Journal of Intelligent Systems and Internet of Things
Vol. 18, No. 01, PP. 80-102, 2026
DOI: https://doi.org/10.54216/JISIoT.180106

Keystroke Dynamics System for User Authentication Using SVM Classifier

Rasha Khalid Ibrahim[1,*], Mays M. Hoobi[1]
[1]Computer Science Department, College of Science, University of Baghdad, 10070, Baghdad, Iraq
Emails: Rasha.Khaled2201m@sc.uobaghdad.edu.iq; mays.m@sc.uobaghdad.edu.iq

Abstract

As people increasingly rely on computers to store sensitive information and interact with various technologies, the need for low-cost, effective security measures has become more critical than ever. One such method is keystroke dynamics, which analyzes a person's typing rhythm on digital devices. This behavioral biometric approach enhances the security and reliability of user authentication systems and contributes to improved cybersecurity. This study aims to reduce authentication risks by encouraging the adoption of keystroke-based verification methods. The research uses a fixed-text password dataset (.tie5Roanl), collected from 51 users who typed the password over eight sessions conducted on alternating days, capturing variations in mood and typing behavior. Seven models were developed, each following a structured seven-phase process. The first phase involved loading the CMU Keystroke Dynamics Benchmark dataset. The second focused on data preprocessing. In the third phase, new keystroke features were engineered from the original dataset. The fourth phase involved feature selection across various types: unigraph (Hold), digraph (Down-Down, Down-Up, Up-Down, Up-Up), trigraph (Hold-Tri), and their combinations. Training and testing were conducted in the fifth and sixth phases using a Support Vector Machine (SVM) classifier, leveraging keystroke patterns for behavioral biometric identification. The final phase focused on evaluating the models. Each model was tested under two scenarios: one where only the first user is treated as the authorized user, and another where the first three users are considered authorized. Each scenario was further divided into two cases based on preprocessing conditions. The models were assessed using multiple performance metrics, including Accuracy, F1-Score, Recall, Precision, ROC-AUC, and Equal Error Rate (EER). The highest achieved results were Accuracy of 99.35%, F1-Score of 94.2%, Recall of 91.8%, Precision of 98.8%, ROC-AUC of 99.56%, and a minimum EER of 0.02. These outcomes demonstrate the effectiveness of the proposed approach in enhancing authentication reliability using keystroke dynamics.

Received: February 02, 2025 Revised: May 21, 2025 Accepted: July 02, 2025

Keywords: Behavioral biometric; Cybersecurity; Feature engineering; Machine learning; Keystroke dynamics

1. Introduction

In today's digital age, the world is increasingly reliant on personal digital devices, all of which require robust protection. However, unauthorized access remains a persistent global threat [1]. The field of cybersecurity ensures that only authorized users can access and interpret sensitive information, thereby safeguarding data from intrusions [2]. While computers offer immense benefits, their security remains paramount—data loss or damage can result in significant consequences for individuals and organizations alike [3]. Secure data transmission must uphold the three foundational principles of information security: authenticity, confidentiality, and integrity [4][5]. User authentication has become a routine part of digital life, with most systems still relying on traditional methods such as usernames and passwords. However, these approaches suffer from well-known vulnerabilities, including password theft, shoulder surfing, brute-force and dictionary attacks, phishing, and speculation [6][7].

To address these challenges, keystroke dynamics has emerged as a promising biometric-based authentication method. This technique analyzes an individual's unique typing rhythm to verify their identity [6]. As a behavioral biometric, keystroke dynamics captures detailed data from user interactions with physical or virtual keyboards [8]. These patterns can be characterized by duration (short vs. long), typing context (restricted vs. unrestricted), and

content type (fixed vs. variable) [9][10]. The key contributions and novelties of this research include the following points:

- One of the key novelties of this study lies in the pre-processing steps, particularly their application to the CMU Keystroke Dynamics Benchmark dataset. The proposed pre-processing method significantly contributed to enhancing the system's security, as evidenced by high F1-score and accuracy values, as detailed in the results section.
- The development of novel keystroke dynamics features through advanced feature engineering. These newly extracted features have been shown to greatly improve the security and reliability of the authentication system.
- A further innovation introduced in this research is the customized data splitting strategy, which plays a crucial role in boosting the performance of the proposed keystroke dynamics system. This strategy has a direct positive impact on evaluation metrics, confirming its effectiveness.

This paper is organized as follows: Section 2 reviews the existing literature on keystroke dynamics-based authentication. Section 3 outlines the common features used in keystroke dynamics. Section 4 provides details about the dataset employed in this study. Section 5 describes the evaluation metrics. Section 6 presents the Support Vector Machine (SVM) classifier used. Section 7 illustrates the framework of the proposed system. Section 8 discusses the experimental results, and the final section concludes the study.

2. Literature Review

A large number of researchers with the intention of using keystroke characteristics to distinguish between authorized and unauthorized users presented keystroke dynamics authentication models. According to these studies, an authentication system may reliably and computationally efficiently learn each user's unique patterns.
In [11], CMU Keystroke Dynamics Benchmark Dataset are used. The method used is neural network architecture. The metrics are used EER=0.049 and accuracy =94.7%.the features used are Hold Time, Key Down-Key Down Time and Key Up-Key Down Time. In [12], utilized a dataset comprising keystroke dynamics data from 150 subjects. Each participant was asked to enter two types of Personal Identification Numbers (PINs): Short PINs: Consisting of 4 digits and Long PINs: Consisting of 11 digits. The method used is CNN. The metric used is EER=4.5.the features used is hold time and flight time. In [13], involved 104 typing samples collected using an Intelligent Keyboard (IKB). The IKB is a self-powered device that converts mechanical stimuli from keystrokes into local electronic signals. The method used is multilayer Deep Belief Network (DBN). The metric used is recognition accuracy; the proposed method demonstrated promising recognition accuracy across the collected typing samples, highlighting its effectiveness in keystroke dynamics identification. The feature used is raw electronic signals, unlike traditional methods that rely on predefined keystroke features (e.g., Hold time, Up-Down time), this approach directly utilizes raw electronic signals generated by the IKB. The DBN autonomously extracts pertinent features from these signals, streamlining the identification process. In [14] the dataset used is GREYC Keystroke dataset. The methods used is Manhattan Distance, Euclidean Distance, Support Vector Machines (SVM) and Random Forest (RF). The metric used is EER, in Manhattan Distance EER= 8.45%, Euclidean Distance EER = 8.67%, SVM EER = 3.61%, and in RF EER = 3.15%.the feature used is Down _Down, up _Up, Hold time, up _Down. In [15] SVM used, the metric used is accuracy=95% and the features used are key press and release events and gender, age category, and handedness. In [16] this article 31 volunteers (5 women and 26 men) aged between 20 to 40 years; each participant provided a password input and 50 swipe inputs on an Android platform. Over 1,500 samples of keystroke and swipe dynamics were collected. The method used is RNN and CNN. The metric used is accuracy and EER and f1_score.the features used is Keystroke Dynamics: Temporal and spatial characteristics of typing behavior, and Trajectory-based features capturing detailed movement patterns during typing. In addition, Swipe Dynamics: Motion patterns, pressure, and other dynamic aspects of swipe gestures. In [17] the dataset comprises keystroke dynamics data collected from 56 subjects using a Nexus 7 touchscreen smartphone. Each participant was asked to type the password "tie5Roanl" 51 times, resulting in 2,856 records. The dataset includes 71 attributes, capturing various aspects of typing behavior such as Hold Time, Up-Down Time, Pressure, and Finger Area. The method used is Dense Neural Network (DNN). The metrics used is accuracy, f1_score, and recall. In [7], this research examines the use of keystroke dynamics as a biometric measure to improve cybersecurity. The researchers proposed RNN models to evaluate keystroke dynamics, which capture the temporal features of typing patterns. This algorithm was learned to identify authorized users and un- authorized based on their distinct typing rhythms. In this research also compared RNNs' performance to that of classic classifiers, such as SVMs, to determine their efficacy in this setting. The key features extracted from the keystroke data include: Up-Down time, hold time, di-graph, from CMU Keystroke Dynamics Benchmark Dataset the proposed models were evaluated using EER=0.066 with SVM classifier. In [18], in this research, the experiment was conducted with 50 participants who were asked to write about their best and worst learning experiences, focusing on both subjects and teachers. The following keystroke features were retrieved: Hold time, Up-Down time, typing speed, digraph features, trigraph features. The machine learning techniques, including SVM, Random

