Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242

DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Research Article

User Authentication Based on Mouse Dynamics Using an Efficient-Net Model

Semaa Hatem Youssef[1]
[1]Informatics Institute for Postgraduate Studies.
University of Information Technology and Communications
Baghdad, Iraq
ms202310752@iips.edu.iq

Mohammed Salih Mahdi[1,2]
[1]Informatics Institute for Postgraduate Studies
University of Information Technology and Communications
Baghdad, Iraq
[2]Business Informatics College, University of Information Technology and Communications,
Baghdad, Iraq
mohammed.salih@uoitc.edu.iq

A R T I C L E I N F O
Article History
Received: 21/06/2025
Accepted: 09/08/2025
Published: 10/10/2025

ABSTRACT

As a digital threats become increasingly sophisticated, user authentication has become vitally important in cybersecurity. Traditional authentication methods such as passwords are under increasing assault from a range of attacks, behavioral biometrics, as a mouse dynamics have the potential to address these attacks in a way that is largely passive and continuous. In this article we provided a new solution that rests on mouse dynamics behavior together with a lightweight deep learning model inspired by EfficientNet, specifically designed for Behavioral Assessment of Numerical Data (BAND). The SapiMouse dataset, consisting of mouse tracking data from 120 actual users, is harnessed, and applying preprocessing techniques such as Quantile Transformation and Min-Max Encoding, along with encoding the raw data were prepared for model training, the modified EfficientNet model retains its computational efficiency while also being tailored to work with numerical input, its structure uses compact convolutions along with compound scaling to capture time-series mouse data discriminative features, lowering the processing burden while maintaining accuracy. To stabilize training and enhance generalization, dropout and batch normalization layers were added, ensuring robustness to overfitting, even when using data generated by a model. CGAN's capacity for class sample synthesis was harnessed towards improving recognition of unused user profiles, resulting in a total of 240 unique classes (120 real + 120 synthetic). The model reached an accuracy of 99.24% for classification and a macro-averaged F1-score of 0.991 on the testing set. An inference time of only 0.2331 seconds per sample, alongside a cumulative training duration of 158.25 seconds, suggests real-time applicability, so these findings support the promise of repurposing advanced deep learning models for behavioral biometrics, providing affordable, scalable and efficient user verification for sensitive security contexts.

Keywords: Mouse dynamics; Behavioral biometrics; Efficient Net; User authentication; Deep learning; Transfer learning; Cybersecurity.

1. INTRODUCTION

Behavioral biometrics are the measurable variations of an individual's behaviors when interacting with a digital system, mouse dynamics includes measurable behavioral characteristics such as how fast the user moves the mouse. The rhythm in click locations, and angles of movement. Behavioral authentication techniques are more difficult to imitate than static authentication methods such as passwords and tokens, but they also provide continuous and near real-time user verification, this has led to a significant amount of interest in behavioral data, as an eligible alternative to user authentication, distinguishing it from other methods of verification or identification. The proliferation of internet services and the quick development of computer technology have resulted in the collection of more sensitive and important digital data, as a result in contemporary digital ecosystems, protecting the privacy and security of such data has become crucial. Because biometric authentication techniques may offer continuous, non-intrusive, and user-specific verification, they have become a promising solution to these problems [1].

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

this face of advanced cyberattacks like phishing, credential stuffing, and brute-force techniques, conventional authentication techniques like passwords and PINs are no longer sufficient. In these approaches draw attention to the weaknesses of static, one-time authentication methods, which has led to a move toward behavioral biometrics as a more secure substitute [2,3], physiological biometrics which include fingerprints, iris scans, facial recognition, and palm geometry, and behavioral biometrics, which depend on patterns of human-computer interaction like keystroke dynamics, widget usage, swipe gestures, motion, gait, and most notably, mouse dynamics, are the two main categories into which biometric authentication is typically separated [4].

Mouse dynamics is a unique among behavioral modalities because it is inexpensive simple to implement and considerate of user privacy [5,6] and mouse-based authentication relies entirely on monitoring users' natural interactions with the computer, as opposed to physiological characteristics that call for specific gear, we have a perfect for continuous and real-time authentication with little user interference. Mouse dynamics assesses behavioral traits that are hard to mimic, making them resistant to impersonation, including clicking behavior, pause intervals, acceleration, movement speed, and cursor path [7,8], and this method works well in desktop and web-based settings where passive, persistent verification is needed.

The validity of mouse dynamics for user authentication has been confirmed by numerous researches. For example, [9] presented a 92% accurate binary classification model for mouse data gathered during gaming sessions using Random Forest. In a similar vein, [5] assessed several machine learning models and proved that mouse behavior might serve as a reliable indicator for ongoing confirmation. On the DFL and Balabit datasets, the LT-MAuthen model, which was introduced in [6], obtained AUC scores above 98% by utilizing both local and global patterns in mouse behavior.

Hybrid models that combine mouse and keystroke functionality have demonstrated better outcomes in secure applications [11]. Additional efforts, including [10] and [7] focused on integrating behavioral biometrics into adaptive authentication systems, particularly for web and financial applications. In addition, [8] investigated frequency-domain analysis of mouse data and demonstrated encouraging results with an Equal Error Rate (EER) of 7.46%.

Even with the encouraging results, there are still significant gaps in the current body of research. With little to no support for identifying unknown or unseen users, many models are limited to binary classification (genuine vs. imposter). Others hinder their practical implementation by depending on extensive deep learning models that are not tuned for real-time performance. Moreover, security is compromised by the fact that behavior-based systems are still vulnerable to synthetic and imitation assaults.

This study suggests a scalable, lightweight, and reliable mouse-dynamics-based multi-class user authentication system to get around these restrictions. By adding artificial "Unknown" classes produced by a Conditional Generative Adversarial Network (CGAN), the issue is reformed into a 240-class classification task using the full 2020 SapiMouse dataset, which contains behavioral data from 120 actual users. To normalize distributions and lessen the effect of outliers, thirty manually created behavioral features were taken from session segments and preprocessed using MinMax Scaling and Quantile Transformation. With an inference time of just 0.2331 seconds per sample, the classification model achieves a macro-averaged F1-score of 0.991 and test accuracy of 99.24% because of its neural architecture, which is inspired by EfficientNet and tuned for quick inference and minimal processing overhead.

Unlike earlier research, this study uses the whole 2020 SapiMouse dataset of 120 real users and presents an innovative lightweight user identification system grounded on a small EfficientNet-inspired neural network design. By adding 120 artificial "Unknown" users created with a Conditional GAN (CGAN), we rephrase the authentication issue as a 240-class multi-class classification challenge.

In this work, we use CGAN-generated impostor behavior in a new way to specifically train the model to detect and reject out-of-distribution behaviors, unlike previous approaches which rely solely on actual session data. Additionally, to classification statistics, empirical validation with confusion matrices shows the model's strength and ability to perform near perfect discrimination even among noisy user classes. This integrated pipeline via mouse dynamics demonstrates a new standard for safe, functional, and scalable real-time user authentication with behavioral feature engineering, adversarial data synthesis, and compact model design.

Essential contributions of this research include creating a dataset from mouse sessions into a fixed set of 50 events, from which 30 behavioral features were derived based on motion and statistical aspects. The data were normalized based on distribution using Quantile Transformation and MinMax Scaling prior to applying one hot encoding through LabelEncoder, which are required by the SoftMax classification layer. The models

DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

inference latency was 0.2331 seconds per example and training took 158.25 seconds with an accuracy of 99.24% and a macro-averaged F1-score of 0.99.

2. Related work

[12] addressed the problem of limited training data and feature dependency in mouse dynamics authentication. They proposed a deep learning architecture using Fully Convolutional Networks (FCN) trained on raw mouse movement data over 15-second segments. This method eliminated the need for handcrafted features by automatically extracting patterns from temporal data, and model was trained on the SapiMouse dataset and achieved an AUC of 0.94, demonstrating high potential for real-time behavioral authentication systems.

in [13] investigated the vulnerability of authentication systems to spoofing and replay attacks, to mitigate these threats, they developed a CNN-LSTM-based model that analyzed the similarity between mouse movement trajectories across sessions, their approach enabled both user authentication and CAPTCHA inconsistency detection within a unified framework, this model achieved an AUC of 97.7% for authentication tasks and 94.3% for CAPTCHA inconsistency detection, highlighting its dual-purpose effectiveness.

