comparable performance metrics with a shorter block length suggests that our dataset encapsulates a higher density of behavioral information within a smaller window, a key finding established in [12]. Supplementary behavioral features—encompassing velocity, acceleration, jerk, angular velocity, alongside the geometric angle and spatial length of the cursor trajectory—were derived from each mouse action along with their corresponding descriptive statistics. These were compiled into a feature vector paired with the respective target label. Consequently, 10 separate binary random forests were optimized for individual user validation, utilizing a completely balanced distribution of positive and negative training samples. A positive case represented an authentic user action validating ongoing system access, whereas negative cases simulated fraudulent imposter interactions intended to trigger heightened monitoring or immediate access denial.

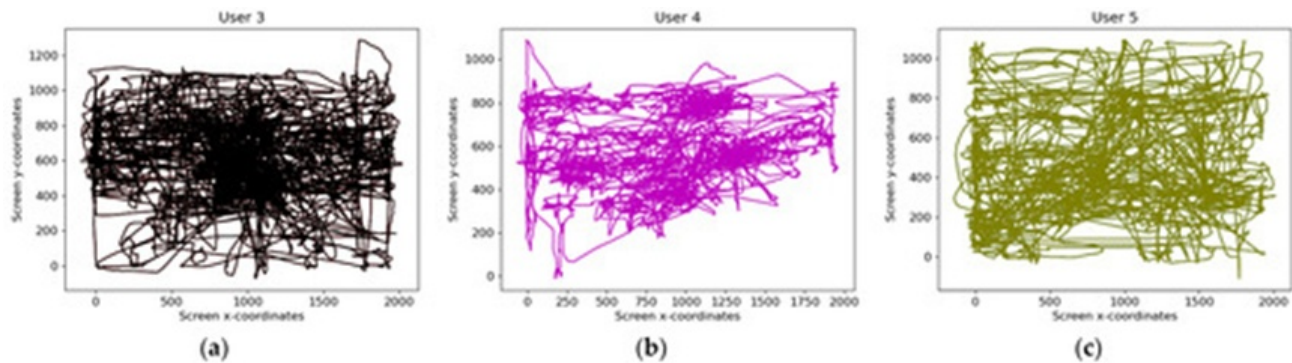


**Figure 1.** (a) Visualization of raw mouse from User 3's session; (b) visualization of raw mouse from User 4's session; and (c) visualization of raw mouse from User 5's session.

Numerous data-processing and architecture-design frameworks in [12] were modeled after the pioneering study that established mouse dynamics as a credible behavioral biometric [15]. In [15], the investigators introduced mouse dynamics as an innovative modality for human-computer interaction capable of serving as a substitute security measure. In contrast to traditional security paradigms where user verification depends on what "one knows" (such as PINs or passwords) or what "one has" (such as physical keys or token cards), mouse dynamics validates identity based on what "one is." Consequently, non-reproducible behavioral biometrics like mouse dynamics are exceptionally resistant to spoofing and represent excellent candidates for user authentication systems. Mirroring the observations in [16], the researchers in [15] highlight that because human behavior naturally shifts over time, behavioral biometrics must be engineered with sufficient elasticity and robustness to accommodate evolving user patterns. Reference [15] also outlines foundational data preparation and performance evaluation standards, such as extracting secondary features from raw data streams, adopting false positive, false negative, and equal error rates as assessment benchmarks, and aggregating mouse movements into variable-length action sequences. Additionally, it provides baseline benchmarks using traditional statistical learning techniques to guide subsequent comparative research.

Despite the relative youth of this biometric modality, the discipline has advanced significantly since the publication of [15]. Most prominently, the systematic performance advantages that machine learning models hold over older statistical approaches are highly evident within mouse dynamics research [17]. Furthermore, [17] introduced novel engineered features specifically for authentication pipelines utilizing mouse dynamics and quantified their respective gain ratios. The ongoing development of innovative, more descriptive attributes compared to those in [15], along with analyzing their exact influence on classification accuracy, is vital for the progression of the field and remains a heavily investigated topic [18, 19]. However, this intensive feature engineering process can also represent a significant bottleneck for machine learning integration, particularly when rigorous validation methodologies and strict evaluation metrics are recommended [20]. This limitation is even more pronounced when deep learning paradigms are introduced, as they achieve superior performance levels implicitly without requiring manual feature engineering [21, 22]—a capability largely attributed to the capacity of deep networks to achieve superhuman accuracy when supplied with extensive training datasets.

The efficacy of deep learning networks on mouse dynamics datasets depends heavily on how the analytical problem is formulated. For example, due to the inherent spatiotemporal attributes of tracking data, mouse dynamics can be treated as a time-series challenge while incorporating deep learning through diverse modalities. Multiple methodologies exist for integrating CNN architectures into mouse dynamics workflows [23]. One approach involves mapping raw X and Y coordinate trajectories to construct spatial user mouse maps, as seen in [12], and analyzing these spatial patterns via a two-dimensional CNN [24]. By isolating and processing the spatial attributes of mouse movement data, the image-recognition proficiencies of CNNs can be directly adapted for user verification tasks. Conversely, the mathematical advantages of convolutions within CNNs are not restricted to spatial structures; a temporal framework that treats data as a time-series signal and applies a 1D-CNN to convolve along the single time dimension of that signal has also been developed [25]. Moreover, [25] relies on X and Y velocities to generate tracking signals from mouse events rather than absolute coordinate positions, thereby yielding a translation-invariant classification model. Because of these characteristics, recurrent neural networks—particularly long short-term memory variations (LSTM-RNN)—have been widely endorsed due to their proven mastery of time-dependent sequential datasets [26, 27]. The divergent paths through which these models exploit the underlying spatiotemporal features of mouse dynamics data can heavily influence final accuracy, which motivated their inclusion in our comparative experiments.

It is vital to mention that two-class binary classification frameworks consistently exhibit superior performance compared to one-class variants across the broader spectrum of behavioral biometrics [28]. To guarantee a more comprehensive evaluation of this dataset, we also deploy a unified multi-class classification architecture designed to differentiate among all network users simultaneously. Traditional binary user authentication focuses exclusively on determining whether a specific mouse trajectory was generated by the authorized user or an imposter; however, this paradigm only scores a model's capacity to resolve inter-class variances (i.e., distinguishing the legitimate user from an outside attacker). The capability to further separate one legitimate user from another within a trusted corporate network (intra-class variances) offers immense practical security value and should not be overlooked. Consequently, this monograph introduces an artificial neural network tailored to operate under these multi-user constraints within mouse dynamics, raising the broader question of whether more sophisticated deep learning configurations can successfully scale with continued exploration.