Forst (RF), Naive Bayes (NB), and K-Nearest Neighbors (K-NN), were employed to classify the opinions based on the extracted keystroke features. The average values of precision, recall and f1-score measurements were around 0.62. In [19] the dataset comprises samples from 50 subjects (20 authorized users and 30 un-authorized), all university students in a computer science department. Participants entered their chosen login ID and password (6 to 15 characters) over ten trials during specified periods across a 14-day span. Data collection aimed to replicate typical login times throughout a working day. The methods used are SVM, RF and ANN. The features used are Hold time, Up-Down time and Down-Down time. The ANN showing better results among the three algorithms implemented at 91.8% accuracy. In [20] dataset was created by acquiring both keystroke dynamics and EEG signals simultaneously from 10 users. Each user participated in 500 trials at 10 different sessions (days) to replicate real-life signal variability. The features used are EEG Signals: Statistical, time-domain, and frequency-domain features capturing neural patterns. In addition, keystroke dynamics: features related to typing patterns, such as key press duration and inter-key intervals. In [21] utilized the Buffalo dataset, which includes free-text keystroke data from 148 participants, each providing responses to specific prompts across multiple sessions. The methods used are CNN and a Gated Recurrent Unit (GRU), the features used keystroke timing features included key Hold time, inter-key intervals, and other timing-related metrics derived from the sequence of keystrokes. Features examined typing habits such as the frequency of using specific keys (e.g., shift, control) to capture unique aspects of individual typing styles. Achieved an average accuracy of 94.2% across different test sessions using the CNN-GRU model. In [8], this article use EmoSurv dataset is a recent dataset containing keystroke data for 124 subjects, grouped into five classes: Anger, Happiness, Calmness, Sadness, and Neutral State. The following important characteristics were retrieved from the keystroke data: Hold time, Up-Down time, digraphs D1D2. The researcher has utilized machine-learning techniques, particularly CNN, to classify emotional states based on the extracted keystroke features. Multi-Instance Learning (MIL), SVM used also. Moreover, the best results in MIL_SVM Variable Bags (VB) accuracy = 0.76, precision = 0.80, recall = 0.69 and f1_score = 0.74. In [22] this research suggests a mechanism for user authentication based on typing patterns. The authors designed and implemented anomaly detection methods based on distance metrics and machine learning techniques. The methodologies used in this Artificial Neural Network (ANN) and CNN. The following important characteristics were retrieved from the keystroke data: Down-Down, Up-Down, and Hold times. The dataset comprised 51 users typing a password across 8 sessions conducted on alternate days to capture mood fluctuations. This approach aimed to account for variations in typing patterns over time. The ANN with a negative class achieved an accuracy of 95.05%. In [10] this research presents a novel technique to enhance password authentication by incorporating various keystroke dynamic information. SVM, ANN, and RF are three machine-learning techniques that have demonstrated impressive success in identifying users based on their typing habits. The SVM and RF algorithms, rising to 91.8% and 97%, respectively, have achieved notable accuracy levels. The results of this research show that the RF model performs better than other models in terms of accuracy when machine learning is applied to keystroke dynamics.

3. Kestroke Dynamics Features

Keystroke-dynamics-based authentication is a type of behavior-based authentication method. Often, it cooperates with an existing knowledge-based authentication method to strengthen the security where the knowledge-based authentication by itself is weak with regard to shoulder-surfing attacks [23]

The following are some typical feature types that can be taken from a human keystroke: [24]:-

1. Hold time: This is the period of time that passes between pressing a key (down) and releasing it (up).
2. Up-down time: This is the period of time that passes between a key release (Up) and the next key push (Down).
3. Down-Down: This is the time between two sequential down keystrokes.
4. Up-Up: This is the time between two key releases that follow one another.
5. Tri-graph: Is the elapsed time between the first key press (Down) and the third key press (Down).
6. Finger placement: a camera is necessary for this kind.
7. Keystroke pressure: In this situation, a unique kind of pressure that works with a sensitive keyboard must be applied.

4. CMU Kestroke Benchmark Dataset

This research uses the CMU Keystroke Dynamics Benchmark Dataset, where CMU is short for (Carnegie Mellon's university). This dataset includes keystroke timing information acquired from 51 typists. Each typist has a subject (id) and 8 sessions, with each session consisting of typing 400 times a fixed strong password (. tie5Roanl). The features extracted from raw data were Hold time, Up-Down time and Down-Down time, that generated 31 timing vectors for each of the 51 users [23]. Figure (1) illustrates the contents of original CMU dataset. The total samples of CMU dataset still 20,400, (i.e. 51 subjects multiplied by 400 times, for each subject is typing as the original dataset) [1].

	A	B	C	D	E	F	G	H	I
1	subject	sessionIndex	rep	H.period	DD.period.t	UD.period.t	H.t	DD.t.i	UD.t.i
2	s002	1	1	0.1491	0.3979	0.2488	0.1069	0.1674	0.0605
3	s002	1	2	0.1111	0.3451	0.234	0.0694	0.1283	0.0589
4	s002	1	3	0.1328	0.2072	0.0744	0.0731	0.1291	0.056
5	s002	1	4	0.1291	0.2515	0.1224	0.1059	0.2495	0.1436
6	s002	1	5	0.1249	0.2317	0.1068	0.0895	0.1676	0.0781
7	s002	1	6	0.1394	0.2343	0.0949	0.0813	0.1299	0.0486
8	s002	1	7	0.1064	0.2069	0.1005	0.0866	0.1368	0.0502
9	s002	1	8	0.0929	0.181	0.0881	0.0818	0.1378	0.056
10	s002	1	9	0.0966	0.1797	0.0831	0.0771	0.1296	0.0525
11	s002	1	10	0.1093	0.1807	0.0714	0.0731	0.1457	0.0726
12	s002	1	11	0.0887	0.166	0.0773	0.0876	0.156	0.0684
13	s002	1	12	0.0911	0.1525	0.0614	0.0824	0.1516	0.0692
14	s002	1	13	0.1114	0.162	0.0506	0.09	0.1547	0.0647
15	s002	1	14	0.0903	0.1871	0.0968	0.0805	0.1919	0.1114
16	s002	1	15	0.1169	0.2562	0.1393	0.0739	0.1549	0.081
17	s002	1	16	0.127	0.1839	0.0569	0.0911	0.1381	0.047
18	s002	1	17	0.1016	0.1799	0.0783	0.0792	0.1434	0.0642
19	s002	1	18	0.1056	0.1755	0.0699	0.0781	0.1391	0.061
20	s002	1	19	0.1177	0.2237	0.106	0.0837	0.188	0.1043
21	s002	1	20	0.1027	0.1781	0.0754	0.0729	0.1418	0.0689
22	s002	1	21	0.1016	0.1374	0.0358	0.0861	0.1629	0.0768
23	s002	1	22	0.1072	0.2217	0.1145	0.0726	0.1349	0.0623
24	s002	1	23	0.1243	0.1841	0.0598	0.0768	0.1568	0.08
25	s002	1	24	0.1241	0.2019	0.0778	0.0829	0.1745	0.0916
26	s002	1	25	0.1098	0.1567	0.0469	0.0768	0.1642	0.0874
27	s002	1	26	0.0916	0.1691	0.0775	0.0818	0.1502	0.0684
28	s002	1	27	0.1272	0.195	0.0678	0.0824	0.1442	0.0618
29	s002	1	28	0.0874	0.1726	0.0852	0.0747	0.1243	0.0496
30	s002	1	29	0.0948	0.1979	0.1031	0.076	0.1412	0.0652

Figure 1. Contents of Original CMU Dataset

The subject must type the 10 characters of the password correctly, in sequence, and then press Enter. If any errors in the sequence are detected, the subject is prompted to retype the password. The subject must type the password correctly 50 times to complete a data-collection session. Whenever the subject presses or releases a key, the application records the event (i.e., key down or key up. CMU dataset subjects consisted of 30 males and 21 females, with 8 left-handed and 43 right-handed subjects. The median age group was 31–40, the youngest was 18–20 and the oldest was 61–70. The subjects' sessions took between 1.25 and 11 minutes, with the median session taking about 3 minutes [22] [25].