The authors in [14] addressed the challenge of benchmarking authentication techniques under high-intensity usage. They evaluated several traditional machine learning and deep learning models on a 40-user dataset. Among these, the 1D-CNN achieved the highest binary classification accuracy of 85.73%, while the Artificial Neural Network (ANN) reached 92.48% for multi-class classification. These results reflect the effectiveness of deep models under noisy operational conditions.

In [15] the problem of capturing continuous mouse-based authentication features was tackled using recurrence plots, the authors generated images from spatial, temporal and velocity-based behaviors, By applied a 2D Convolutional Neural Network (2D-CNN) for classification, using the Balabit dataset, they obtained an AUC of 0.9688 and an Equal Error Rate (EER) of 0.0789, highlighting the robustness of recurrence-plot-based modeling.
To support low-resource environments. In [16] proposed a lightweight mouse dynamics authentication system using a K-Nearest Neighbor (KNN) classifier combined with simple random sampling, their model achieved a classification accuracy of 92% and a False Acceptance Rate (FAR) of 0.037, making it suitable for real-world applications requiring efficiency and reliability.

In [17] focused on enhancing silent user authentication through image-based aggregation of mouse dynamics. The authors developed a model that converts session-level mouse activity into composite images, which were then analyzed using deep learning, to be ensure explain ability, Grad-CAM visualization was applied. Experiments conducted on data from 10 users showed promising results in both classification and model interpretability.

In this work [18] the authors addressed the challenge of avoiding manual feature engineering in behavioral biometrics, they proposed a simple yet effective model using two-layer Bidirectional GRU (BiGRU) trained directly on raw (x, y, timestamp) mouse coordinate sequences. Despite its simplicity the model achieved 95.39% accuracy, AUC of 98.65%, and a low EER of 4.34% on the Balabit dataset, indicating the power of recurrent networks on raw input.

In [19] the authors has investigated insider threat detection using behavioral biometrics. developed an ensemble learning model with frequency domain feature extraction and proposed a legality scoring scheme to assess session authenticity based on mouse activity. Using the Balabit dataset, they reported an AUC of 96.47% and an EER of 7.46%, demonstrating the viability of their approach for internal security profiling.

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242

DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

3. Methodology

This type of user authentication based on mouse dynamics revolves around the typical phases of preprocessing and modeling, as shown in Fig. 1. Raw mouse activity logs are initially maintained in a (CSV) format, holding sequential data of user interactions. Quantile Transformation is applied first for features to follow an equal distribution, thereby diminishing the effect of outliers. Afterwards, under-sampling is applied to balance classes and maintain equal segments per user for all 120 users. Each session is split into blocks of fixed size (50 rows per segment), and from each segment, 32 statistical and behavioral features (such as mean, standard deviation, velocity, and acceleration) are extracted. Label Encoding follows to represent user IDs as integers. A Conditional Generative Adversarial Network (CGAN) is then leveraged to generate artificial user segments to enrich the minority classes and bring synthetic diversity. Subsequent to these steps of encoding, One-Hot Encoding must be performed to allow compatibility for classification training. Next, Min-Max Scaling is used to normalize features within the [0,1] range. The prepared dataset is subsequently split into stratified training and testing sets (80/20) to preserve class distribution, classification is done via a lightweight neural architecture inspired by EfficientNet. It contains three dense layers of 128, 64, and 32 units, respectively and with batch normalization and dropout after each layer to prevent overfitting, class weights help manage leftover imbalances within the dataset. EarlyStopping ensures model prevention of overfitting by checking real-time training progress. The test set receives an evaluation based on performance indicators that include accuracy and loss as well as inference time.

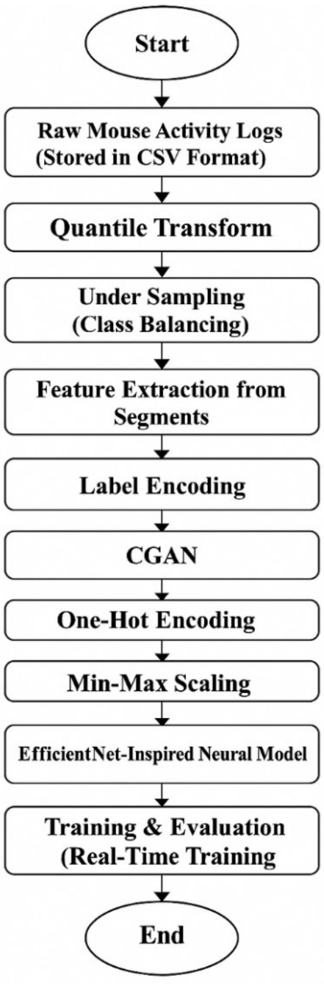
Start
Raw Mouse Activity Logs (Stored in CSV Format)
Quantile Transform
Under Sampling (Class Balancing)
Feature Extraction from Segments
Label Encoding
CGAN
One-Hot Encoding
Min-Max Scaling
EfficientNet-Inspired Neural Model
Training & Evaluation (Real-Time Training
End

Fig.1. Flowchart of Methodology

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

3.1 Dataset [12]

The SapiMouse dataset was put together in 2020 and has mouse movement data from 120 people (92 men and 28 women). They were all students from Sapientia University and were between 18 and 53 years old.

A. How the data was collected

This data was gathered using a JavaScript web app on the users' own computers and anyone can access the data collection app, participants used different browsers and mice, and even some touched the trackpad, with all sorts of DPI settings, when people interacted with their devices, their movements were tracked. The game screen was split into two sections: the top showed the remaining time and the action they needed to take, while the bottom was where they actually played the game, and Once the time was up the collected data was saved.
Participants had to do four different actions based on different shapes:
1) Click on a triangle.
2) Right-click if a reversed triangle appeared.
3) Double-click on a square.
4) Drag and drop something onto a circle.

The aim was to get as many mouse actions done as possible in the limited time. Each subject completed two rounds: one lasting three minutes and the other lasting one minute.

B. Session files

Each user gets a dedicated folder with two files. One file has data from a 3-minute session, and the other is for a 1-minute session. The files are named like this: session [date] [1min|3min].csv, where the date is when the data was collected. Each line in the file shows a mouse event with these details: (timestamp, button, state, x, y). Here, timestamp is how much time has passed since the session started, button shows the status of the mouse buttons, state tells us what's happening with the mouse, and x and y are the cursor's position on the screen.

3.2 Quintile transform and minmax normalization

Prior to training the model, a pre-defined preprocessing pipeline was applied to a 1,184,434-mouse movement record dataset. x, y, and client timestamp features were exposed to Quantile Transformation in order to scale their distributions by transforming non-uniform input into a uniform distribution. This was required to reduce the impact of outliers and offer stable feature scaling for sessions of varying users. Subsequently, during training time, a MinMax Scaling operation was applied to scale the transformed features back to the [0, 1] range. The two-step normalization procedure was designed to enhance training stability, prevent feature dominance, and facilitate more efficient convergence of the deep learning model.