## Materials and Methods

Data preprocessing workflows are specialized for each model category. Traditional machine learning frameworks depend entirely on manual feature engineering, whereas deep learning architectures are fed raw mouse dynamics sequences to perform implicit feature extraction and internal data analysis. Due to fundamental structural deviations between the networks, the varying spatial and temporal components of the data streams are harnessed differently and at asymmetrical rates based on the model type—as exemplified by the previously discussed differences between an LSTM-RNN and a 1D-CNN. Furthermore, across both machine learning and deep learning implementations within mouse dynamics, binary classification schemes stand as the most prevalent operational strategy. This methodology centers on constructing a dedicated, independent classifier for every distinct user. Each individual user's sub-dataset is curated to contain an identical balance of positive and negative class instances. This equilibrium is typically achieved by extracting all available movement samples from the target individual (representing the positive class) and appending an equivalent volume of unauthorized imposter samples (representing the negative class). These imposter points are derived via random selection from the remaining cohort of non-target users. In an experimental dataset encompassing  $k$  users, there will be  $n$  positive and  $n$  negative instances inside a given user's sub-dataset, which dictates that  $nk-1$  imposter samples are extracted from each non-target individual. This process generates a strictly balanced dataset that eliminates frequency-based algorithm bias during training. Thus, rather than being forced to resolve the identities of all users simultaneously, an individual classifier is only responsible for categorizing a mouse trajectory as either a genuine user action or an imposter movement. Our feature engineering pipeline generated 33 supplementary attributes from each participant's raw tracking data, capturing metrics such as velocity, acceleration, jerk, angular curvature, and vector trajectories. Comprehensive architectural details regarding our data preparation workflow for the machine learning algorithms are documented in our prior study [12]. While highly prevalent throughout mouse dynamics literature, binary classification models obscure more

granular analytical insights, such as revealing which users exhibit overlapping behavioral patterns. Accordingly, we also incorporate a multi-class classification methodology for tracking mouse dynamics, which introduces notable deviations from the conventional pipelines historically proposed. This section delineates our overarching methodology alongside the subtle yet vital distinctions between the models and their corresponding data preparation routines. All models were constructed, optimized, and validated in Python utilizing the Keras/TensorFlow ecosystem alongside the scikit-learn toolkit.

## Deep learning models

Every deep learning architecture and its constituent layers detailed throughout this manuscript were implemented using the native TensorFlow and Keras software libraries.

### LSTM-RNN

Recurrent neural networks have historically received widespread exposure within mouse dynamics research [23, 29] and, with the integration of specialized LSTM-RNN cells, continue to deliver state-of-the-art results across a multitude of technical domains. Broadly speaking, mouse dynamics can be mathematically structured as a continuous series of distinct behavioral actions occurring over an elastic timeframe, establishing LSTM-RNNs as an ideal model selection for this time-series data. As a specialized sub-branch of neural networks, LSTM-RNNs excel at accepting sequential input matrices, propagating this data through hierarchical layers to synthesize deeper levels of abstraction, and generating predictive outputs. Most notably, LSTM-RNN architectures benefit from internal memory cells that permit the identification of long-term temporal dependencies. This structural advantage has been most prominently realized in sub-fields like natural language processing due to the highly sequential and contextual dependencies inherent to linguistics; nevertheless, these structural advantages adapt effectively to generalized time-series signals. This configuration dictates that input matrices must be formatted as 3-dimensional arrays, where the core dimensions represent the training batch size, the number of input features, and the total count of chronological time steps. Consequently, the data preparation sequence for our LSTM-RNN involved assembling a singular mouse movement block utilizing 10 sequential mouse events, replicating the framework established in [12]. Due to the rigid length of 10 applied to every mouse action block—contrasting with the flexible window lengths seen in alternative literature—a time step parameter of 3 was specified for the LSTM-RNN. We hypothesized that using a time step of 3 would encode vital contextual characteristics into every data coordinate delivered to the LSTM-RNN while preserving a large enough data volume to train a model of our specific structural proportions.

The structural layout of this network consists of an initial input layer, three distinct hidden LSTM layers configured with 128 neurons each, followed sequentially by a dropout layer and a batch normalization layer, matching conventional LSTM-RNN engineering standards. These hidden layers are succeeded by a standalone fully connected layer of size 64 utilizing a ReLU (Rectified Linear Unit) activation function, which subsequently terminates into a final fully connected layer of size 1 that applies a softmax activation function to produce the final classification probabilities. The Adam optimization algorithm is deployed with an experimental learning rate of  $\alpha = 0.001$  combined with a mathematical decay rate of  $1.0 \times 10^{-6}$ . Finally, a binary cross-entropy loss function is applied to accommodate the dual-class orientation of the target data (determining whether the movement sequence matches the designated target user or an outside operator).

### 1D-CNN

Mouse dynamics data can be captured through various analytical modalities. Rather than merely examining long-term behavioral trends through a temporal lens, 1D-CNNs uniquely capitalize on both the spatial and temporal properties of the data. Convolutional networks broad-scale have demonstrated state-of-the-art performance across computer science domains, most prominently within computer vision. These architectures are founded on processing data configured in grid-like topologies using convolutional arithmetic, substituting standard matrix multiplication within network layers.

Regarding CNN applications specifically, diverse methodologies have been used to represent mouse dynamics data because a definitive, optimal representation remains unestablished due to the infancy of the field. Certain strategies advocate for utilizing velocity instead of absolute spatial coordinates to achieve a translation-invariant model [25], while others plot mouse paths onto a two-dimensional grid to employ a conventional 2D-CNN [24]. Several of these frameworks align with our dataset, as illustrated in **Figure 2**. Consequently, we integrate velocity values for mouse movement sequences rather than raw spatial coordinates to engineer a more resilient and versatile model. Because our mouse tracking data can be structured as a one-dimensional time-series signal, it is highly compatible with a 1D-CNN analysis. Mirroring our LSTM-RNN configuration, data matrices are fed into the network in a 3-dimensional format. However, instead of specifying a time step value of 3, we feed a single discrete mouse action directly into the model. Given that an action block already encompasses 10 distinct events, the network convolves through each sequential event to analyze the X and Y speed vectors at every UNIX timestamp, eliminating the necessity for batched time steps.

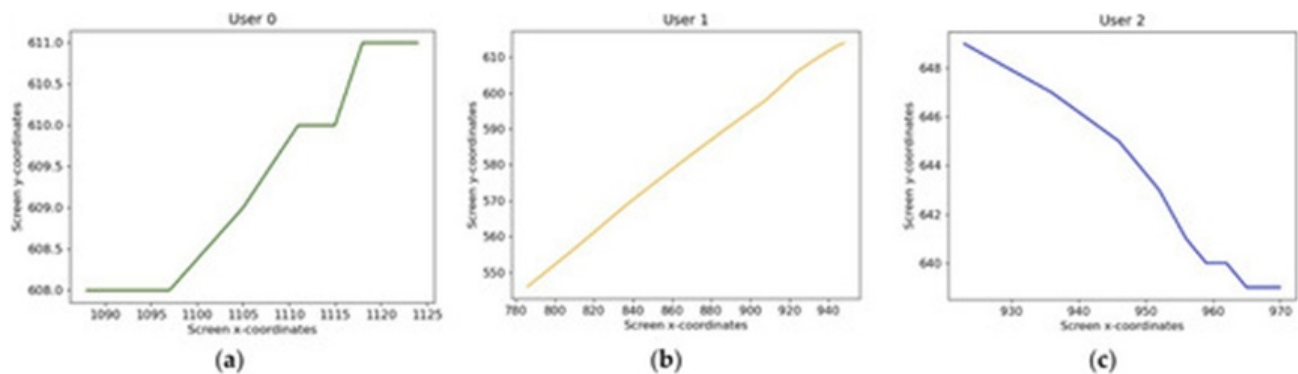


**Figure 2.** (a) Visual representation of one of user 0's mouse action with block length 10; (b) visual representation of one of user 1's mouse action with block length 10; and (c) visual representation of one of user 2's mouse action with block length 10.