5. Evaluation Metrics

Evaluation metrics offer insight into several facets of the deep learning model's efficacy. A confusion matrix is useful for assessing deep learning models, as Table (1) displays. Columns in the matrix display actual classes, while expected classes are represented by rows. Among the most important keywords are:

- The number of approved transactions that are accurately categorized as authorized is known as the True Negative (TN).
- The quantity of unauthorized transactions that are accurately identified as such is known as the True Positive (TP).
- The number of permitted transactions that are mistakenly categorized as un-authorized is known as a false positive (FP).
- The number of unauthorized transactions that are mistakenly categorized as approved transactions is known as the False Negative (FN).

Table 1: Confusion matrix

		Actual	
		Positive	**Negative**
Predicted	**Positive**	TP	FN
	Negative	FP	TN

Six metrics are taken from the confusion matrix in this study to assess the predictive performance of the suggested keyboard authentication mechanism. Accuracy (Acc), F1-Score, Recall, Precision, ROC-AUC, and Error Equal Rate (EER) are the names of these metrics. The following are the formulas used to calculate each of these measurements.- [7] [23] [26] [27]

- F1-score: like in Equation (1), offers a trade-off between Precision (P) and Detection Rate (DR).

$$F1 - \text{score} = 2 * (DR + P) / (DR + P) \quad (1)$$

- Accuracy (Acc): frequently used for evaluating the model's overall performance in categorization tasks. According to Equation (2), it is the proportion of accurately predicted cases to all instances.

$$\text{Acc} = \frac{TP + TN}{TP + FN + TN + FP} \quad (2)$$

- EER: is a common metric used to evaluate the performance of biometric authentication systems (such as fingerprint, face recognition, or keystroke dynamics). EER is the point where the FP equals the FN.

$$FP(\tau) = FN(\tau) \quad (3)$$

Where:

- τ is the threshold used for classification (e.g., match score, distance).
- FP(τ) = False Positive at threshold τ
- FN(τ)= False Negative at threshold τ

$$EER = \min_{\tau} |FP(\tau) - FN(\tau)| \quad (4)$$

- Recall: assesses how well the model can identify unauthorized transactions. According to Equation (5), it is the proportion of accurately predicted positive instances to all instances in the actual class.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

- Precision: indicates the frequency with which a model accurately identifies unauthorized, as specified in Equation (6)

$$P = TP/(TP + FP) \quad (6)$$

- The Area Under Curve (AUC) method evaluates the general performance of the model by computing the whole two-dimensional area under the ROC curve as indicated in Equation (7).

$$\text{AUC} = \frac{1 + \frac{TP}{TP + FN} - \frac{FP}{TN + FP}}{2} \quad (7)$$

- The ROC curve is a graphical diagram that shows how well a binary classifier system can diagnose problems as its discrimination threshold is changed.

6. SVM Classifier

SVMs are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. SVM constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space for classification or regression. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier. Multiclass SVM aims to assign labels to instances by using SVMs, where the labels are drawn from a finite set of several elements. The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems and separate between of them, SVM utilized for accurate and fast classification. [25] [28] [29] [30]. Figure (2) presents the SVM classifier [31].

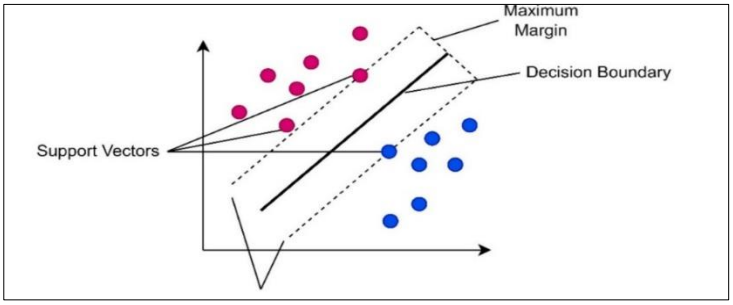

Figure 2. SVM Classifier

The hyperplane equation illustrated in Equation (8)

$$wTX + b = 0 \quad (8)$$

Where:

w: is the weight vector.

x : is the input feature vector.

b : is the bias term.

The decision function illustrates in equation (9)

$$FK(x) = wkT + bk \quad (9)$$

The predicted class equation as follows in equation (10)

$$y\hat{} = \arg\max(fk(x)), K = 1,2 \quad (10)$$

$y\hat{}$: represents the predicted class.

7. Proposed Keystroke Dynamics System (SVM-KD)

The authentication via keystroke is based on the idea that each user posse's unique typing dynamics, this research proposed a keystroke dynamics system for user authentication by exploiting the SVM algorithm, for short this proposed system called (SVM-KD). To obtain the efficient keystroke dynamics system, seven models applied using different keystroke timing features, some of these with features originally founded as default in data set such as unigraph feature, and di-graph features, in addition to new generated features to enhance the efficiency of SVM-KD. Additionally for each model consists of two scenarios according to the number of legitimate users for each class in training and testing phases, finally, each scenario consists of two cases according to preprocessing status. For more illustration, see Figure (3).

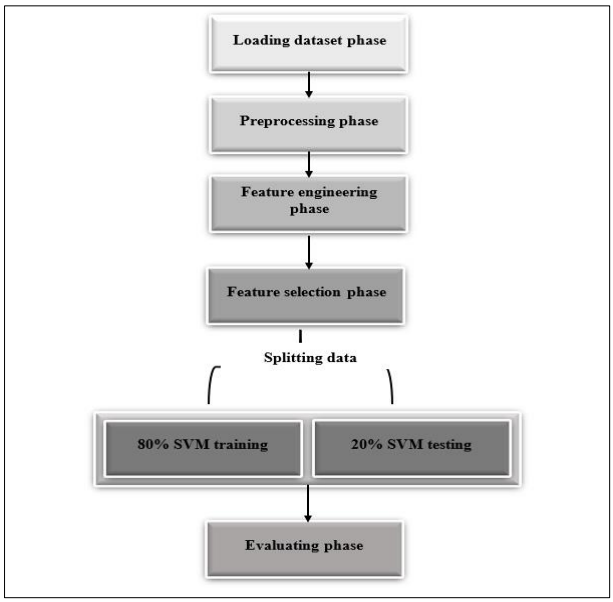

Figure 3. General Structure of Proposed System SVM-KD

Each proposed model of SVM-KD consists of seven phases, as illustrated in Figure (4).

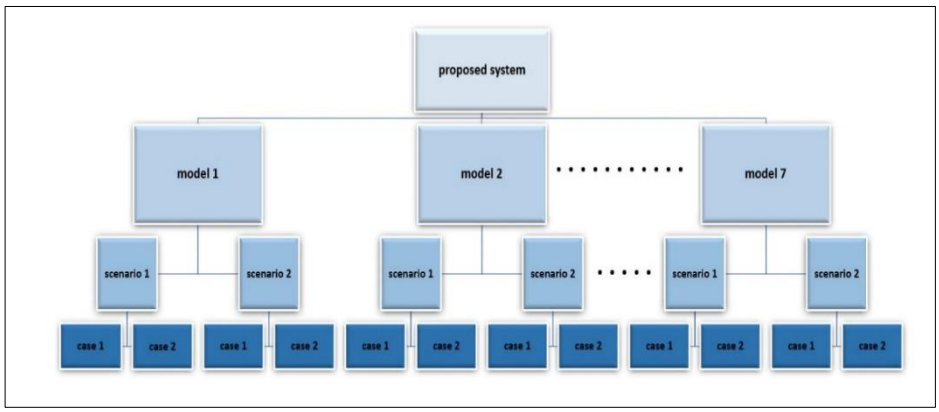

Figure 4. Seven Phases of Proposed

The following subsections will explain the seven phases of the proposed models

7.1 Preprocessing Phase

In this research, the SVM-KD used several steps of preprocessing for standard CMU Keystroke Dynamics Benchmark dataset features for all models, that able the proposed models to be very efficient for keystroke authentication. Figure (5) demonstrates the proposed preprocessing steps. The initial step is to import the dataset into the system for further processing. To prevent training mistakes, the dataset is verified for missing (NaN) or infinite values. If such values are discovered, suitable steps (such as imputation or removal) are implemented. Ensures that the dataset has a 'subject' column, which is required for labeling the data. Each row in the dataset has a binary label: 1 (authorized): If the 'subject' column meets predefined approved values. 0 (unauthorized) → Otherwise. Machine learning models primarily operate with numerical input; therefore, only numeric characteristics are picked for processing. The dataset is segmented into x (features). Input variables for the model. Noise filtering using Z-Score: This stage uses Z-score computation to eliminate noisy data points and detect outliers. Data points that surpass a predetermined Z-score threshold (2.5) are considered noise and removed. The cleaned dataset is divided into training and testing sets for evaluating the model's performance. The numeric characteristics are standardized with StandardScaler to ensure that all variables have the same scale, allowing the model to learn more successfully.