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242

DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

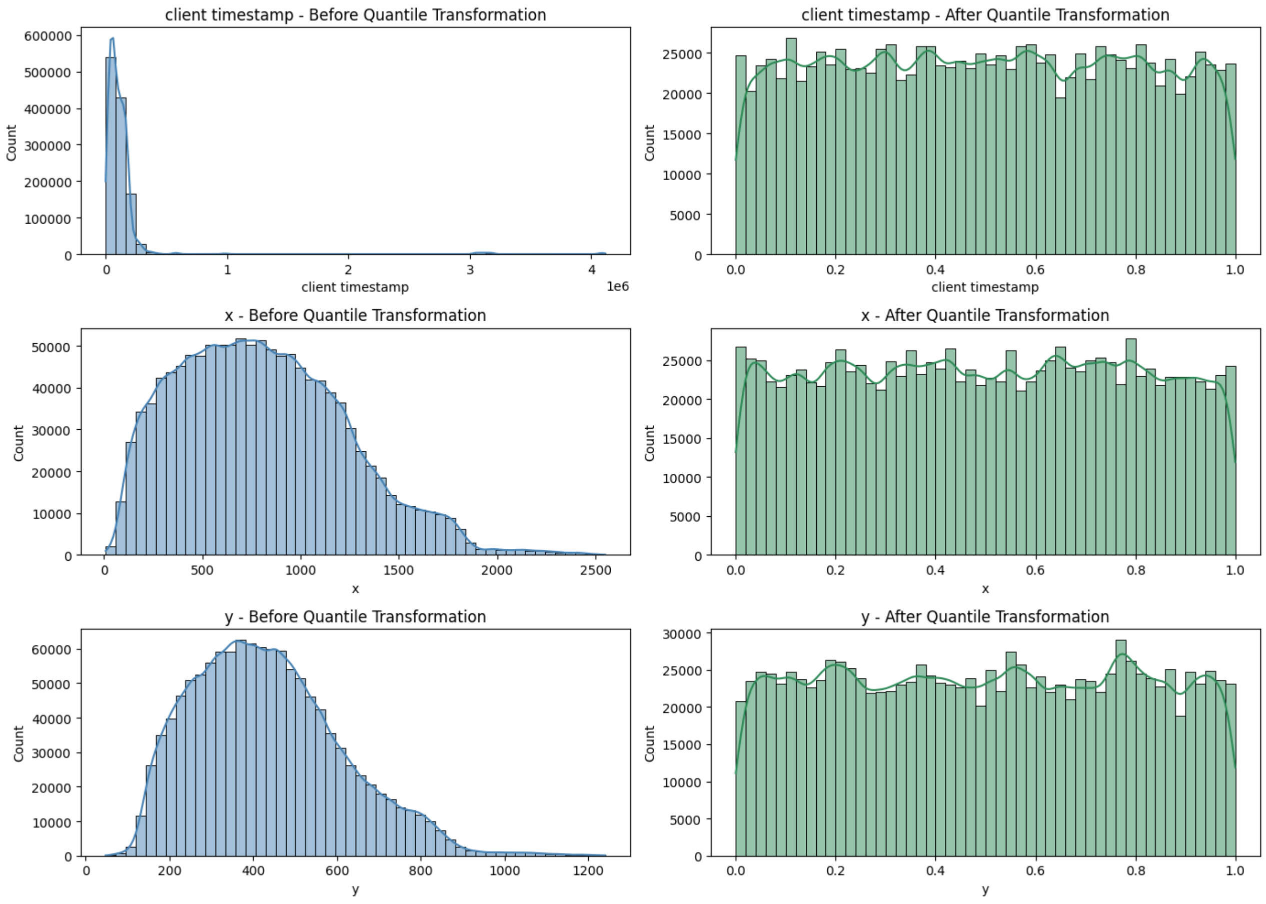

Fig.2. Effect of Quantile Transformation on Mouse Dynamics Features

3.3 Data Balance using Under Sampling Strategy

Before segmenting the mouse movement sessions into segments of a fixed length of 50 rows, there were 474,000 segments created. The segments were derived from approximately 9,480 full user sessions for all users. The original dataset consisted of 120 users, but due to the unbalance in the number of segments per user, ranging from a minimum of 79 segments to a few hundred per user, under-sampling was applied. This technique was used in order to keep all the users with the same number of segments for every class. Therefore, the resulting balanced dataset is comprised of almost 4,000 segments for every user with an equal multi-class classification environment without losing any user identity. As can be observed from the Figure 3 below, the number of user samples is heavily skewed. Users have more than 30,000 samples, and the smallest users 3969 samples, thus a clear unequal distribution. This significant imbalance may bias the model towards the large sample classes, decreasing its performance in recognizing underrepresented users. Therefore, an under-sampling technique was used to decrease the quantity of samples from the large classes and ensure all classes have an equal number of samples 3969. This achieves better balance at training and improves the performance of the model in recognizing all users equally.

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

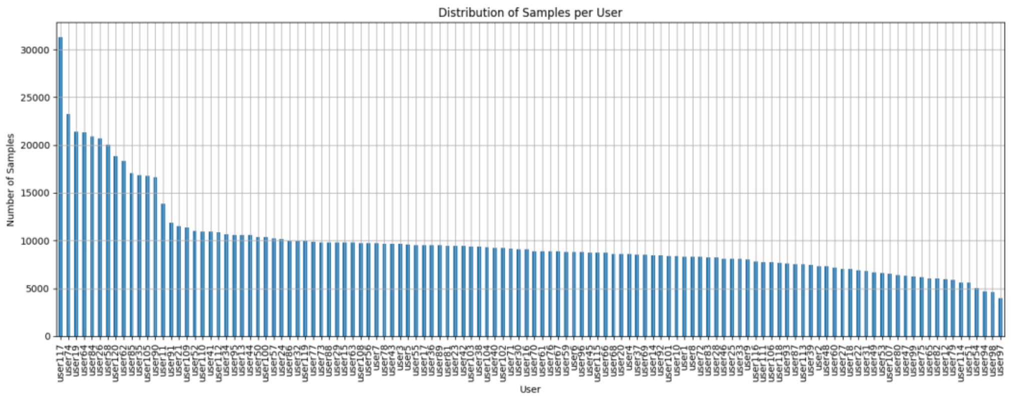

Fig.3. Distribution of Samples per user

Removing majority samples randomly with or without replacement reduces excessive majority samples and decreases training time of the predictive model with potential to exclude helpful information [20-30]. In this multi-class user classification task research, the dataset consisted of 120 users, each with a different number of session segments, to retain each users in the training process and ensure fair representation across classes, we adopted a random under sampling strategy. This approach allowed us to balance the dataset by reducing the number of segments from users with abundant data to match the user with the lowest available number of segments (79 segments). This method, although potentially discarding valuable instances, is practical and necessary in scenarios where data uniformity across classes is crucial. As supported by [31], "random under sampling helps in balancing target distributions and reducing the classifier's bias towards the majority class," particularly when the dataset is sufficiently large to tolerate data reduction without performance degradation.

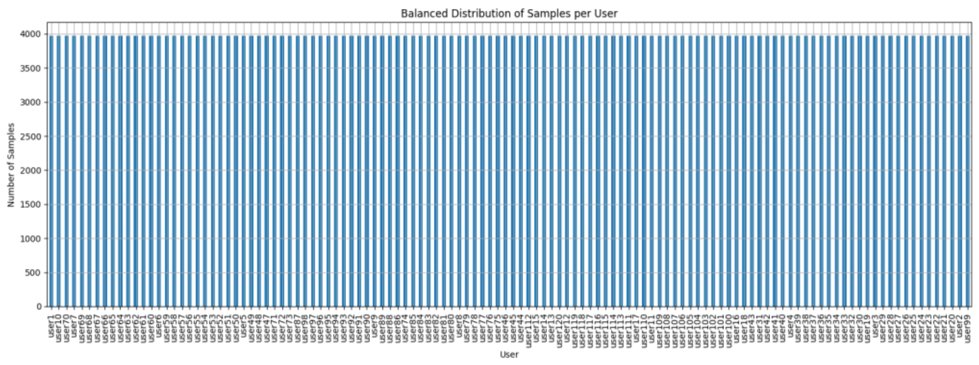

Fig.4. Balance Distribution of Samples per user

As shown in Fig.4, this bar plot indicates the distribution of samples per user after applying the under-sampling technique. As can be seen, all 120 users now have the same number of approximately 4,000 segments, preserving class balance for the multi-class classification problem. This balance addresses the high-level imbalance in the original dataset and allows the model to learn from all users equally without showing bias towards users with a large amount of data.

3.4 Feature extraction from segments

The aim of the processes dealing with mouse dynamics features extraction is to identify the specific characteristics associated with a user. The primitive data collected from the mouse can be used to build a mouse dynamics signature for the user. A mouse dynamics signature is built by adding new features that are used alongside features already described in [28]. As to modeling behavior based on mouse activity, the data was first segmented into 50-event blocks using the x, y, and timestamp attributes. This step preserves the input dimension across all user sessions, allowing training to be conducted irrespective of session duration. After the segmentation, 30 features were extracted from each segment, representing user interactions as well as spatial and temporal movements. Which these features were chosen with regard to the motion dynamics, their statistical signature and cost of computation. The training features are as follows:

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242

DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

TABLE I. The Training Features

Category	Features
Spatial Features	x_mean, y_mean, x_std, y_std, delta_x_mean, delta_y_mean
Velocity & Acceleration	velocity_mean, velocity_std, velocity_max, acceleration_custom_mean, acceleration_custom_std
Curvature & Direction	curvature_mean, curvature_std, direction_change_sum
Pause & Click Timing	pause_time_mean, pause_time_std, click_interval_mean, click_interval_std
Interaction Features	clicks_mean, button_encoded_mean, state_encoded_mean, button_change_sum, state_change_sum
Efficiency Metrics	movement_per_click_mean, movement_per_click_std, movement_ratio_mean, movement_ratio_std, velocity_click_ratio_mean, velocity_click_ratio_std, total_path_last

Each extracted features were normalized by using Quantile Transformation followed by MinMax Scaling to standardize the input distribution for the neural network.