The structural design of this network aligns closely with the framework in [25], though it features a narrower layer width. While model configurations of diverse capacities were evaluated, they produced negligible variances in overall validation accuracy. We implement an identical binary classification strategy to the one . Consequently, our network architecture comprises two 1D convolutional layers followed immediately by a global max pooling layer. A flatten layer is then applied to transform the max pooling output matrix into a vector, which feeds into two successive fully connected layers: an initial hidden layer with a size of 60 utilizing a ReLU activation function, and a terminating output layer with a size of 1 utilizing a softmax activation function for binary classification. The identical loss operations and hyperparameters are maintained here.

## ANN

Virtually all deep learning models developed for mouse dynamics have been constrained to binary classification schemes. Although this verification technique offers explicit advantages, a multi-class framework provides more granular behavioral insights. Continuous user verification via mouse dynamics maintains broad utility beyond simple intrusion detection; hence, specific operational contexts may arise where multi-class classification frameworks are far more appropriate. For instance, monitoring and separating verified users within a expansive network infrastructure is only achievable when models have been validated to perform effectively in a multi-class environment. Thus, it is essential to investigate the alternative operational environments where mouse dynamics can reliably function as a behavioral biometric for identity verification. Overcoming challenges that could push the boundaries of this biometric and the broader domain of continuous user authentication requires that mouse dynamics undergoes exhaustive and comprehensive exploration. To address this, we introduce an ANN designed to differentiate among all 40 users concurrently within a multi-class classification environment.

ANNs represent one of the foundational iterations of neural network technology, modeled loosely after contemporary hypotheses of biological neural processing. Raw input matrices propagate forward through the network architecture, undergoing a succession of affine non-linear transformations executed via multiplicative weights and additive biases. Because these ANNs are optimized in a supervised learning environment, the variance between network predictions and ground-truth target values is quantified by a loss function and minimized using variations of first-order iterative optimization algorithms, most frequently gradient descent. Furthermore, ANNs function as universal approximators, giving them the capacity to model infinitely complex mathematical functions. Consequently, they serve as an essential baseline tool when introducing deep learning methodologies to a novel field.

The network implements a straightforward architecture, consisting of two successive layers of size 256, two layers of size 128, two layers of size 64, and a final classification output layer of size 40. Every layer implements a ReLU activation function, with the exception of the terminating output layer, which applies a softmax activation function. Data preparation workflows vary only marginally from the steps used for the 1D-CNN and LSTM-RNN. A single mouse action block is ingested at a time, but raw absolute coordinates are substituted for velocity values since translation invariance does not apply to this specific multi-user task. Finally, a sparse categorical cross-entropy loss function is substituted for binary cross-entropy to accommodate the multi-class classification layout.

## Machine learning models

Every machine learning model evaluated in this study was constructed utilizing the Scikit-learn framework, specifically within the sklearn module ecosystem. The KNN and RF algorithms are drawn from the "neighbors" and "ensemble" sklearn modules, respectively.

### SVM

Support vector machines (SVM) represent a class of machine learning algorithms widely deployed for regression and classification tasks. Fundamentally, SVMs establish hyperplanes (alternatively called decision boundaries) that optimize the separation between distinct data classes. Given the multidimensionality inherent to complex machine learning problems, discovering an absolute spatial separation between classes can be challenging. Broadly, SVMs function by identifying the specific hyperplane that achieves the maximum margin of separation between classes within an  $n$ -dimensional space. In less complex environments, SVMs frequently utilize linear classification boundaries effectively to achieve a classification verdict; however, this is not always sufficient. Mirroring the theoretical underpinnings of neural networks, non-linear transformations regularly yield superior outcomes when processing complex, high-dimensional data profiles—a characteristic typical of mouse dynamics data and wider machine learning challenges. Classes often become more easily segmentable when mapped into higher-dimensional feature spaces; however, explicitly converting every sample vector into a higher dimension can be computationally inefficient and resource-intensive. Kernel functions are integrated by SVMs to circumvent this hurdle by serving as a similarity function over raw data structures, eliminating the need to explicitly calculate distances by projecting sample vectors into a target feature space. This mathematical approach allows SVMs to operate efficiently in high-dimensional spaces without experiencing a prohibitive increase in computational complexity, which is a vital attribute for an authentication framework utilizing mouse dynamics.

Additionally, SVMs typically function within a binary classification setup. While multi-class SVM frameworks have been introduced via "one-vs-rest" systems or aggregated "one-vs-one" voting schemas, this study deploys SVMs as straightforward binary classifiers to maintain methodological consistency with the other machine learning pipelines under evaluation. Although SVMs demonstrate strong capabilities in high-dimensional spaces and offer high memory efficiency, they remain highly vulnerable to data noise and present interpretability challenges. Because classification boundaries are determined strictly by a separating hyperplane, they lack probabilistic components that reveal the magnitude of error the

model experiences. Within the domain of mouse dynamics, this limitation could hinder the widespread commercial deployment of SVMs, as diagnosing model vulnerabilities and troubleshooting errors represents a more ambiguous process compared to alternative modeling options. Lastly, hyperparameter optimization is just as critical in machine learning workflows as it is in deep learning architectures; however, the reduced structural complexity of most traditional machine learning algorithms results in fewer hyperparameters requiring calibration. The kernel function of an SVM represents its most critical "hyperparameter"; therefore, we utilize a radial basis function kernel due to its versatile performance across diverse machine learning applications and its straightforward interpretability of distance metrics using an l2-norm.

## KNN

The k-nearest neighbor algorithm (KNN) is a nonparametric statistical learning method frequently utilized for classification and regression tasks. KNN models function under the core premise that similar data instances reside in close geometric proximity to one another. Operating on this assumption, a KNN model computes the distance from an unclassified data point to previously categorized groups. The algorithm assigns this unclassified point to a specific group provided there is an adequate number of neighbors,  $K$ , in close proximity, which necessitates the integration of a distance function. Mirroring SVM methodologies, l2-norms represent the most frequent choice for KNN algorithms, though they are not strictly mandatory.

In contrast to SVM architectures, KNN models have proven to be exceptionally robust against data noise and demonstrate superior efficiency on larger datasets. Because of the nonparametric, instance-based learning mechanisms underlying KNNs, they require no preliminary training phase on a dataset prior to generating classification outputs. Consequently, there is only a single hyperparameter requiring calibration:  $K$ , representing the quantity of neighboring points an instance must be near to be clustered into that group. Regrettably, optimizing this hyperparameter is notoriously difficult. Following the implementation of a cross-validated random hyperparameter search, we determined that configuring  $K$  to equal 13 produced the strongest generalized results across the majority of users. As observed with most traditional statistical learning workflows, KNN models face challenges in maintaining high accuracy rates when processing high-dimensional data structures. They also necessitate explicit feature scaling, such as data normalization, and exhibit high sensitivity to data outliers. The contrasting operational advantages and vulnerabilities of SVMs and KNNs allow us to evaluate these algorithms within a pseudo-adversarial framework, determining which underlying characteristics of our tracking data (such as dimensionality, dataset volume, or outlier frequency) exert a more substantial impact on final classification accuracy.