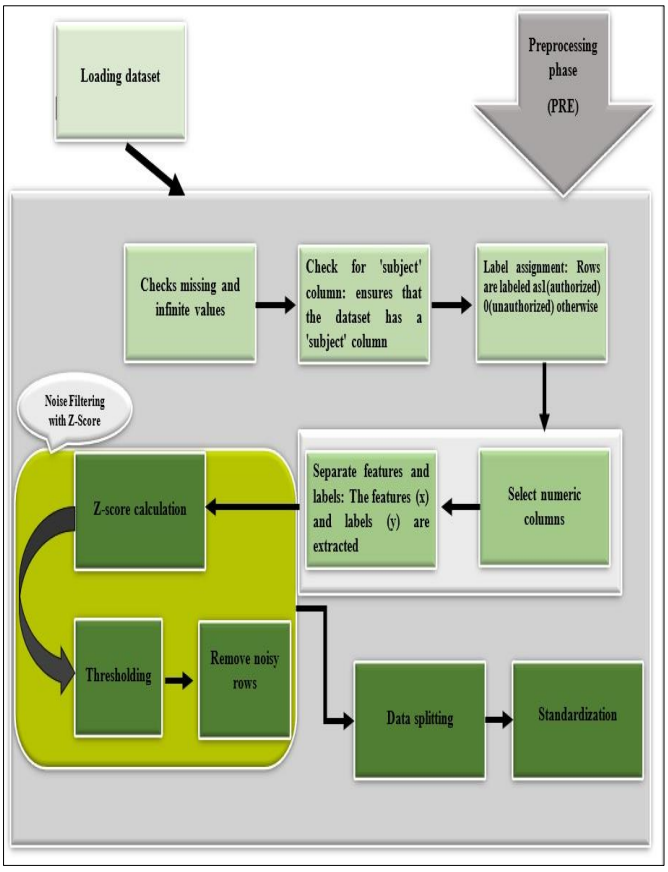

Figure 5. Proposed Preprocessing Phase

For more illustration of preprocessing phase, see Algorithms (1) and (2).

Algorithm 1: Proposed Z _ score Calculation

Input: CSV file of CMU Keystroke Dynamic Benchmark Dataset (CMU-KDB) without missing or infinite values.

Output: CSV file of CMU Keystroke Dynamic Benchmark Dataset (CMU-KDB) without outliers.

Step1: Calculate the mean (μ) and standard deviation (σ) for each numeric feature.

for each feature in dataset:

μ= mean of feature

$$\mu = \frac{1}{N}\sum_{i=1}^{N} X_i$$

Where, N is the number of samples and X_i is the feature value.

σ = standard deviation of feature

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2}$$

Step2: Apply the z-score formula to each feature value

$$Z_i = \frac{X_i - \mu}{\sigma}$$

Step3: Identify noisy data

For each feature value in dataset

if Z_i > threshold (2.5)

consider row i as outliers

Step 4: Remove outlier rows from the dataset

For each row i that is consider outliers

Remove row i from dataset.

Step5: Return the output.

END

Algorithm 2: Proposed Preprocessing (PRE)

Input: (CMU-KDB)-Before CSV file of CMU Keystroke Dynamic Benchmark Dataset before preprocessing.

Output: (CMU-KDB)-After- CSV file of CMU Keystroke Dynamic Benchmark Dataset after preprocessing.

Step1: Clean data check if there are missing values, put " NaN " word in this cell according to the following formula:

$$M(j) = \sum_{i=1}^{N} 1\left(x_{i,j} = NaN\right)$$

Where:

M(j)is the count of missing values in column (j).

$x_{i,j}$ is the value at row (i) and column (j).

$1(x)$ is an indicator function that returns 1 if x is NaN, otherwise 0.

N is the total number of rows.

Step2: check if there are infinite values in this cell according to the following formula:

$$I(j) = \sum_{i=1}^{N} 1(x_{i,j} = \pm\infty)$$

Where:

$I(j)$counts the number of infinite values in column (j).

The indicator function 1(x)returns 1 if x=+ ∞ or x= - ∞ , otherwise 0.

Step3: Check for 'subject' column (If missing, raise error or exit).

Step4: Assign labels to the 'subject' column

Authorized subjects -> label 1
Unauthorized subjects -> label 0

Step 5: Select only numeric columns (X).

Step6: Separate features (X) and labels (Y)

X = Numeric features
Y= Labels.

Step7: Noise filtering using Z-score (apply algorithm x)

Step8: Split data into training and testing sets

X_train, X _test
Y _train, Y _test.

Step9: Standardize features using StandardScaler Fit scaler on training data and apply it to both training and testing data according to the following formula:

$$\boldsymbol{X_{(scaled)}} = \frac{\boldsymbol{x} - \boldsymbol{\mu}}{\boldsymbol{\sigma}}$$

Where:

X is the original feature value.
μ (mean) is the average of all values in the feature.
σ (standard deviation) measures the spread of the values.
$X_{(scaled)}$is the transformed value after standardization.

END

7.2 Feature Engineering and Feature Selection Phases

Each sample in the data set is represented by a series of timing information that expresses the exact time when keys were pressed and released. From the timing information, many types of features can be extracted. As mentioned previously, the CMU Keystroke Dynamics Benchmark dataset by default consists of three features only, the first one is a unigraph feature: Hold(H) with two diagraph features: Down-Down(D-D), and Up-Down (U-D). Finding the important properties in the raw data is critical for reducing the classification error. In order to address the issue of creating features from attributes, an effective feature extraction approach was developed; outstanding classification performance is presently being pursued.

The main function of feature engineering phase is to extract new features from the original features founded in standard dataset. The feature engineering is surely the crucial means in terms of keystroke dynamics-based biometrics, and it affects the performance of the SVM-KD. Figure (6) illustrates the representation of all original features included (H, D-D, and U-D) consequently for the used password (.tie5Roanl).