Descriptive and Statistical Analysis of Extracted Features:

In order to examine the statistical properties of the features extracted, we computed descriptive statistics like mean, std dev, min, max, skewness, and kurtosis for each feature across all user sessions. Features like x_mean and y_mean are approximately around ~0.5. Which is in accordance with normalized screen coordinates. Features like acceleration_custom_std and velocity_max exhibit extremely high variance and skewness, suggesting dynamic and unstable movement patterns. Features delta_x_mean, delta_y_mean, and pause_time_mean have values close to zero with very little variability, suggesting that they have low discriminative power, to determining statistical significance, ANOVA and Kruskal-Wallis tests were employed. It was ensured by the tests that: large number of features (e.g., velocity_std, clicks_mean, acceleration_custom_mean, state_change_sum) had p-values ≈ 0, which indicates extreme differences between users. Delta_x_mean, delta_y_mean, and pause_time_mean had p-values ≈ 1, reporting no difference significant at a user level.
Top 5 Variance Features are:

- acceleration_custom_std
- acceleration_custom_mean
- curvature_std
- velocity_max
- total_path_last

These are very informative features since they have high variance, and thus they are of great help in user classification.

Feature Correlation Matrix

The following heatmap represents the correlation matrix of the 30 features used for training. It presents pairwise Pearson correlation coefficients between features:

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

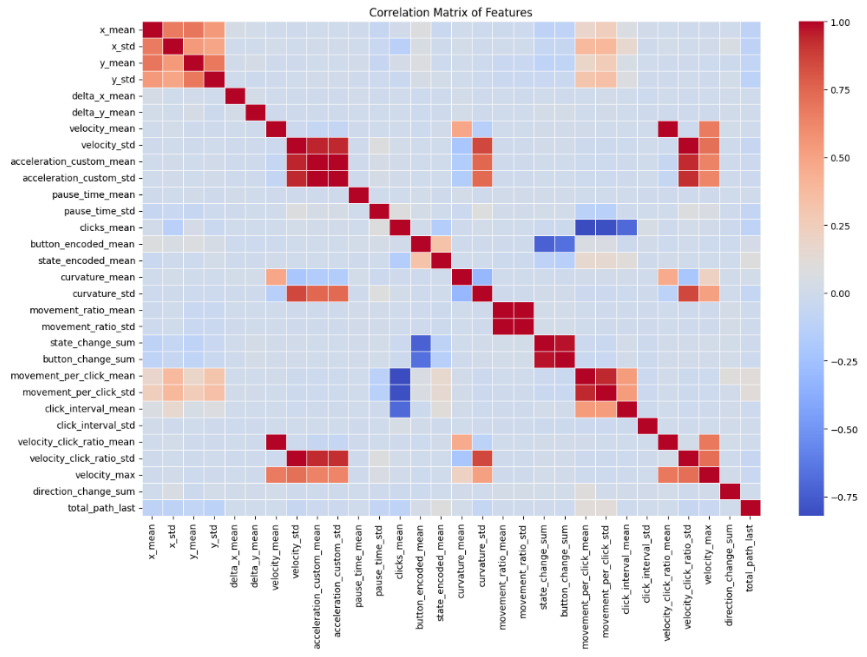

Fig.5. Correlation Matrix Heatmap of 30 Selected Features

- Red cells indicate strong positive correlations (values close to +1).
- Blue cells indicate strong negative correlations (values close to -1).
- White or light-colored cells indicate low or no correlation.

Key takeaways:

- Features like x_mean, y_mean, acceleration_custom_std, and velocity_max are moderately to strongly correlated with other motion dynamics.
- Features related to mouse clicks (e.g., movement_per_click_std, velocity_click_ratio_std, click_interval_std) tend not to be correlated with motion features, suggesting they capture complementary behavioral information.
- Highly correlated features, though present, were retained because they had high variance and statistical significance determined by ANOVA/Kruskal tests.

Data Segmentation

In this article, methodology uses mouse session files that store sequential data of structured mouse events which include timestamp data alongside x-y coordinate information and button states along with other event indicators. The movement dynamics of the system were analyzed using the three numerical aspects of x and y and timestamp values for behavioral modeling. The process began with introducing a segmentation step which divided all sessions into 50-event blocks in order to create standardized input lengths. The system maintains uniform feature dimensions throughout the feature extraction process and model training because of this standardization method. We created behavioral and statistical attributes by analyzing every segment through calculations of x-y coordinate averages and standard deviations combined with derived speed and acceleration measures. The extracted features create a brief and informative profile of how users interact thus allowing the model to detect individual behavioral patterns.

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

3.5 Label Encoding and One-Hot Encoding

For this research, there were 240 user classes: 120 genuine users and 120 synthetic impostor users created by a Conditional Generative Adversarial Network (CGAN). Each user label was originally a string (i.e., user1, unknown35, etc.). These categorical labels were prepared for deep learning classification by employing Label Encoding with scikit-learns LabelEncoder, which transformed each user identity into a distinct integer from 0 to 239.Then the label encodings were converted into one-hot encoded vectors with TensorFlow's to_categorical () function. The encoding resulted in binary vectors of length 240 with the user's class position set to 1 and the rest as 0. This was carried out so that the Softmax output layer of the neural network can be trained using categorical cross-entropy loss. Through the union of label encoding and one-hot encoding, the model was successfully trained to distinguish between a large set of genuine and artificial impostor users, making it more generalized and robust in real-time authentication systems.

3.6 Synthetic User Generation via CGAN

CGAN was constructed with both generator and discriminator conditioned on class labels via the embedding layers. The input features were normalized using Quantile Transformation and MinMax Scaling. The generator consumes random noise vectors and user-specific labels to produce synthetic samples, while the discriminator is trained exclusively to distinguish real and synthetic samples based on the same conditioning. It was trained for over 10,000 epochs using batch-based adversarial training. Regular checks of discriminator accuracy and generator loss were carried out in an attempt to monitor the learning dynamics' progression.
In order to introduce diversity and robustness to the authentication model, synthetic user data was generated using Conditional Generative Adversarial Networks (CGAN) to introduce diversity and robustness to the authentication model. Synthetic users were conditioned on a unique class label and learnt on mouse dynamics' true feature distributions. The CGAN model learned very well to generate realistic behavioral segments that had statistical properties and variability present in true user data. Use of CGAN allowed the dataset to be expanded from 120 real users to 240 classes of total users (120 real + 120 synthetic), thus adding diversity to the classes and reducing the risk of overfitting. The synthetic samples were validated through both visual inspection and statistical tests to ensure their plausibility and separability. The synthetically generated user profiles greatly helped to regularize the classification model and improve its generalizability to new behavioral patterns.

T-SNE Visualization of Real and Synthetic Users

T-distributed stochastic neighbor embedding (t-SNE) is frequently used to project high-dimensional data into a two-dimensional space, allowing for effective visualization and exploration of complex data structures [32], to visually assess the distribution and separability of user representations, a t-distributed stochastic neighbor embedding (t-SNE) projection was applied to the extracted feature vectors of both real and CGAN-generated users. As shown in Fig.6, in 2D embedding highlights distinct and well-separated clusters corresponding to individual users, confirming the discriminative nature of the extracted features. Notably synthetic users (with encoded labels > 120) are interspersed across the space and do not overlap with a single cluster of real users, suggesting that CGAN generated behaviorally diverse yet class-consistent patterns, this spatial distribution supports the conclusion that the synthetic samples contribute to enhancing the model's generalization capabilities without introducing redundancy.