## Random forest

Random forests (RF) represent an ensemble learning approach composed of a diverse collection of decision trees that utilize a majority voting mechanism to generate a final classification or regression prediction. Their widespread adoption is driven by their inherent simplicity, structural flexibility, and high interpretability. At their fundamental core, RF architectures function as an extensive network of conditional if-else operations. A specific "test" is executed on the incoming data value at each node of an internal decision tree, with every radiating branch mapping the complete potential sample space for that input. As an input vector moves downward through the forest layout, the data is progressively segmented and further classified by the constituent decision trees. Because it is theoretically feasible to graphically map every individual decision tree within an RF framework, the underlying logic is remarkably transparent and more straightforward to decipher than nearly any alternative machine learning algorithm. Additionally, differences in feature magnitudes do not compromise the operational efficacy of RFs, which renders data scaling techniques unnecessary. While RF models are celebrated for being computationally efficient when processing test queries, their execution cost stands in sharp contrast to their required training resources. Training on an extensive dataset can necessitate an exceptionally deep RF configuration, as

the system must construct a vast array of decision trees to account for all possible feature scenarios. Consequently, the bulk of the computational burden is confined to the initial model training phase rather than its operational deployment. RFs are highly efficient and economical during active data classification; it is strictly the foundational framework synthesis that demands significant resources.

An ensemble training methodology involves navigating numerous operational trade-offs. The primary catalysts for prediction errors across most learning frameworks are noise, bias, and variance. Therefore, any algorithmic approach that systematically minimizes these specific errors has a strong likelihood of enhancing the precision and resilience of machine learning architectures. Ensemble systems are designed to aggregate multiple predictions from distinct models to mitigate the unique errors generated by any individual component. In the previously detailed algorithms, the structures operate independently and rely exclusively on the verdict of a singular model to categorize a data instance. Conversely, ensemble training permits multiple weak classifiers to be integrated into a unified collective, where the aggregated decision of these weak classifiers regularly yields a balanced, averaged outcome. This ensemble strategy operates under the assumption that each subset of weak classifiers will naturally produce errors. To account for this, the weak classifiers are assembled in a manner that minimizes the correlation among different groups. Consequently, the vulnerability of a specific group to a particular type of error may not manifest within the remaining groups. An illustration of this mechanism within the specific domain of mouse dynamics has been documented in [30]. Consolidating the outputs from all internal classifiers counteracts individual model errors, establishing a singular, powerful classification system that is highly resistant to overfitting. Within the context of mouse dynamics research, RF models function as a dependable middle ground between the highly divergent approaches of KNNs and SVMs. Analyzing the variations in evaluation metrics across these three distinct paradigms can offer deeper insights into which frameworks achieve superior performance for user classification based on mouse movements. Following an exhaustive grid search to pinpoint optimal hyperparameter configurations, we determined that the following parameters for a Keras random forest yielded the most optimal generalized outcomes:  $n\_estimators = 1600$ ,  $min\_samples\_split = 2$ ,  $min\_samples\_leaf = 1$ ,  $max\_features = "sqrt"$ ,  $max\_depth = 30$ , and  $bootstrap = "False"$ . For complete clarity, the entire suite of hyperparameters for each model (along with optimizer hyperparameters for the deep learning architectures) is cataloged in [Table 1](#).

**Table 1.** List of hyperparameters for each model.

| Model                   | Hyperparameter                    | Value    |
|-------------------------|-----------------------------------|----------|
| 1D-CNN / LSTM-RNN / ANN | $\alpha$ (learning rate)          | 0.001    |
| 1D-CNN / LSTM-RNN / ANN | $\epsilon$ (decay rate)           | 0.000001 |
| KNN                     | Number of neighbors (k)           | 13       |
| SVM                     | C (regularization parameter)      | 1        |
| SVM                     | Kernel type                       | "rbf"    |
| RF                      | Number of estimators              | 1600     |
| RF                      | Minimum samples required to split | 2        |
| RF                      | Minimum samples per leaf          | 1        |
| RF                      | Maximum features considered       | "sqrt"   |
| RF                      | Maximum tree depth                | 30       |
| RF                      | Bootstrap usage                   | "False"  |

# Results and Discussion

To maintain direct comparability with the outcomes presented in [12], we assessed each binary classification framework utilizing three distinct performance dimensions: accuracy (ACC), false positive rate (FPR), and false negative rate (FNR). Within theoretical evaluation contexts, FPR and FNR can be viewed as equally problematic errors; however, FPR metrics are significantly more hazardous within a user verification architecture. Specifically, a failure to authenticate an authorized user presents a minor operational security risk compared to the critical vulnerability of mistakenly authenticating an unauthorized intruder. For the sake of conciseness, we report only the FPR when assessing the deep learning architectures. While the equal error rate (EER) identifies the precise balance point where the FPR matches the FNR and is utilized in [12, 31] as a generalized benchmark to compare diverse classifiers, we have opted to employ the F1-score—the harmonic mean of precision and recall—throughout this investigation because it provides a more descriptive representation of a model's precise sensitivity and retrieval capacities [32]. Additionally, documenting individualized performance metrics for every single user within this manuscript is logistically unfeasible due to space constraints; therefore, we limit our presentation to the individual records and macro-averages of the top-10 users ranked by accuracy (applicable exclusively to the binary classification systems). The ANN outcomes are derived using the complete, unsegmented dataset without partitioning users into distinct binary classes, thereby offering a more authentic representation of how a baseline framework functions across the entire user population in this dataset. All performance tracking and visualization steps were executed utilizing the "metrics" module within the sklearn library. The compiled outcomes from each model, organized hierarchically by classification accuracy, are presented below in Tables 2-7, with the optimal and poorest values highlighted in green and red, respectively:

Table 2. 1D-CNN results.

| User               | F1 Score | FPR     | ACC      |
|--------------------|----------|---------|----------|
| 13                 | 0.9197   | 0.1563  | 0.8637   |
| 24                 | 0.9202   | 0.1602  | 0.8634   |
| 2                  | 0.8551   | 0.1839  | 0.8587   |
| 39                 | 0.9181   | 0.1588  | 0.8581   |
| 15                 | 0.9225   | 0.1607  | 0.8577   |
| 30                 | 0.9191   | 0.1194  | 0.8575   |
| 19                 | 0.9171   | 0.1513  | 0.8567   |
| 14                 | 0.8934   | 0.1652  | 0.8525   |
| 0                  | 0.9182   | 0.1488  | 0.8524   |
| 9                  | 0.9151   | 0.1419  | 0.8521   |
| Average            | 0.9099   | 0.1546  | 0.8573   |
| Standard Deviation | 0.02091  | 0.01668 | 0.004156 |

Table 3. LSTM-RNN results.