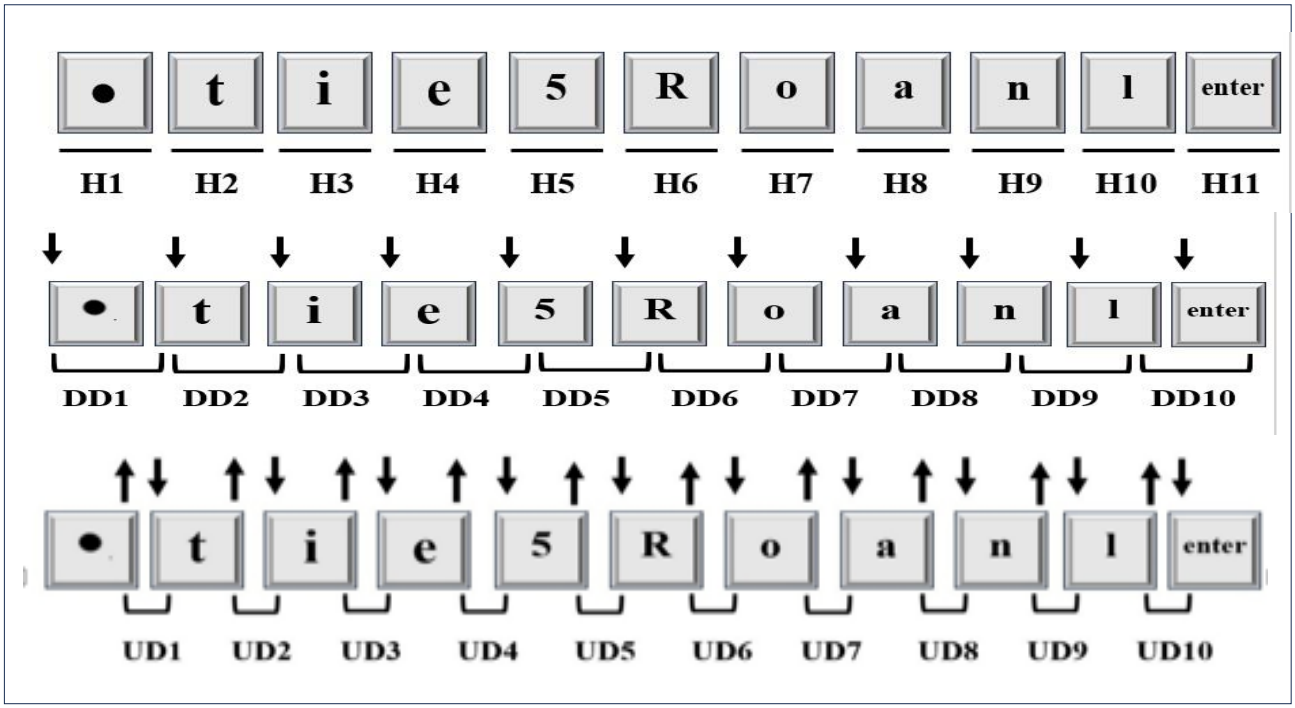

Figure 6. Representation of Original Features: (a) H feature, (b) D-D feature, and (c) U-D feature

In a research three new features are generated, these features include two diagraph features: Up-Up (U-U), Down-Up (D-U) in addition to a trigraph feature: Trigraph-Hold (T-H). The calculation for each new keystroke feature is formulated as below:

a. U-U feature: $U\text{-}D_{1\text{-}2}+H_2$

Where

$U\text{-}D_{1\text{-}2}$: is the time from releasing the first key to pressing the second key.

H_2: is the time of pressing the second key.

See Figure (7) depicts the representation of U-U generated feature for used password (.tie5Roanl).

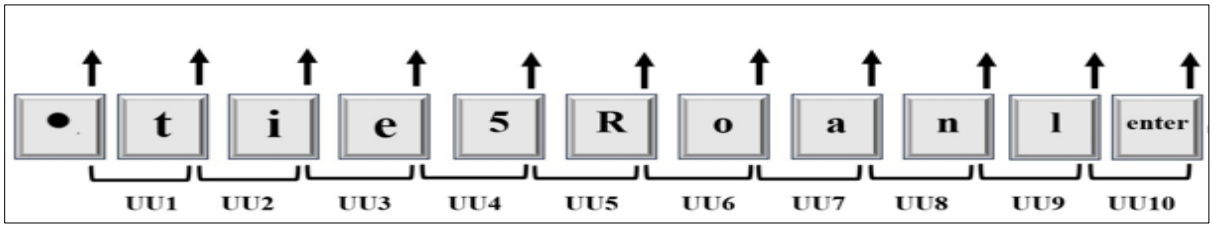

Figure 7. Representation of U-U Feature

b- D-U feature: $H_1 + U\text{-}D_{1\text{-}2}+H_2$

Where

H_1: is the time of pressing the first key.

$U\text{-}D_{1\text{-}2}$: is the time from releasing the first key to pressing the second key.

H_2: is the time of pressing the second key.

See Figure (8) depicts the representation of D-U generated feature for used password (.tie5Roanl).

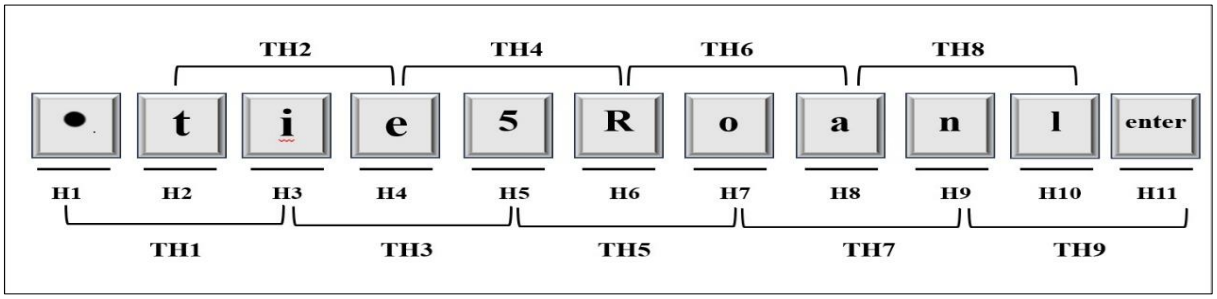

Figure 8. Representation of D-U Feature

c- T-H feature: $H_1 + H_2 + H_3$

Where

H_1: is the time of pressing the first key.

H_2: is the time of pressing the second key.

H_3: is the time of pressing the third key.

See Figure (9) depicts the representation of T-H generated feature for used password (.tie5Roanl).

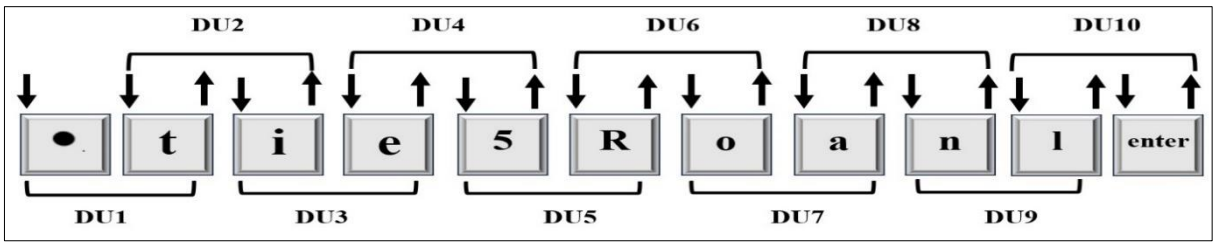

Figure 9. Representation of T-H Feature

After generation new three features, the original dataset updated three times. For more illustration, Figure (10) clarify the contents of new three datasets after generated three new features, in addition to the original dataset, resulted in having four basic datasets (O-D: Original Dataset, U-UD: the first new dataset after generation the first new feature U-U, D-UD: the second new dataset after generation the second new feature D-U, and T-HD: the third new dataset after generation the third new feature T-H).

	A	B	C	D	E	F	G	H
1	subject	sessionInd	rep	H.period	DD.period.t	UD.period.t	DU.period.t	H.t
2	s002	1	1	0.1491	0.3979	0.2488	0.5048	0.1069
3	s002	1	2	0.1111	0.3451	0.234	0.4145	0.0694
4	s002	1	3	0.1328	0.2072	0.0744	0.2803	0.0731
5	s002	1	4	0.1291	0.2515	0.1224	0.3574	0.1059
6	s002	1	5	0.1249	0.2317	0.1068	0.3212	0.0895
7	s002	1	6	0.1394	0.2343	0.0949	0.3156	0.0813
8	s002	1	7	0.1064	0.2069	0.1005	0.2935	0.0866
9	s002	1	8	0.0929	0.181	0.0881	0.2628	0.0818
10	s002	1	9	0.0966	0.1797	0.0831	0.2568	0.0771
11	s002	1	10	0.1093	0.1807	0.0714	0.2538	0.0731
12	s002	1	11	0.0887	0.166	0.0773	0.2536	0.0876
13	s002	1	12	0.0911	0.1525	0.0614	0.2349	0.0824
14	s002	1	13	0.1114	0.162	0.0506	0.252	0.09
15	s002	1	14	0.0903	0.1871	0.0968	0.2676	0.0805
16	s002	1	15	0.1169	0.2562	0.1393	0.3301	0.0739
17	s002	1	16	0.127	0.1839	0.0569	0.275	0.0911
18	s002	1	17	0.1016	0.1799	0.0783	0.2591	0.0792
19	s002	1	18	0.1056	0.1755	0.0699	0.2536	0.0781
20	s002	1	19	0.1177	0.2237	0.106	0.3074	0.0837
21	s002	1	20	0.1027	0.1781	0.0754	0.251	0.0729
22	s002	1	21	0.1016	0.1374	0.0358	0.2235	0.0861
23	s002	1	22	0.1072	0.2217	0.1145	0.2943	0.0726
24	s002	1	23	0.1243	0.1841	0.0598	0.2609	0.0768
25	s002	1	24	0.1241	0.2019	0.0778	0.2848	0.0829