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

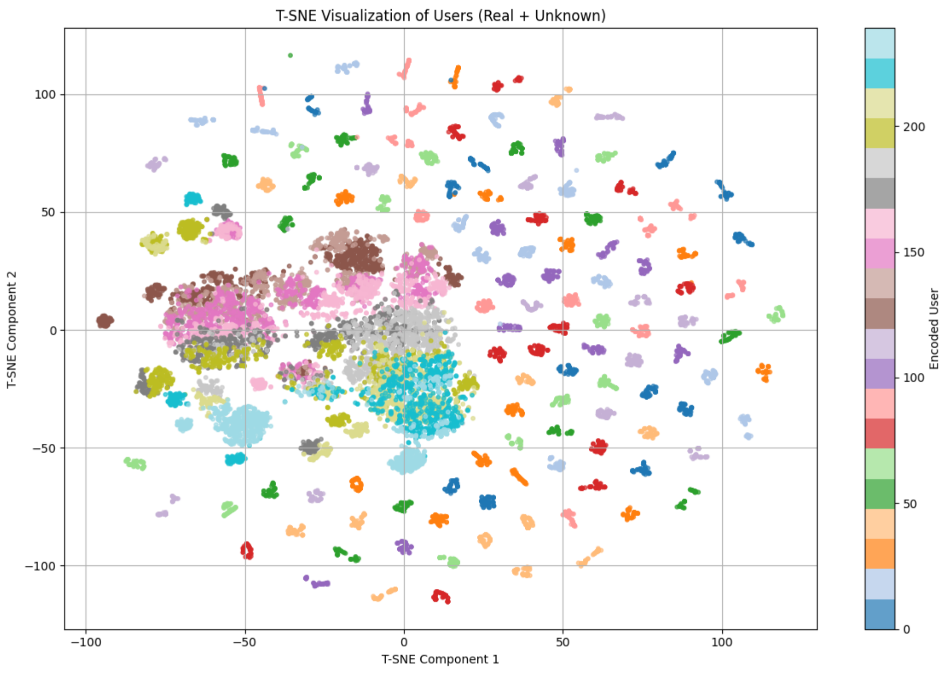

Fig.6. *T-SNE visualization of real and unknown users based on extracted mouse dynamics features.*

As in Fig.6 all color represents a unique encoded user, including synthetic "unknown" users, the separation between clusters indicates the model's ability to distinguish users based on behavioral patterns.

4. EfficientNet-Inspired Neural Model and Real-Time Prediction

Final action in the authentication chain involves a lightweight EfficientNet-inspired neural model, designed for precise classification with a focus on maintaining real-time processing, this model consists of three fully connected dense layers and has utilized class weighting to solve potential class imbalance, especially between the legitimate users and synthetic class Unknown. Once trained the model is evaluated on a holdout test set to monitor performance metrics such as accuracy, F1-score, and latency, once trained the model is deployed into operation for real-time prediction, with incoming mouse behavior segments directly fed into the network for real-time identity classification, so this deployment ensures the system is made to function in real-world environments with no or minimal inference latency, making it eligible for security-oriented applications that require effective and smooth authentication.

4.1 Model Architecture

The study employs a lightweight deep neural network model which uses EfficientNet design principles for tabular data but lacks convolutional layers. The model begins with a set of hand-engineered features (e.g. 30 dimensions) that move through three fully connected layers that decrease the number of units to 128, 64, and 32 units sequentially. All dense layers use swish activation functions to create better non-linearity with smooth gradient propagation. The training process gains stability through Batch Normalization which accelerates training velocity and Dropout helps prevent overfitting. The first, second and third dense layers in Figure 6 receive dropout rates of 0.3 and 0.3 and 0.2 respectively. The final classification layer contains a Dense layer with num_classes number of neurons and Softmax activation performs multi-class classification across user classes. The training optimization techniques focused on incorporating class weighting for class imbalance and early stopping to prevent overfitting and a ReduceLROnPlateau learning rate scheduler which controlled learning rate changes throughout the training process. The training lasted for 200 epochs and the model assessment included accuracy evaluation together with F1-score measurement and inference time analysis to determine suitability for real-time authentication operations.

DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

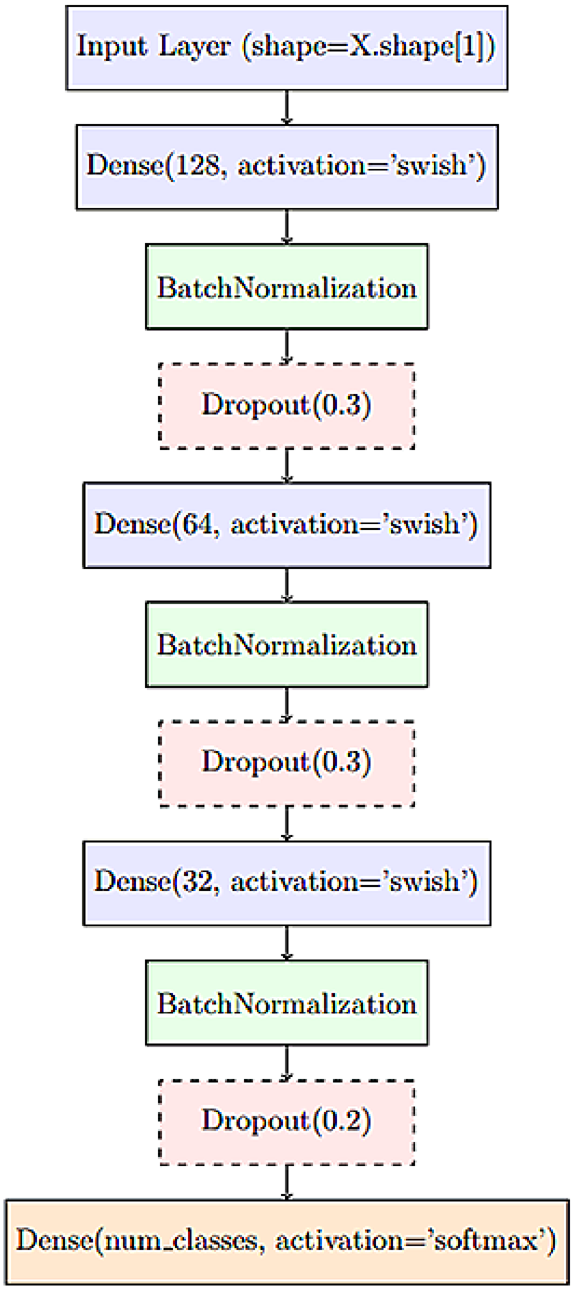
Input Layer (shape=X.shape[1])
Dense(128, activation='swish')
BatchNormalization
Dropout(0.3)
Dense(64, activation='swish')
BatchNormalization
Dropout(0.3)
Dense(32, activation='swish')
BatchNormalization
Dropout(0.2)
Dense(num_classes, activation='softmax')

Fig.7. The Architecture of the Proposed EfficientNet-inspired Lightweight Model.

Figure 7. The neural network model consists of three dense blocks with 128, 64, and 32 units, all utilizing the swish activation function. After every dense block, batch normalization and dropout layers are added to enhance stability and prevent overfitting. The final output layer utilizes a Softmax activation for multi-class classification between all user classes. Optimization during training was achieved using class weighting, early stopping, and adaptive learning rate schedule.

DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

5. Experimental Setup and Model Architecture

Experiments were performed in Python on Google Colab Pro with GPU acceleration using an NVIDIA Tesla T4 (16GB RAM). The environment was TensorFlow 2. x, scikit-learn, pandas, matplotlib and seaborn packages. The training dataset was obtained from the 2020 SapiMouse dataset, which offers mouse interaction data of 120 real users. Activity sessions of mice were divided into equivalent lengthen blocks of 50 trials, and 30 hand-crafted behavior features were calculated from each block. To smooth the features and attenuate the impact of outliers, the Quantile Transformation (output distribution: normal) and MinMax Scaling were performed to stretch the feature-a longs between 0 and 1. User labels have been encoded by means of LabelEncoder from scikit-learn and then converted into one-hot vectors using the to_categorical function that is compatible with the Softmax output layer. The data set was augmented with another 120 impostor classes synthesized using a Conditional Generative Adversarial Network (CGAN) in order to make the model distinguish between enrolled and non-enrolled users better, thus totaling 240 classes (120 real + 120 unknown).
Optimized for behavioral feature inputs, the classification model was a small EfficientNet-inspired neural network. The design comprised of:

The model architecture implemented the Swish activation function and employed 3 fully connected Dense layers with 128, 64, and 32 units, respectively. The model's convergence was accelerated and stabilized during training with batch normalization layers after each dense layer. In addition, dropout layers with rates of 0.3, 0.3, and 0.2 were added after each corresponding block to further reduce overfitting. The final classification was implemented using a Dense layer with 240 output units with a Softmax activation function for multi-class classification on the whole set of known and synthetic user identities. With a final approximation of a macro F1 -score of 0.985, inference time of 0.2331 seconds and a 99.24% accuracy on testing, the model is demonstrated as effective for real-time user authentication.