| User | F1 Score | FPR | ACC |
|------|----------|-----|-----|
|      |          |     |     |

|                    |         |              |          |
|--------------------|---------|--------------|----------|
| 19                 | 0.8948  | 0.1553<br>15 | 0.8613   |
| 7                  | 0.9052  | 0.1564       | 0.8589   |
| 24                 | 0.8948  | 0.1457       | 0.8562   |
| 6                  | 0.8920  | 0.1569       | 0.8561   |
| 38                 | 0.9110  | 0.1515       | 0.8560   |
| 30                 | 0.9142  | 0.1659       | 0.8554   |
| 15                 | 0.9174  | 0.1511       | 0.8548   |
| 37                 | 0.9084  | 0.1482       | 0.8525   |
| 33                 | 0.9136  | 0.1737       | 0.8525   |
| 12                 | 0.8836  | 0.1685       | 0.8526   |
| Average            | 0.9035  | 0.1561       | 0.8556   |
| Standard Deviation | 0.01141 | 0.008327     | 0.002849 |

**Table 4.** SVM results.

| User               | F1 Score | FNR    | FPR     | ACC     |
|--------------------|----------|--------|---------|---------|
| 4                  | 0.6541   | 0.6024 | 0.1692  | 0.6518  |
| 39                 | 0.6426   | 0.4408 | 0.3996  | 0.6417  |
| 10                 | 0.6256   | 0.4718 | 0.3964  | 0.6292  |
| 0                  | 0.6206   | 0.4718 | 0.4325  | 0.6174  |
| 32                 | 0.5930   | 0.3013 | 0.4362  | 0.6066  |
| 5                  | 0.6031   | 0.6511 | 0.2954  | 0.6037  |
| 1                  | 0.5895   | 0.6158 | 0.2875  | 0.5890  |
| 3                  | 0.5832   | 0.4238 | 0.4112  | 0.5841  |
| 15                 | 0.5873   | 0.7013 | 0.2361  | 0.5741  |
| 31                 | 0.5662   | 0.6191 | 0.2818  | 0.5730  |
| Average            | 0.6065   | 0.5299 | 0.3345  | 0.6071  |
| Standard Deviation | 0.02819  | 0.1259 | 0.09274 | 0.02771 |

**Table 5.** KNN results.

| User | F1 Score | FNR | FPR | ACC |
|------|----------|-----|-----|-----|
|------|----------|-----|-----|-----|

|                    |         |         |         |         |
|--------------------|---------|---------|---------|---------|
| 8                  | 0.6670  | 0.4030  | 0.4994  | 0.6584  |
| 35                 | 0.6515  | 0.4306  | 0.4260  | 0.6536  |
| 12                 | 0.6192  | 0.3311  | 0.3697  | 0.6195  |
| 6                  | 0.6281  | 0.3118  | 0.4754  | 0.6190  |
| 34                 | 0.6026  | 0.4744  | 0.4028  | 0.6066  |
| 32                 | 0.6054  | 0.2484  | 0.3650  | 0.6054  |
| 30                 | 0.6064  | 0.4165  | 0.4610  | 0.6033  |
| 11                 | 0.6114  | 0.4045  | 0.4741  | 0.6021  |
| 1                  | 0.5936  | 0.3558  | 0.4745  | 0.5938  |
| 39                 | 0.5921  | 0.4102  | 0.4327  | 0.5932  |
| Average            | 0.6177  | 0.3786  | 0.4380  | 0.6155  |
| Standard Deviation | 0.02465 | 0.06647 | 0.04683 | 0.02309 |

**Table 6.** RF results.

| User               | F1 Score | FNR     | FPR     | ACC     |
|--------------------|----------|---------|---------|---------|
| 8                  | 0.7016   | 0.3764  | 0.2582  | 0.6827  |
| 12                 | 0.6739   | 0.3442  | 0.3076  | 0.6740  |
| 35                 | 0.6564   | 0.4108  | 0.2732  | 0.6580  |
| 11                 | 0.6673   | 0.3158  | 0.3692  | 0.6575  |
| 6                  | 0.6745   | 0.3734  | 0.3171  | 0.6548  |
| 34                 | 0.6501   | 0.4165  | 0.2820  | 0.6507  |
| 32                 | 0.6401   | 0.3259  | 0.3923  | 0.6408  |
| 30                 | 0.6575   | 0.3569  | 0.3761  | 0.6355  |
| 1                  | 0.6247   | 0.4937  | 0.2454  | 0.6305  |
| 18                 | 0.6210   | 0.3553  | 0.4022  | 0.6212  |
| Average            | 0.6567   | 0.3768  | 0.3223  | 0.6506  |
| Standard Deviation | 0.02440  | 0.05232 | 0.05838 | 0.01912 |

**Table 7.** ANN results.

| Peak Training Accuracy | Peak Testing Accuracy |
|------------------------|-----------------------|
|                        |                       |

|        |        |
|--------|--------|
| 0.9589 | 0.9248 |
|--------|--------|

## Discussion and analysis

In comparison to [12], the integration of an additional 30 users within this dataset did not appear to exert a severe negative impact on the performance of the random forest models. On the contrary, the average accuracy improved with this expanded dataset, as evidenced by Table 5, where only 3 of the 10 users from the foundational dataset ranked among the top performers. Furthermore, the random forests demonstrated considerably lower FNR values than those documented in [12]. The data profiles of certain users seem to be more easily deciphered by one specific model architecture over another. For instance, user 8 achieved top-tier performance within both the RF and KNN frameworks but failed to place within the top-10 across the remaining architectures. The obtained F1-scores also reveal that the FPR and FNR values at the 0.50 decision threshold are lower on a per-user basis for this dataset relative to the original study. The standard deviation for classification accuracy within the random forest models was 0.01912, marking the lowest variance among the machine learning alternatives and falling below the variance noted in [12]. Both the 1D-CNN and LSTM-RNN displayed marginal standard deviations among their top-10 users, indicating highly consistent identity verifications when the networks operate at peak performance, though these findings cannot be automatically extrapolated across the entire user base. In agreement with existing literature [33, 34], random forests emerge as one of the generally superior machine learning algorithms for user verification via mouse dynamics, a conclusion supported by the results of our machine learning implementations.

The SVM and KNN configurations yielded lower overall performance benchmarks on this dataset than the random forest model. This discrepancy is most apparent when analyzing the ROC curves displayed in Figure 3, where the KNN framework demonstrated weak capabilities. This outcome may indicate that the inherent vulnerability of KNN algorithms when handling high-dimensional data profiles could be highly challenging to mitigate, a factor requiring deeper investigation in subsequent studies. Compared against the SVM outcomes reported in [30] and [35], our deployed SVM outperformed both prior frameworks by achieving lower F1-scores on our dataset relative to both the Balabit and TWOS benchmarks. We previously hypothesized in [12] that gathering tracking data within a faster-paced operational environment could embed richer informational content within the sequences compared to data captured during monotonous, administrative operations. The investigators in [25] evaluated a deep 1D-CNN architecture on the Balabit dataset while utilizing longer mouse action blocks alongside transfer learning protocols. Rather than utilizing randomly initialized weight matrices at the start of deep learning optimization, transfer learning permits the initialization of weights from a pre-trained network. This strategy not only accelerates training times but can also facilitate more optimal loss convergence, thereby boosting overall efficacy. Despite the integration of these sophisticated learning techniques in [25], our baseline SVM and KNN models both surpassed that framework in overall accuracy, lending support to the premise that our dataset may possess higher baseline quality and capture real-world behavioral entropy in mouse usage more effectively.