(a) U-UD

	A	B	C	D	E	F	G	H	I	J	K
1	subject	sessionInd	rep	H.period	DD.period.t	UD.period.t	H.t	DD.t.i	UD.t.i	H.i	H.period.t.i
2	s002	1	1	0.1491	0.3979	0.2488	0.1069	0.1674	0.0605	0.1169	0.3729
3	s002	1	2	0.1111	0.3451	0.234	0.0694	0.1283	0.0589	0.0908	0.2713
4	s002	1	3	0.1328	0.2072	0.0744	0.0731	0.1291	0.056	0.0821	0.288
5	s002	1	4	0.1291	0.2515	0.1224	0.1059	0.2495	0.1436	0.104	0.339
6	s002	1	5	0.1249	0.2317	0.1068	0.0895	0.1676	0.0781	0.0903	0.3047
7	s002	1	6	0.1394	0.2343	0.0949	0.0813	0.1299	0.0486	0.0744	0.2951
8	s002	1	7	0.1064	0.2069	0.1005	0.0866	0.1368	0.0502	0.08	0.273
9	s002	1	8	0.0929	0.181	0.0881	0.0818	0.1378	0.056	0.0747	0.2494
10	s002	1	9	0.0966	0.1797	0.0831	0.0771	0.1296	0.0525	0.0839	0.2576
11	s002	1	10	0.1093	0.1807	0.0714	0.0731	0.1457	0.0726	0.0766	0.259
12	s002	1	11	0.0887	0.166	0.0773	0.0876	0.156	0.0684	0.0839	0.2602
13	s002	1	12	0.0911	0.1525	0.0614	0.0824	0.1516	0.0692	0.0731	0.2466
14	s002	1	13	0.1114	0.162	0.0506	0.09	0.1547	0.0647	0.0797	0.2811
15	s002	1	14	0.0903	0.1871	0.0968	0.0805	0.1919	0.1114	0.0842	0.255
16	s002	1	15	0.1169	0.2562	0.1393	0.0739	0.1549	0.081	0.0892	0.28
17	s002	1	16	0.127	0.1839	0.0569	0.0911	0.1381	0.047	0.0895	0.3076
18	s002	1	17	0.1016	0.1799	0.0783	0.0792	0.1434	0.0642	0.076	0.2568
19	s002	1	18	0.1056	0.1755	0.0699	0.0781	0.1391	0.061	0.0898	0.2735
20	s002	1	19	0.1177	0.2237	0.106	0.0837	0.188	0.1043	0.0919	0.2933
21	s002	1	20	0.1027	0.1781	0.0754	0.0729	0.1418	0.0689	0.0792	0.2548
22	s002	1	21	0.1016	0.1374	0.0358	0.0861	0.1629	0.0768	0.0774	0.2651
23	s002	1	22	0.1072	0.2217	0.1145	0.0726	0.1349	0.0623	0.0768	0.2566
24	s002	1	23	0.1243	0.1841	0.0598	0.0768	0.1568	0.08	0.085	0.2861
25	s002	1	24	0.1241	0.2019	0.0778	0.0829	0.1745	0.0916	0.0734	0.2804

(b) O-D

	A	B	C	D	E	F
1	subject	sessionInd	rep	H.period	DD.period.t	UD.period.t
2	s002	1	1	0.1491	0.3979	0.2488
3	s002	1	2	0.1111	0.3451	0.234
4	s002	1	3	0.1328	0.2072	0.0744
5	s002	1	4	0.1291	0.2515	0.1224
6	s002	1	5	0.1249	0.2317	0.1068
7	s002	1	6	0.1394	0.2343	0.0949
8	s002	1	7	0.1064	0.2069	0.1005
9	s002	1	8	0.0929	0.181	0.0881
10	s002	1	9	0.0966	0.1797	0.0831
11	s002	1	10	0.1093	0.1807	0.0714
12	s002	1	11	0.0887	0.166	0.0773
13	s002	1	12	0.0911	0.1525	0.0614
14	s002	1	13	0.1114	0.162	0.0506
15	s002	1	14	0.0903	0.1871	0.0968
16	s002	1	15	0.1169	0.2562	0.1393
17	s002	1	16	0.127	0.1839	0.0569
18	s002	1	17	0.1016	0.1799	0.0783
19	s002	1	18	0.1056	0.1755	0.0699
20	s002	1	19	0.1177	0.2237	0.106
21	s002	1	20	0.1027	0.1781	0.0754
22	s002	1	21	0.1016	0.1374	0.0358
23	s002	1	22	0.1072	0.2217	0.1145

(d) T-HD

	A	B	C	D	E	F	G	H
1	subject	sessionInd	rep	H.period	DD.period.t	UD.period.t	UU.period.t	H.t
2	s002	1	1	0.1491	0.3979	0.2488	0.3557	0.1069
3	s002	1	2	0.1111	0.3451	0.234	0.3034	0.0694
4	s002	1	3	0.1328	0.2072	0.0744	0.1475	0.0731
5	s002	1	4	0.1291	0.2515	0.1224	0.2283	0.1059
6	s002	1	5	0.1249	0.2317	0.1068	0.1963	0.0895
7	s002	1	6	0.1394	0.2343	0.0949	0.1762	0.0813
8	s002	1	7	0.1064	0.2069	0.1005	0.1871	0.0866
9	s002	1	8	0.0929	0.181	0.0881	0.1699	0.0818
10	s002	1	9	0.0966	0.1797	0.0831	0.1602	0.0771
11	s002	1	10	0.1093	0.1807	0.0714	0.1445	0.0731
12	s002	1	11	0.0887	0.166	0.0773	0.1649	0.0876
13	s002	1	12	0.0911	0.1525	0.0614	0.1438	0.0824
14	s002	1	13	0.1114	0.162	0.0506	0.1406	0.09
15	s002	1	14	0.0903	0.1871	0.0968	0.1773	0.0805
16	s002	1	15	0.1169	0.2562	0.1393	0.2132	0.0739
17	s002	1	16	0.127	0.1839	0.0569	0.148	0.0911
18	s002	1	17	0.1016	0.1799	0.0783	0.1575	0.0792
19	s002	1	18	0.1056	0.1755	0.0699	0.148	0.0781
20	s002	1	19	0.1177	0.2237	0.106	0.1897	0.0837
21	s002	1	20	0.1027	0.1781	0.0754	0.1483	0.0729
22	s002	1	21	0.1016	0.1374	0.0358	0.1219	0.0861
23	s002	1	22	0.1072	0.2217	0.1145	0.1871	0.0726
24	s002	1	23	0.1243	0.1841	0.0598	0.1366	0.0768
25	s002	1	24	0.1241	0.2019	0.0778	0.1607	0.0829
26	s002	1	25	0.1098	0.1567	0.0469	0.1237	0.0768

(c) D-UD

Figure 10. Contents of Four Different Datasets

The length of the timing data the vector varies depending on the length of the password., such that a password (.tie5Roanl) which contains ten characters in addition to the enter key will result in eleven H, ten for each one of D-D, U-D, U-U, and D-U, in addition to nine T-H. Generally, a password with n character will yield n number of H feature and n - 1 number for each one of the remaining features. The representation of keystroke features for user 'X' is demonstrated in Table (2). For more illustration, for each user's trail, the user has 6*n-6 features values, i.e. (6*11-6=> 66-6=> 60 values), that produced for 400 trails for each user (400*60=24000 values), by the same way, the total number of features values for 51 users equal to (24000*51= 1224000 values).