6. Visual Results and Performance Analysis

To evaluate the classification performance, three key metrics were utilized: **Precision**, **Recall**, and **F1-Score**, which are defined as follows:

- **Precision**: The TP value divided by its sum with FP determines the precision rate. The highest level of accuracy is passed by the 1.0 score, which indicates flawless predictions, but the 0.0 score shows completely incorrect forecasts [33].
$$Precision = \frac{Tp}{TP+FP} \tag{1}$$
- **Recall**: indicates how well a classifier recognizes positive cases through its ability to identify cases as positive [34].
$$Recall = \frac{T_P}{T_P+FN} \tag{2}$$
- **F1-Score**: The equation computes this measure as the harmonic mean of precision and recall values [35].
$$F1score = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \tag{3}$$

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

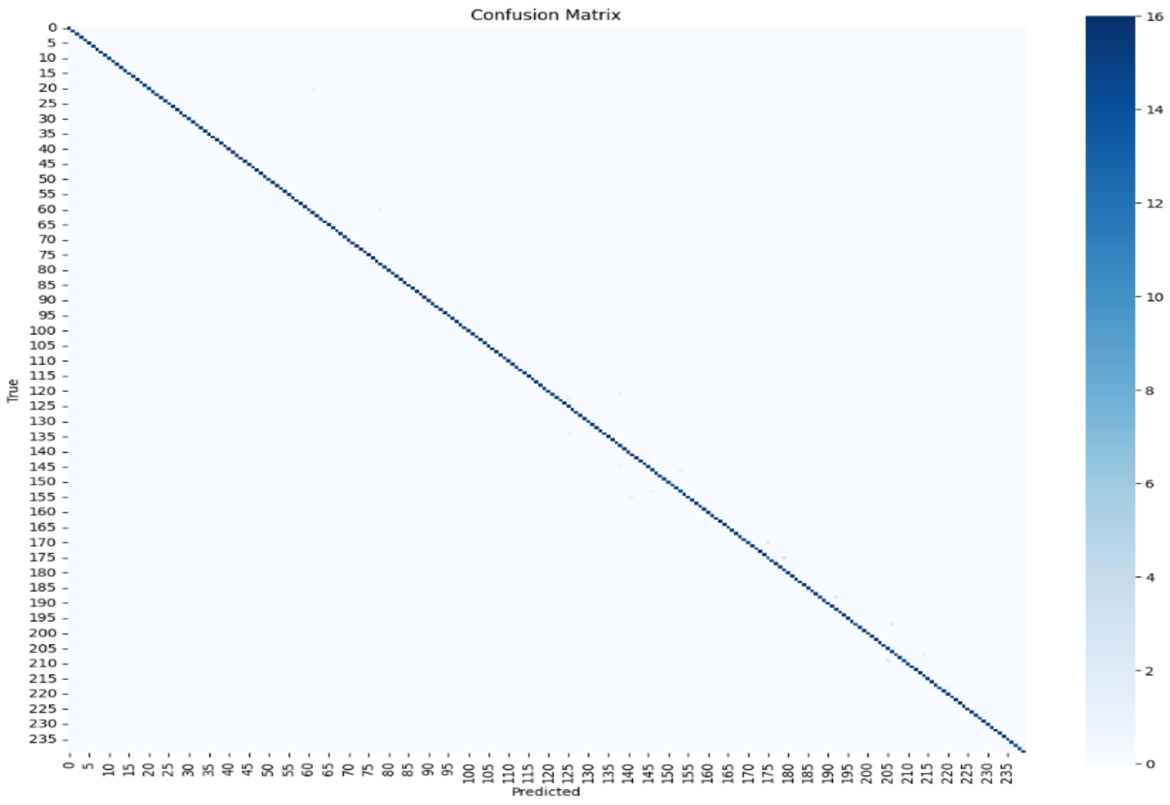

Fig.8. The Confusion Matrices

Figure 8 shows the confusion matrix for the classification model proposed with 240 classes (120 real users and 120 CGAN-generated synthetic users). The abundance of diagonal dominance demonstrates low levels of misclassification. This model was trained, with class weights, on balanced data leveraging dropout and early stopping to improve generalization. We observed a strong macro-averaged F1-score of 0.991 as well as an overall test accuracy of 99.24%. Furthermore, in terms of running time, the time taken to classify a mouse movement was 0.2331 seconds, confirming the successful development of a suitable model for use as an authentication application based on mouse dynamics in real time.

The evaluation confirms that this model maintains robustness and high accuracy rates while offering efficient computation for deployment in operational mouse dynamic authentication systems in urgent applications that require quick responses. Table II presents a brief comparison of the most pertinent previous research in mouse dynamics-based user authentication together with the confusion matrix and classification metrics. Key elements including the problem addressed, the techniques employed, the datasets used, and the outcomes provided are outlined in this table. It provides a comprehensive picture of the development in this area of study and emphasizes the unique features and contributions of the suggested approach.

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

TABLE II. Summary of Mouse Dynamics Authentication Studies

Research	Year	Method	Key Idea	Best Result
[12]	2020	1D-CNN	Raw mouse data for authentication	AUC = 0.94
[13]	2023	Embedding-based classifier	Unified model for auth + CAPTCHA detection	AUC: 94.3%, 97.7%
[14]	2022	1D-CNN, ANN, RF, SVM	Multi-model benchmark for binary & multi-class	1D-CNN: 85.73%, ANN: 92.48%
[15]	2024	2D-CNN	Recurrence plots with velocity (lightweight CNN)	AUC = 0.9688, EER = 0.0789
[16]	2023	VGG16	Data mapping with deep learning	AUC = 0.953, Prec = 0.897, Rec = 0.896
[17]	2024	2-layer BiGRU	Raw sequences, no preprocessing	Accuracy = 95.39%, AUC = 98.65%
[18]	2022	GBM + Frequency Domain	Legality scoring using domain features	AUC = 96.47%, EER = 7.46%
[19]	2023	KNN + Random Sampling	Simple model with lowest FAR	Accuracy = 92%, FAR = 0.038
—	2025	EfficientNet (Proposed)	30 handcrafted features + CGAN for unknown detection	Accuracy = 99.24%, F1 = 0.991, Time = 0.2331s

6.1 Statistical Confidence

Figure 9 shows the results for the EfficientNet-based model. It reports the average Precision, Recall, and F1-score for all 240 user classes—120 real users and 120 unknown impostors—along with their 95% confidence intervals.
The model's scores are:

- Mean Precision: 0.991
- Mean Recall: 0.992
- Mean F1-Score: 0.991

The confidence intervals are narrow, as shown by the error bars in the figure. These metrics were calculated using a stratified test set that includes both real and synthetic user sessions.

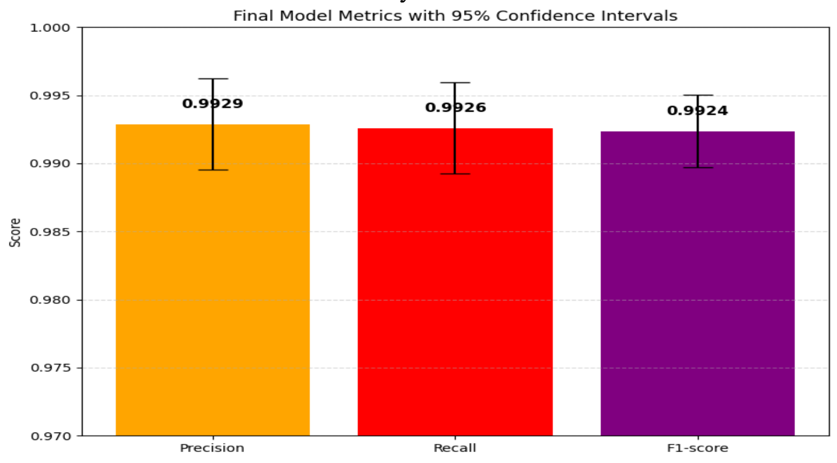

Fig.9. Final Model Performance with 95% Confidence Intervals for Precision, Recall, and F1-Score

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

6.2 K-Fold Cross-Validation and Visualization of Error Bar

The robustness of the model was tested using a 5-fold cross-validation. The performance can be seen in Figure 10, and overall, there was a consistent level of performance across each of the five folds. The accuracy for each fold ranged from 0.9901 to 0.9941, and the average accuracy reported was 0.9920 with a standard deviation of approximately ±0.0014. The results of the 5-fold cross-validation indicate that the model provides consistent results no matter where the data is split.