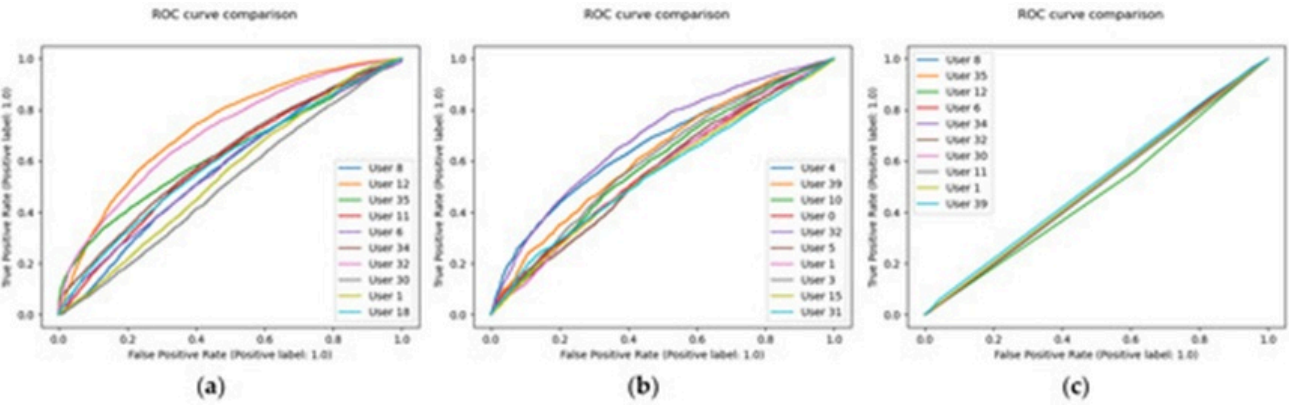


**Figure 3.** (a) RF ROC curve; (b) SVM ROC curve; and (c) KNN ROC curve.

Regarding deep learning implementations, every deep network evaluated in this study outperformed the earlier random forest framework presented in [12]. Deep learning integration was highlighted within the limitations of [12] and represented a vital technological milestone we aimed to address in this research. Based on the average accuracy rates of approximately 80% achieved by our 1D-CNN and LSTM-RNN models, it can be concluded that deep learning frameworks perform at a superior level on this dataset when contrasted with simpler machine learning alternatives. Furthermore, our proposed ANN achieved an exceptional test accuracy of 92.48%, the highest performance benchmark observed for this dataset to date. This outcome is particularly remarkable given that the ANN functions within the inherently more complex multi-class classification paradigm. The authors of [36] introduced an ANN for multi-modal user authentication that integrated both mouse dynamics and eye-tracking metrics, yet their framework underperformed relative to our standalone ANN. Moreover, their evaluation dataset was limited to 32 users. Although the researchers in [36] utilized diverse evaluation scenarios to acquire a broader perspective of model capabilities, every tested configuration involved fewer distinct classes than the 40 users analyzed in this study. More critically, their workflow required manual feature engineering from raw data streams before network input, whereas our ANN operates completely without manual feature extraction. Our ANN consistently surpassed theirs regarding F1-scores across virtually every evaluation scenario. Beyond F1-score comparisons, our ANN also exceeded the raw accuracy rates achieved by the models introduced in [37, 38]. Notably, [38] leverages substantially larger mouse action block dimensions, spanning from 500 to 2500 events. While our dataset is not extensive enough to support such block sizes, the highly comparable outcomes achieved with our compact block size of 10 indicate that our models can function at equivalent performance tiers while requiring less descriptive and voluminous data. A consolidated summary of our findings relative to historical literature is presented in Table 8.

**Table 8.** Comparison of previous literature and our novel contributions.

| Reference | Strengths                                                                                                              | Weaknesses                                                                                                                                                                                                     | Our Contribution                                                                                                                                                                                                    |
|-----------|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [12]      | <ul style="list-style-type: none"><li>Introduces a novel use of Random Forest (RF) on our dataset</li></ul>            | <ul style="list-style-type: none"><li>Does not include additional baseline models for broader comparison</li></ul>                                                                                             | <ul style="list-style-type: none"><li>Incorporation of five extra models applied to our dataset</li><li>Expansion of dataset with 30 additional users</li></ul>                                                     |
| [25]      | <ul style="list-style-type: none"><li>Among the few studies investigating 1D-CNN for mouse dynamics analysis</li></ul> | <ul style="list-style-type: none"><li>Reliance on the Balabit dataset restricts findings to low-intensity scenarios</li><li>Large mouse-action segments require more data and increase training time</li></ul> | <ul style="list-style-type: none"><li>Evaluation of 1D-CNN using our high-intensity mouse dynamics dataset</li><li>Use of shorter mouse-action segments reduces complexity and increases data granularity</li></ul> |
| [29]      | <ul style="list-style-type: none"><li>Investigates multiple CNN and RNN architectures for</li></ul>                    | <ul style="list-style-type: none"><li>Use of Balabit and TWOS datasets again limits evaluation</li></ul>                                                                                                       | <ul style="list-style-type: none"><li>Demonstration of RF and SVM effectiveness on shorter mouse-action</li></ul>                                                                                                   |

|      |                                                                                                                                                           |                                                                                                                                                                 |                                                                                                                                                                                                                                               |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      | mouse dynamics applications                                                                                                                               | to low-intensity settings• SVM shows weak performance with small mouse-action segments                                                                          | segments in a high-intensity setting• Introduction of LSTM-RNN instead of GRU-RNN for mouse dynamics analysis                                                                                                                                 |
| [36] | • Proposes multimodal authentication using both mouse dynamics and eye-tracking data• Performs extensive hyperparameter tuning                            | • Very poor ANN performance when applied to mouse-based authentication across 32 users• Eye-tracking data introduces more invasive data collection requirements | • Our ANN significantly outperforms [37] across all evaluation metrics and aspects                                                                                                                                                            |
| [37] | • Strong performance reported for SVM and ANN models• Uses unconventional feature engineering that yields promising and potentially generalizable results | • ANN requires manual feature engineering• Restricts mouse movement to a constrained GUI environment                                                            | • Improved ANN performance using a larger, free-form interaction environment that better reflects real-world behavior• ANN operates without requiring manual feature extraction                                                               |
| [38] | • Introduces novel noise-reduction techniques for mouse dynamics data• Applies Gaussian Naïve Bayes classifier to the dataset introduced in [32]          | • Limited methodological detail may reduce reproducibility• Evaluates only a small number of models, limiting baseline comparisons                              | • Provides a more detailed experimental description to support reproducibility and future research• Demonstrates improved performance using smaller action blocks• Expands model diversity to better assess dataset strengths and limitations |

The addition of new users to the dataset relative to [12] appears to have enhanced overall model performance, suggesting that expanding sample sizes in future studies could yield even higher accuracy bounds. To varying degrees, all of our developed models outperformed frameworks documented in earlier literature, with the most conspicuous improvements over our previous study being the deployment of deep learning architectures and the substantial reduction of FNR metrics within the random forest models. While these results do not guarantee that our models would sustain identical performance levels if deployed within live, real-world operational environments, they do demonstrate that our novel mouse dynamics dataset possesses high baseline quality and offers strong potential for future research.