Table 2: Representation of Keystroke Features for User 'X'

	Password Times (H-T, U-DT, D-DT, UUT, D-UT, D-UT)								
Reptation no.	T_1	T_2	T_3	T_4	T5	T6	T7	…	T_{6n-6}
1	H-T1	U-DT1	D-DT1	U-UT1	D-UT1	T-HT1	H-T1	…	T-HT1
2	H-T2	U-DT2	D-DT2	U-UT2	D-UT2	T-HT2	H-T2		T-HT2
….								…..	
400	H-T400	U-DT400	D-DT400	U-UT400	D-UT400	T-HT400	H-T400	…	T-HT400

7.3 SVM Training and Testing Phases

This research adopts SVM deep learning algorithm, this section explains how to utilize this algorithm for keystroke dynamics authentication. To train the SVM algorithm, a set of timing vectors from each user class is required. These timing vectors are collected for each user and stored in CMU Keystroke Dynamics Benchmark Dataset as mentioned in section 4. The timing vectors are preprocessed to form a set of patterns that will be used to train the network. In SVM_KD, the structure of SVM depends on the extracted feature(s) and the used password. See Figure (11).

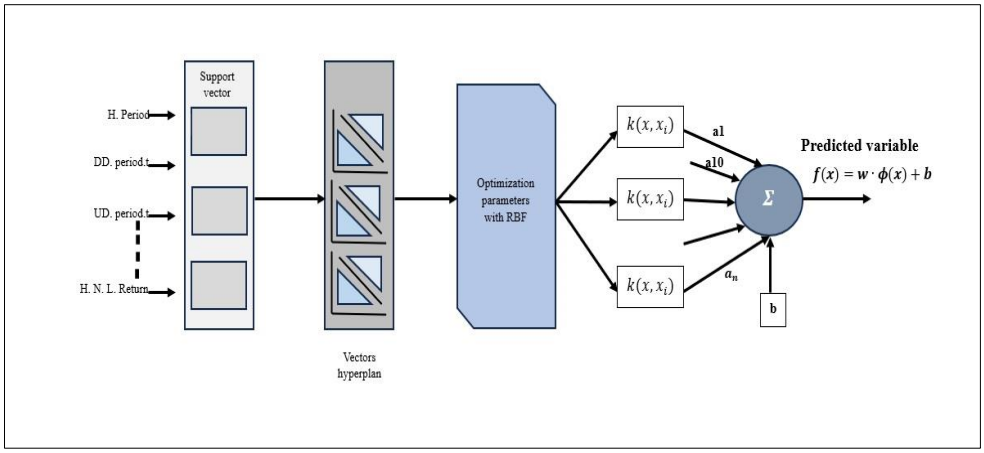

Figure 11. SVM Structure of SVM-KD

Algorithm (3) demonstrates the steps of SVM training and testing for SVM_KD.

Algorithm 3: SVM Training and Testing
Input: (CMU-KDB)-After- CSV file of CMU Keystroke Dynamic Benchmark Dataset after preprocessing
Output: Probabilities for each class
Step1: Preprocess Data Standardize features using StandardScaler Fit scaler on training data and apply it to both training and testing data according to the following formula: $$\boldsymbol{X_{(scaled)} = \frac{x - \mu}{\sigma}}$$ Where: X is the original feature value. μ (mean) is the average of all values in the feature. σ (standard deviation) measures the spread of the values. $X_{(scaled)}$is the transformed value after standardization.
Step2: Split the dataset into training and testing sets 80% training & 20% testing.
Step3: Choose Kernel Function (rbf) $$K_{(x_i, x_j)} = \exp(-\gamma \parallel x_i - x_j \parallel^2)$$ Where: x_i, x_j Input vectors (feature points). $\parallel x_i - x_j \parallel^2$ Squared Euclidean distance between the two vectors. γ (gamma) A parameter that controls how far the influence of a single training example reaches. exp, The exponential function.

Step4: Define the Optimization Problem

$$\min_{w,b,\xi} \frac{1}{2} \| w \|^2 + C \sum_i \xi_i$$

Subject to:

$$Y_i(w \cdot x_i + b) \geq 1 - \xi_i , \xi_i \geq 0$$

Where:

w Weight vector (defines hyperplane orientation).
b Bias (defines hyperplane position).
$\| w \|^2$ Squared norm of weights — we want to minimize this to maximize the margin.
ξ_i Slack variables — measure how much a point violates the margin.

C Regularization parameter — controls the trade-off between maximizing margin and minimizing misclassification.

Step5: Use Quadratic Programming to Solve It

Solve the dual optimization problem using Lagrange multipliers.
Get optimal values of α_i which indicate support vectors.

Step6: Construct the Hyperplane

Compute:

$$w = \sum_i \alpha_i y_i x_i$$

Compute:

$$b = y_k - w \cdot x_k$$

Where

α_i the importance (Lagrange multiplier).
y_i the class label.
x_i the feature vector

Step7: Make Predictions

Given a new sample x, compute the decision function:

$$f_{(x)} = sign(w \cdot x + b)$$

In kernel

$$f_{(x)} = sign\left(\sum_i \alpha_i y_i K(x_i , x) + b\right)$$

END

The training phase of SVM aims to discover the optimal value of the Gamma (γ) parameter. In this research, multiple values of γ were generated based on the feature(s) employed, as detailed in Algorithm (4), to achieve the lowest rate mistakes in classification. This value of γ is regarded the ideal value for good classification; nevertheless, γ values differ depending on the features utilized. See Table (3).

Algorithm 4: Calculate Gamma(γ) for SVM (Scale Mode)
Input: dataset with numeric features (rows: samples, columns: features)
Step-1: n-features number of columns in(x), for each feature column in(x) , compute variance mean-variance: average of all feature variances
Step-2: $$\boldsymbol{Gamma} = \frac{\mathbf{1}}{\mathbf{n-features} * \mathbf{mean-variance}}$$ $$\mathbf{mean-variance} = \frac{\mathbf{1}}{\mathbf{n-features}} \sum_{j=1}^{\mathbf{n-features}} \boldsymbol{\sigma_j^2}$$
Step-3: Return gamma
END

Table 3: γ Values with Different Models and Cases

Model Name	Gamma (γ) without z-score	Gamma (γ) with threshold 2.5
O	0.00464868	0.00478737
O+2	0.00466649	0.00479453
O+U-U	0.00464971	0.00478988
O+D-U	0.00465804	0.00479330
O+3	0.00465813	0.00479239
O+2+3	0.00466545	0.00479184
2+3	0.00466566	0.00478484

7.4 Data Set Splitting Strategy

In this research, two dataset splitting strategies are used with training and testing phases as follows:

1- Strategy-1

In this strategy suppose only the first user in Class-1(authorized class) and the rest of 50 users are in Class-2 (un-authorized class), resulted in number of rows for Class-1 are equal to 1*400 and the number of rows for Class-2 are equal to 50*400. For more illustration, see Figure (12).

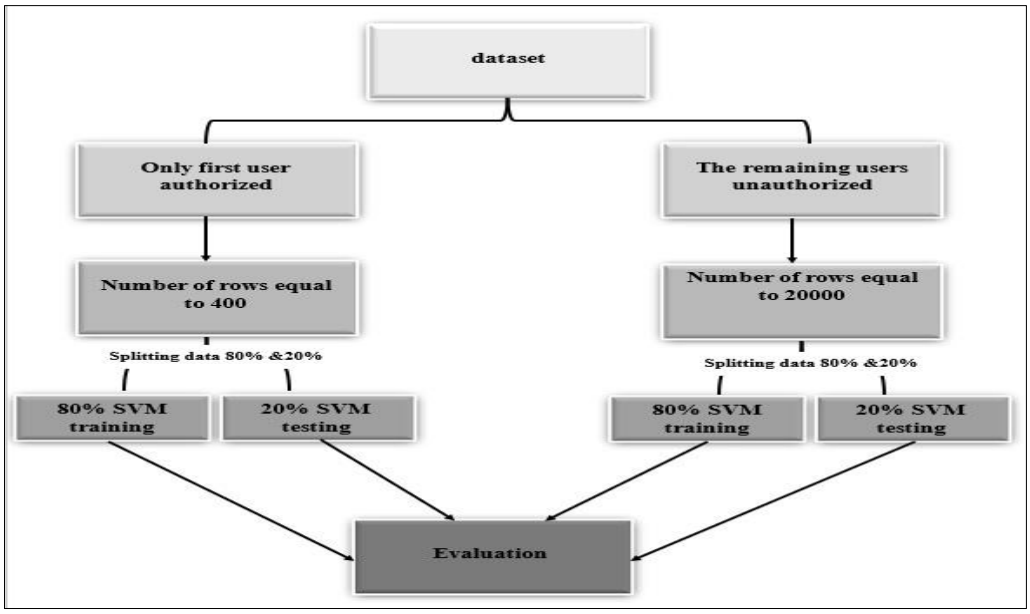

Figure 12. Data Set Splitting Strategy-1

1- Strategy-2

In this strategy suppose only the first three users are in Class-1(authorized class) and the rest of 48 users are in Class-2 (un-authorized class), resulted in number of rows for Class-1 are equal to 3*400 and the number of rows for Class-2 are equal to 48*400. For more illustration, see Figure (13).