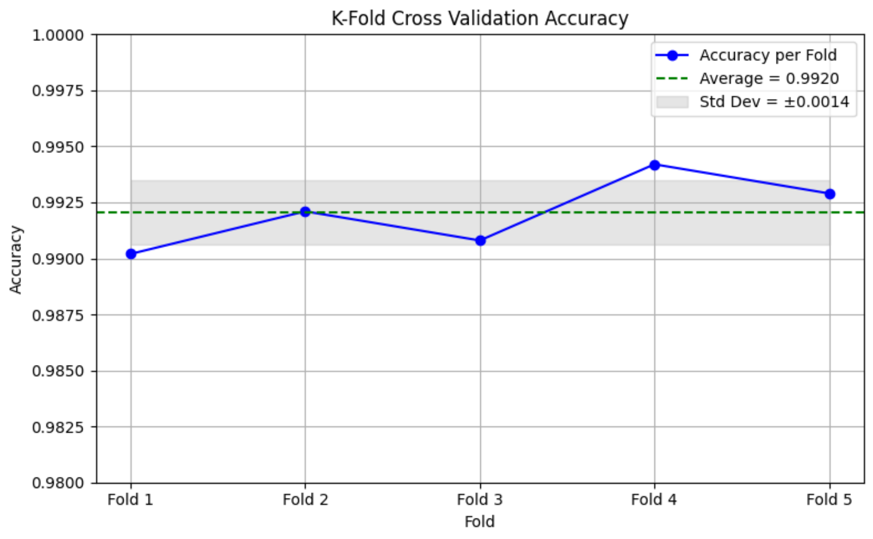

Fig.10. K-fold Cross Validation Accuracy

6.3 Training and Validation Accuracy Analysis

Figures 11 and 12 show the training and validation accuracy and loss over all the training epochs. The accuracy goes up quickly in the first 20 epochs, hitting around 99%. At the same time, the loss goes down for both training and validation. The training and validation curves stay pretty close to each other during the entire process.

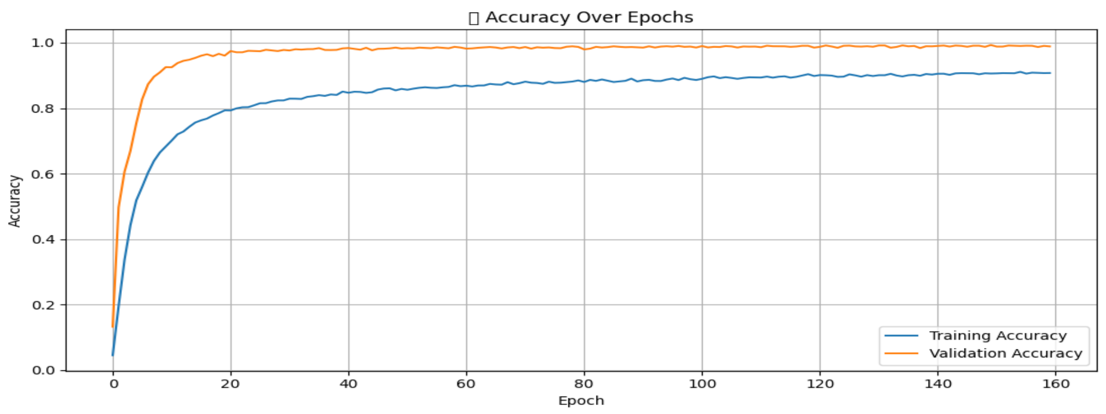

Fig.11. Training and validation accuracy

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

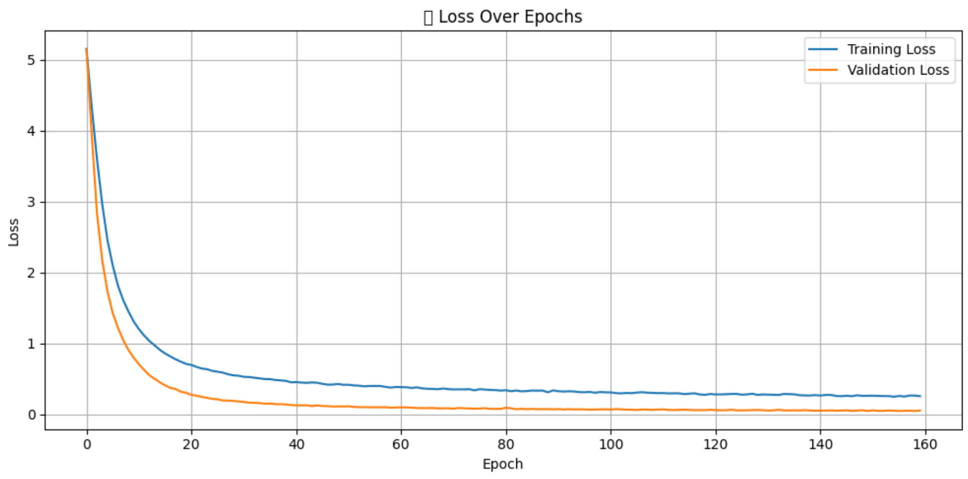

Fig.12. Training and Validation Loss

Figure 13 shows how well the model performed with 240 different classes, which includes 120 real users and 120 fake ones created by CGAN. The graph displays Precision, Recall, and F1-score for each class. The X-axis lists the user classes, starting with the unknown users, and the Y-axis goes from 0 to 1. Most of the user classes scored very close to 1.00 in Precision, Recall, and F1-score. A few classes, like user40, user44, and user68, had slightly lower scores, but all their F1-scores were still over 0.90. Overall, the model nailed a classification accuracy of 99%, with a macro-averaged Precision, Recall, and F1-score of 0.99 across all 240 classes.

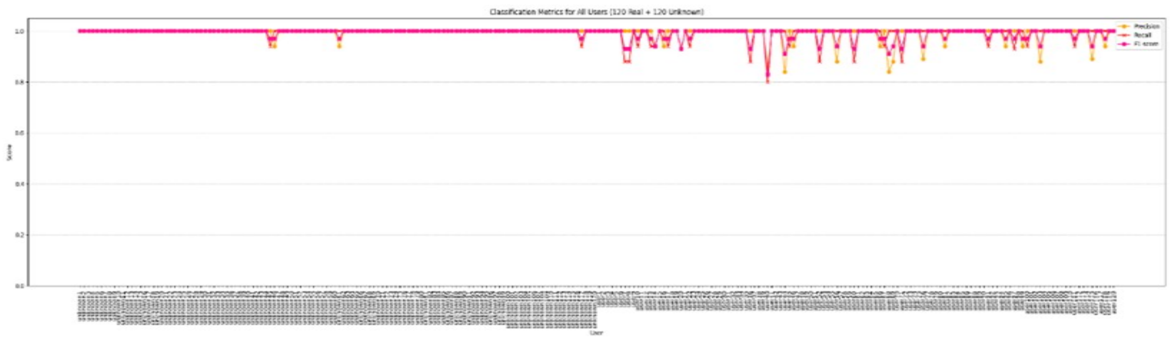

Fig.13. Per-User Classification Metrics (Precision, Recall, F1-Score) for 120 Real and 120 Unknown Users

7. Conclusion and Discussion

In this study introduces a lightweight, real-time user authentication system that uses mouse movements, drawing from the full 2020 SapiMouse dataset, the classification involved 240 categories, including 120 real users and 120 fake “Unknown” users created with Conditional GAN (CGAN). We took 30 manually selected behavioral features from 50-event segments and adjusted them with quantile transformation and MinMax scaling, the EfficientNet-Inspired model we developed which included class weighting and EarlyStopping, achieved a remarkable 5-fold cross-validation accuracy of 99.20% and a macro-averaged F1-score above 0.99. The model performed well across both real and synthetic classes, as the classification metrics and confusion matrices. When compared to past studies on behavioral biometrics and mouse movements, this article results are competitive, especially in adapting to new classes while keeping real-time speed. Visualization methods like the confusion matrix and t-SNE showed the model's strong ability to distinguish different classes, and this system is lightweight and works on different platforms, making it ready for larger deployments. That said, there are some drawbacks. While synthetic data helps balance the

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242
DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

classes, it might not fully reflect the complexity of real impostor behavior. Plus, using fixed-size segments could miss out on understanding longer-term patterns in user behavior. Looking ahead, we aim to explore dynamic segmentation methods, temporal modeling like LSTM, and techniques to adjust the system for different usage environments.