## Limitations

Identifying and addressing limitations is essential, particularly within a specialized discipline like mouse dynamics, to foster ongoing research and optimization. The most conspicuous area requiring improvement is the performance profile of the SVM and KNN models on this dataset. Because a moderate spectrum of hyperparameter configurations was evaluated for both systems, aggressive improvements in classification accuracy via standard hyperparameter tuning may be bounded for these specific SVM and KNN setups. Deploying alternative data preprocessing pipelines or simply gathering a larger volume of data points could facilitate superior accuracy rates. For instance, many classification tasks can be reformulated using regression methodologies, an alternative approach that applies equally to mouse dynamics research [39]. Furthermore, conducting a more detailed analysis of the fluctuating user frequencies within the top-10 accuracy brackets for each framework might uncover hidden structural traits within the data that could be leveraged to refine future models.

The deployed deep learning architectures were intentionally kept straightforward to minimize computational training times and prevent unnecessary hyperparameter complexities. Expanding the capacity and depth of these deep learning models could yield higher accuracy and improved F1-scores, though likely at the expense of computational training duration. Lastly, the relative youth of mouse dynamics as a biometric discipline leaves significant opportunities for major methodological breakthroughs. While ANNs have demonstrated strong performance both in established literature and within this study on mouse dynamics data, they represent a relatively generalized and foundational implementation of deep learning. When engineered specifically for this distinct problem domain, more complex, specialized deep learning architectures that excel at processing time-series data may be uncovered that perform at a superior tier compared to generally applicable models.

## Conclusion

This study expands upon prior research regarding a newly proposed dataset for continuous user verification utilizing mouse dynamics. The preceding publication evaluated a single variant of a machine learning binary classifier on a limited sample of 10 users. This manuscript introduces an expanded dataset comprising 40 users and establishes baseline performance metrics across three deep learning and three traditional machine learning models. Advancing the single RF configuration proposed in [12], we also investigate the efficacy of SVM and KNN frameworks for user authentication tasks within mouse dynamics. The random forest models developed in this study demonstrated generally superior performance compared to the benchmarks in [12], yielding higher average accuracies alongside more balanced distributions of FPR and FNR metrics. While the KNN models appeared to face challenges on our dataset, our SVM implementations achieved a more equitable balance between FPR and FNR than the original RF model in [12].

Furthermore, we introduce a novel multi-class classification evaluation environment. As anticipated, the deep learning architectures outperformed the traditional machine learning models by an average margin of approximately 20% in accuracy, with the ANN model achieving an impressive test accuracy exceeding 92%. This finding is exceptionally promising because the ANN multi-class paradigm completely removes the necessity for complex binary data preprocessing and independent classifier management. The 1D-CNN and LSTM-RNN models were integrated specifically to analyze and contrast alternative structural approaches to sequential time-series data. They yielded highly comparable accuracies and F1-scores, with the 1D-CNN maintaining a marginal advantage over the LSTM-RNN regarding F1 metrics. A wide array of new investigative paths can be pursued utilizing this dataset and the broader discipline of mouse dynamics; continuous research is critical given the promising outcomes detailed in this study.

For instance, historical mouse dynamics frameworks have been designed primarily around low-intensity computer interactions, such as online banking portals or routine administrative workloads, as reflected in the limited selection of publicly accessible datasets. The introduction of high-intensity mouse dynamics assessment creates opportunities for far more resilient user authentication frameworks. With the massive growth of professional esports, continuous user tracking to eliminate cheating and unfair competitive advantages has become critically important. Video games that necessitate multiple rapid, high-precision mouse actions, typical of first-person shooter titles, frequently face the unauthorized deployment of automated aiming programs. Models optimized on low-intensity mouse tracking tasks may lack the sensitivity required to detect and flag such erratic, fast-paced behavioral traits, emphasizing the need for wider-reaching mouse dynamics studies. Additionally the integration of deeper architectures, alternative data preparation pipelines, transfer learning techniques [40], and alternative neural topologies could all be investigated to unlock higher performance thresholds.

To achieve widespread commercial adoption, continuous user authentication systems powered by mouse dynamics must prove to be highly resilient, versatile, and precise. The non-intrusive operational nature of mouse dynamics, paired with the high performance rates achieved by our models across a larger, more diverse user base, highlights the immense capability of this behavioral biometric. Both the verification and classification paradigms evaluated in this study are

demonstrated to be highly viable frameworks for this dataset and for mouse dynamics generally, establishing a strong foundation for future exploration.

## Acknowledgements

Special thanks to the High-Performance Computing Cluster at UWEC, specifically for the generous access to the BOSE supercomputing cluster for training the models in this article.

## Conflict of interest

None

## Financial support

None

## Ethics statement

The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board of The University of Wisconsin—Eau Claire, proposal number 202121068 for studies involving humans approved on 22 September 2021.

Informed consent was obtained from all subjects involved in the study. Written consent was waived due to the fact that data was collected anonymously, and any signed document was unnecessary and could be potentially compromising.

Received: 13 Apr 2025

Revised: 14 May 2025

Accepted: 03 Jul 2025

Published online: 15 September 2025

## Rights and permissions

**Open Access** The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

## References

- Siddiqui N, Pryor L, Dave R. User authentication schemes using machine learning methods—A review. In: Proceedings of International Conference on Communication and Computational Technologies. Singapore: Springer; 2021. p.703-23.  
[https://doi.org/10.1007/978-981-16-3246-4\\_54](https://doi.org/10.1007/978-981-16-3246-4_54).
- Prakash A, Krishnaveni R, Dhanalakshmi R. Continuous user authentication using multimodal biometric traits with optimal feature level fusion. *Int J Biomed Eng Technol*. 2020;34(1):1-19.  
<https://doi.org/10.1504/IJBET.2020.10032546>.
- Wildes RP. Iris recognition: An emerging biometric technology. *Proc IEEE*. 1997;85(9):1348-63.

Zhang L, Tan S, Yang J, Chen Y. VoiceLive: A phoneme localization based liveness detection for voice authentication on smartphones. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016 Oct 24-28; Vienna, Austria. New York: ACM; 2016. p.1080-91.  
<https://doi.org/10.1145/2976749.2978296>.

Teh PS, Zhang N, Teoh ABJ, Chen K. A survey on touch dynamics authentication in mobile devices. *Comput Secur*. 2016;59:210-35.  
<https://doi.org/10.1016/j.cose.2016.03.003>.

Ayotte B, Banavar MK, Hou D, Schuckers S. Group leakage overestimates performance: A case study in keystroke dynamics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops; 2021. p.1410-7.  
<https://doi.org/10.1109/CVPRW53098.2021.00156>.

Bhana B, Flowerday S. Passphrase and keystroke dynamics authentication: Usable security. *Comput Secur*. 2020;96:101925.  
<https://doi.org/10.1016/j.cose.2020.101925>.

Shen C, Cai Z, Guan X. Continuous authentication for mouse dynamics: A pattern-growth approach. In: Proceedings of the 2012 IEEE/IFIP International Conference on Dependable Systems and Networks; 2012. p.1-12.  
<https://doi.org/10.1109/DSN.2012.6263955>.

Pusara M, Brodley CE. User re-authentication via mouse movements. In: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security; 2004. p.1-8.  
<https://doi.org/10.1145/1029208.1029210>.

Antal M, Fejér N, Buza K. SapiMouse: Mouse dynamics-based user authentication using deep feature learning. In: Proceedings of the 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI); 2021. p.61-6.  
<https://doi.org/10.1109/SACI51354.2021.9465583>.