8. Limitations and Future Work

As effective as the proposed system is, some real-life situations still require additional exploring. One of the issues is a mimicry attack in which attackers try to imitate a legitimate user's mouse movements. While the CGAN produced "Unknown" class attempts to enhance dissimilar pattern rejection, there is still the problem of sophisticated impersonation attacks especially if behavioral surveillance is available. Looking at other factors, and user fatigue or stress may affect behaviors and consistency over time, all users who are under such conditions are bound to show at least some unconventional behaviors which deteriorates the model's accuracy, and the current model performed reasonably well on intra-user variability, but extreme emotional or physical states are bound to result in some baseline behavior changes, in these issues we able to be resolved by enhancing the 'Unknown' data via adversarial Ly simulated mimicry behavior and training the model on session data that was collected during different emotional or physical states. At more meaning granular models like BiLSTM or Transformers can track temporal dynamics more effectively, and improve adaptability to changes.

References

[1] Wang, X., Shi, Y., Zheng, K., Zhang, Y., Hong, W., & Cao, S. (2022). User authentication method based on keystroke dynamics and mouse dynamics with scene-irrelated features in hybrid scenes. Sensors, 22(17), 6627.

[2] Bhad, A. (2025). A comparative analysis of behavioral and physiological biometrics in user authentication. University of Oxford. Retrieved from https://www.researchgate.net/publication/389168617.

[3] Liang, W., & Hamzah, F. (2025). Behavioral biometrics and AI for cloud user authentication. Retrieved from https://www.researchgate.net/publication/389717394.

[4] Khan, S., Devlen, C., Manno, M., & Hou, D. (2024). Mouse dynamics behavioral biometrics: A survey. ACM Computing Surveys, 56(6), 1-33.

[5] Dave, R., Handoko, M., Rashid, A., & Schoenbauer, C. (2024). From Clicks to Security: Investigating Continuous Authentication via Mouse Dynamics. arXiv preprint arXiv:2403.03828.

[6] Wang, Y., Wu, C., Liao, Y., & You, M. (2025). Optimizing Mouse Dynamics for User Authentication by Machine Learning: Addressing Data Sufficiency, Accuracy-Practicality Trade-off, and Model Performance Challenges. arXiv preprint arXiv:2504.21415.

[7] Bello, H. O. Integrating Behavioral Biometrics and Machine Learning to Combat Evolving Cybercrime Tactics in Financial Systems.

[8] Yildirim, M., & Anarim, E. (2022). Mitigating insider threat by profiling users based on mouse usage pattern: ensemble learning and frequency domain analysis. International Journal of Information Security, 21(2), 239-251.

[9] Siddiqui, N., Dave, R., & Seliya, N. (2021). Continuous authentication using mouse movements, machine learning, and Minecraft. arXiv preprint arXiv:2110.11080.

[10] Solano, J., Camacho, L., Correa, A., Deiro, C., Vargas, J., & Ochoa, M. (2021). Combining behavioral biometrics and session context analytics to enhance risk-based static authentication in web applications. International Journal of Information Security, 20(2), 181-197.

[11] Sindhu, B., & Kezia Rani, B. (2022). Multi-modal user authentication technique using keystroke, mouse and game dynamics: An effective and secure approach. Metszet Journal, 7(12), 604–615. https://www.researchgate.net/publication/390675845.

[12] Antal, M., Fejér, N., & Buza, K. (2021, May). SapiMouse: Mouse dynamics-based user authentication using deep feature learning. In 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI) (pp. 61-66). IEEE.

[13] Jin, R., Liao, Y., & Zhou, P. (2023). User Authentication and Identity Inconsistency Detection via Mouse-trajectory Similarity Measurement. arXiv preprint arXiv:2312.10273.

[14] Siddiqui, N., Dave, R., Vanamala, M., & Seliya, N. (2022). Machine and deep learning applications to

Iraqi Journal for Computers and Informatics
University of Information Technology and Communications
Vol. 51, No. 2, 2025, pp. 224-242

DOI: https://doi.org/10.25195/ijci.v51i2.612
Print ISSN: 2313-190X, Online ISSN: 2520-4912

mouse dynamics for continuous user authentication. Machine Learning and Knowledge Extraction, 4(2), 502-518.

[15] Mazumdar, K., & Sundaram, S. A Continuous Mouse Dynamics Authentication System Based on a Recurrence Plot Image Representation Framework. Available at SSRN 5073957.

[16] Chandranegara, D. R., Ashari, A., Sari, Z., Wibowo, H., & Suharso, W. (2023). User classification based on mouse dynamic authentication using K-nearest neighbor. Makara Journal of Technology, 27(1), 5.

[17] Ciaramella, G., Fagnano, S., Iadarola, G., Martinelli, F., Mercaldo, F., & Santone, A. (2022, December). Continuous and silent user authentication through mouse dynamics and explainable deep learning. In 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 1791-1798). IEEE.

[18] Houssel, P. R., & Leiva, L. A. (2024). User Re-Authentication via Mouse Movements and Recurrent Neural Networks. In ICISSP (pp. 652-659).

[19] Yildirim, M., & Anarim, E. (2022). Mitigating insider threat by profiling users based on mouse usage pattern: ensemble learning and frequency domain analysis. International Journal of Information Security, 21(2), 239-251.

[20] Wongvorachan, T., He, S., & Bulut, O. (2023). A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. Information, 14(1), 54.

[21] Azeez, R. A., Jamil, A. S., & Mahdi, M. S. (2023). A Partial Face Encryption in Real World Experiences Based on Features Extraction from Edge Detection. Int. J. Interact. Mob. Technol., 17(7), 69-81.

[22] Mahdi, M. S., Abdulhussien, W. R., Najm, H., & Alogali, A. S. M. (2025). Image encryption using modified Serpent algorithm and Harris Hawks Optimization. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 16(1), 154–171. https://doi.org/10.58346/JOWUA.2025.11.009.

[23] Najm, H., Salih Mahdi, M., & Mohsin, S. (2025). Novel key generator-based SqueezeNet model and hyperchaotic map. Data and Metadata, 4, Article 743. https://doi.org/10.56294/dm2025743

[24] Najm, H., Salih Mahdi, M., & Abdulhussien, W. R. (2024). Lightweight image encryption using ChaCha20 and Serpent algorithm. Journal of Internet Services and Information Security (JISIS), 14(4), 436–449. https://doi.org/10.58346/JISIS.2024.I4.027.

[25] Mahdi, M. S., Hassan, N. F., & Abdul-Majeed, G. H. (2021). An improved chacha algorithm for securing data on IoT devices. SN Applied Sciences, 3(4), 429.

[26] Mahdi, M. S., Azeez, R. A., & Hassan, N. F. (2020). A proposed lightweight image encryption using ChaCha with hyperchaotic maps. Periodicals of Engineering and Natural Sciences, 8(4), 2138-2145.

[27] Kumar, K. K., Dinesh, P. M., Rayavel, P., Vijayaraja, L., Dhanasekar, R., Kesavan, R., ... & Alhussen, A. (2023). Brain Tumor Identification Using Data Augmentation and Transfer Learning Approach. Computer Systems Science & Engineering, 46(2).

[28] Najm, H., Hoomod, H. K., & Hassan, R. (2020). A proposed hybrid cryptography algorithm based on GOST and salsa (20). Periodicals of Engineering and Natural Sciences (PEN), 8(3), 1829-1835.

[29] Omran, A. H., Abid, Y. M., Mahdi, M. S., & GH, A. M. (2020). Design of intelligent controller to minimize the power consumption in smart city based on FPGA. Solid State Technology, 63(6), 9120-9136.

[30] Najm, H., Hoomod, H., & Hassan, R. (2021). A new WoT cryptography algorithm based on GOST and novel 5d chaotic system.

[31] Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020, April). Machine learning with oversampling and undersampling techniques: overview study and experimental results. In 2020 11th international conference on information and communication systems (ICICS) (pp. 243-248). IEEE.

[32] Kang, B., Garcia Garcia, D., Lijffijt, J., Santos-Rodríguez, R., & De Bie, T. (2021). Conditional t-SNE: more informative t-SNE embeddings. Machine Learning, 110, 2905-2940.

[33] Erickson, B. J., & Kitamura, F. (2021). Magician's corner: 9. Performance metrics for machine learning models. Radiology: Artificial Intelligence, 3(3), e200126.

[34] Miao, J., & Zhu, W. (2022). Precision–recall curve (PRC) classification trees. Evolutionary intelligence, 15(3), 1545-1569.

[35] Z. C. Lipton, C. Elkan, and B. Narayanaswamy, "Thresholding Classifiers to Maximize F1 Score," 2014, [Online]. Available: http://arxiv.org/abs/1402.1892