Jaiswal AK, Tiwari P, Hossain MS. Predicting users' behavior using mouse movement information: An information foraging theory perspective. *Neural Comput Appl*. 2020;32.  
<https://doi.org/10.1007/s00521-020-05306-7>.

Siddiqui N, Dave R, Seliya N. Continuous user authentication using mouse dynamics, machine learning, and Minecraft. In: Proceedings of the 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET); 2021. p.1-6.

Fülöp Á, Kovács L, Kurics T, Windhager-Poko E. Balabit Mouse Dynamics Challenge Data Set [Internet]. 2016 [cited 2022 Mar 3]. Available from: <https://github.com/balabit/Mouse-Dynamics-Challenge>

Harilal A, Toffalini F, Castellanos J, Guarnizo J, Homoliak I, Ochoa M. TWOS: A dataset of malicious insider threat behavior based on a gamified competition. In: Proceedings of the 2017 International Workshop on Managing Insider Security Threats; 2017. p.45-56.

Gamboa H, Fred AL. An identity authentication system based on human computer interaction behaviour. In: Proceedings of the 3rd International Workshop on Pattern Recognition in Information Systems (PRIS 2003); 2003. p.46-55.

Shen C, Cai Z, Guan X, Sha H, Du J. Feature analysis of mouse dynamics in identity authentication and monitoring. In: Proceedings of the 2009 IEEE International Conference on Communications; 2009. p.1-5.  
<https://doi.org/10.1109/ICC.2009.5199032>.

Antal M, Egyed-Zsigmond E. Intrusion detection using mouse dynamics. *IET Biom*. 2019;8:285-94.  
<https://doi.org/10.1049/iet-bmt.2018.5126>.

Shen C, Cai Z, Guan X, Du Y, Maxion RA. User authentication through mouse dynamics. *IEEE Trans Inf Forensics Secur.* 2013;8(1):16-30.

<https://doi.org/10.1109/TIFS.2012.2223677>.

Kaixin W, Hongri L, Bailing W, Shujie H, Jia S. A user authentication and identification model based on mouse dynamics. In: *Proceedings of the 6th International Conference on Information Engineering*; 2017. p.1-6.

Zheng N, Paloski A, Wang H. An efficient user verification system via mouse movements. In: *Proceedings of the 18th ACM Conference on Computer and Communications Security*; 2011. p.139-50.

<https://doi.org/10.1145/2046707.2046725>.

Chong P, Elovici Y, Binder A. User authentication based on mouse dynamics using deep neural networks: A comprehensive study. *IEEE Trans Inf Forensics Secur.* 2020;15:1086-101.

Hu T, Niu W, Zhang X, Liu X, Lu J, Liu Y. An insider threat detection approach based on mouse dynamics and deep learning. *Secur Commun Netw.* 2019;2019:3898951.

Fu S, Qin D, Qiao D, Amariuca GT. RUMBA-Mouse: Rapid user mouse-behavior authentication using a CNN-RNN approach. In: *Proceedings of the 2020 IEEE Conference on Communications and Network Security (CNS)*; 2020. p.1-9.

<https://doi.org/10.1109/CNS48642.2020.9162287>.

Chong P, Tan YXM, Guarnizo J, Elovici Y, Binder A. Mouse authentication without the temporal aspect—What does a 2D-CNN learn? In: *Proceedings of the 2018 IEEE Symposium on Security and Privacy Workshops (SPW)*; 2018. p.15-21.

<https://doi.org/10.1109/SPW.2018.00011>.

Antal M, Fejér N. Mouse dynamics based user recognition using deep learning. *Acta Univ Sapientiae Inform.* 2020;12(1):39-50.

<https://doi.org/10.2478/ausi-2020-0003>.

Siarni-Namini S, Tavakoli N, Namin AS. The performance of LSTM and BiLSTM in forecasting time series. In: *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*; 2019. p.3285-92.

Karim F, Majumdar S, Darabi H, Chen S. LSTM fully convolutional networks for time series classification. *IEEE Access.* 2018;6:1662-69.

<https://doi.org/10.1109/ACCESS.2017.2779939>.

Antal M, Szabó LZ. An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices. In: *Proceedings of the 2015 20th International Conference on Control Systems and Computer Science*; 2015. p.343-50.

<https://doi.org/10.1109/CSCS.2015.16>.

Tan YXM, Iacovazzi A, Homoliak I, Elovici Y, Binder A. Adversarial attacks on remote user authentication using behavioural mouse dynamics. In: *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*; 2019. p.1-10.

Gao L, Lian Y, Yang H, Xin R, Yu Z, Chen W, et al. Continuous authentication of mouse dynamics based on decision level fusion. In: *Proceedings of the 2020 International Wireless Communications and Mobile Computing Conference (IWCMC)*; 2020. p.210-4.

Ahmed AAE, Traore I. A new biometric technology based on mouse dynamics. *IEEE Trans Dependable Secur Comput.* 2007;4(3):165-79.

<https://doi.org/10.1109/TDSC.2007.70207>.

Pepa L, Sabatelli A, Ciabattini L, Monteriù A, Lamberti F, Morra L. Stress detection in computer users from keyboard and mouse dynamics. *IEEE Trans Consum Electron.* 2021;67(1):12-9.

<https://doi.org/10.1109/TCE.2020.3045228>.

Monaro M, Gamberini L, Sartori G. The detection of faked identity using unexpected questions and mouse dynamics. *PLoS One*. 2017;12(5):e0177851.

<https://doi.org/10.1371/journal.pone.0177851>.

Almalki S, Chatterjee P, Roy K. Continuous authentication using mouse clickstream data analysis. In: *Security, Privacy, and Anonymity in Computation, Communication, and Storage*. Cham: Springer; 2019. p.76-85.

[https://doi.org/10.1007/978-3-030-24900-7\\_6](https://doi.org/10.1007/978-3-030-24900-7_6).

Bours P, Fullu CJ. A login system using mouse dynamics. In: *Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*; 2009. p.1072-7.

Rose J, Liu Y, Awad A. Biometric authentication using mouse and eye movement data. In: *Proceedings of the 2017 IEEE Security and Privacy Workshops (SPW)*; 2017. p.47-55.

Kulkarni S, Fadewar H. Accuracy comparison of classification techniques for mouse dynamics-based biometric CaRP. In: *Proceedings of the Third International Conference on Computational Intelligence and Informatics*. Singapore: Springer; 2020. p.701-11.

Salman OA, Hameed SM. Using mouse dynamics for continuous user authentication. In: *Proceedings of the Future Technologies Conference*. Cham: Springer; 2018. p.776-87.

Lu H, Rose J, Liu Y, Awad A, Hou L. Combining mouse and eye movement biometrics for user authentication. In: *Information Security Practices*. Cham: Springer; 2017. p.55-71.

[https://doi.org/10.1007/978-3-319-48947-6\\_5](https://doi.org/10.1007/978-3-319-48947-6_5).

Chowdhury D, Das A, Dey A, Sarkar S, Dwivedi AD, Rao Mukkamala R, et al. ABCanDroid: A cloud integrated Android app for noninvasive early breast cancer detection using transfer learning. *Sensors (Basel)*. 2022;22:832.