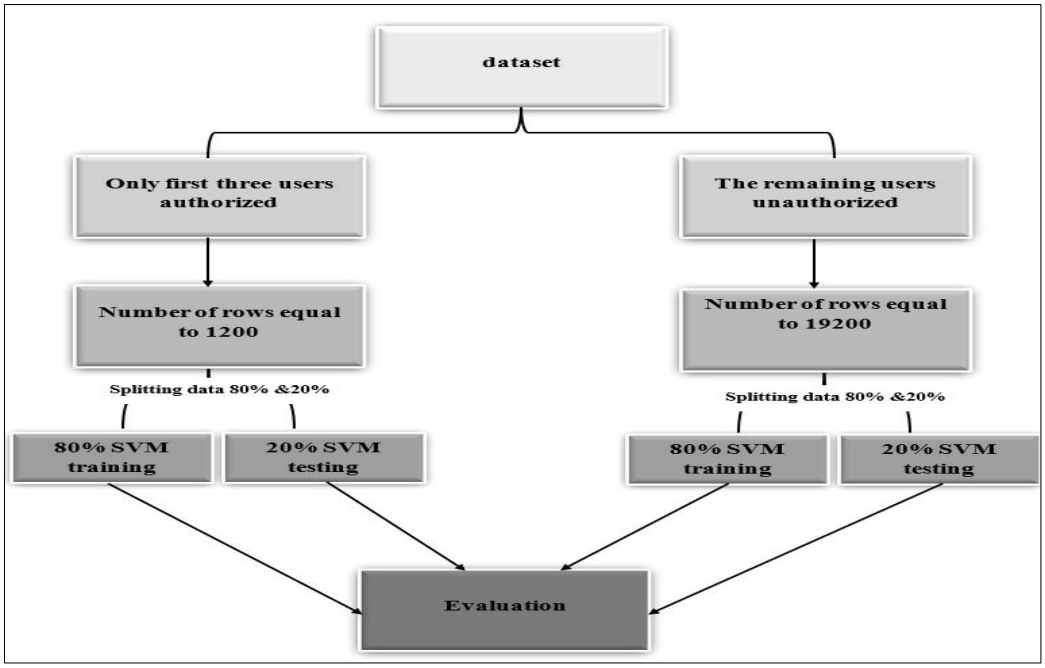

Figure 13. Data Set Splitting Strategy-2

To balance the training and testing samples in two classes, for each one of the above strategies produced two datasets, one for authorized users and another one for un-authorized users, the next step is divided each class samples in to two parts, 80% for training and 20% for testing. So, the 80% represents 320 vectors were randomly chosen from one selected user and the same number of vectors was chosen from the rest of users, in addition to the 20% represents 180 vectors were randomly chosen from one selected user and the same number of vectors was chosen from the rest of users. In this research, the proposed system consists of seven models for keystroke dynamics authentication, these models implemented according to different combination of used features as illustrated in Table (4).

Table 4: SVM-KD Proposed Models

Model no.	Model Name	Description of Dataset contents
1	*O*	Only the original dataset features (H, U-D, D-D)
2	*O+2*	Original dataset features + new generated diagraph features (H, D -U, D-D, U-D, U-U)
3	*O+U-U*	Original dataset features + new generated U-U feature (H, U-D, D-D, U-U)
4	*O+D-U*	Original dataset features + new generated D-U feature (H, D -U, D-D, D-U)
5	*O+3*	Original dataset features +new generated trigraph feature ((H, U-D, D-D, TH))
6	*O+2+3*	Original dataset features+ new generated diagraph features+ new generated trigraph (H, D -U, D-D, U-D, U-U, TH)
7	*2+3*	New generated diagraph features+ new generated trigraph (D-U, U-U, TH)

8. Results and Discussion

This section experimentally evaluates the performance of proposed SVM-KD models that explained in the previous section. Several experiments are conducted to show the applicability of proposed SVM-KD models when different combinations of features are used. This research used the Python programming language, several libraries, and machine learning frameworks and packages. These included NumPy and pandas, which provide numerical computing tools and data manipulation, run on the CPU environment (Intel Core i7-5600U CPU @ 2.60GHz with 6.91 GB of RAM and a 168.13 GB disk) to implement data analysis.
Each user has a distinguishing typing style because the extracted features for each user is different from the other. In other words, keystroke patterns exhibit a degree of variance between samples. Figure (14) depicts the values of H-T feature for random ten users, each user types 50 times in 8 sessions. The x axis refers to nine H-T values obtained from typing password (.tie5Roanl); where the y axis refers to time (MS). The red points refer to average of feature values for 400 trails.

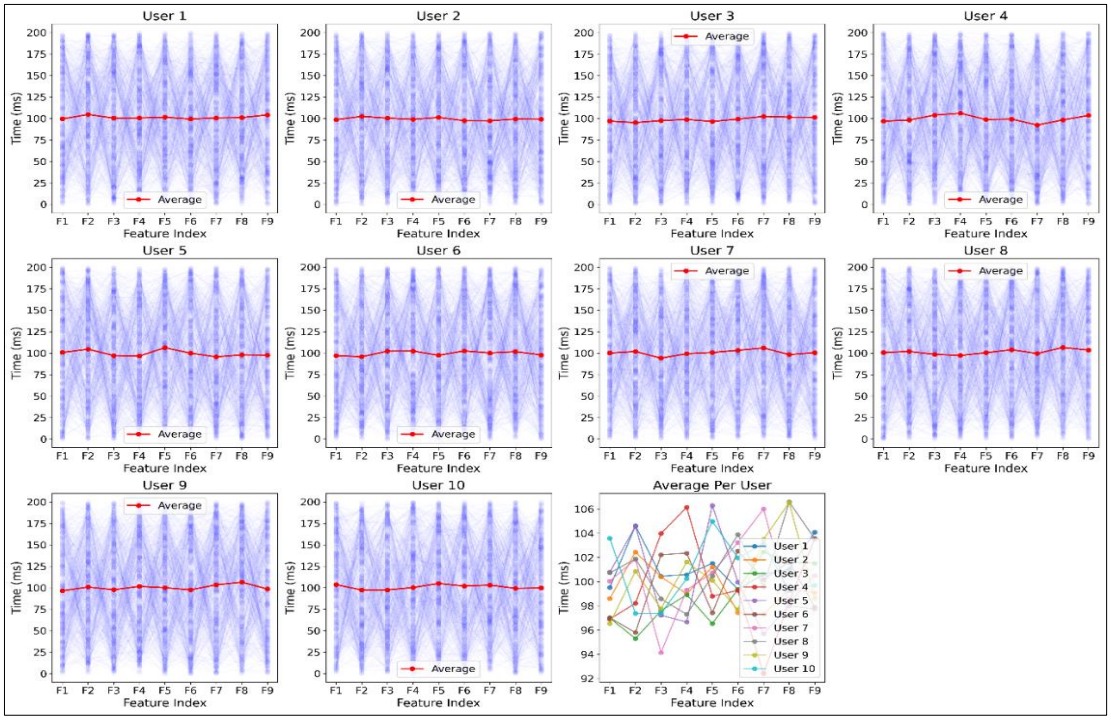

Figure 14. Varity of T-H Feature for Random Ten Users

The evaluating predictive performance of SVM-KD as biometric authentication is calculated by six measures called F1-score (F1), Accuracy (Acc), Equal Error Rate (EER), Recall, Precision, and ROC according the equations mentioned in section 5. This research implemented several experiments for keystroke dynamics are conducted upon the CMU Keystroke Dynamics Benchmark dataset, for each experiment divided in to two cases according to two dataset splitting strategies as mentioned previously in sub section 7.4, then for each case contains two scenarios without z-score and with 2.5 z-score threshold. Figure (15) described the general structure for each experiment.

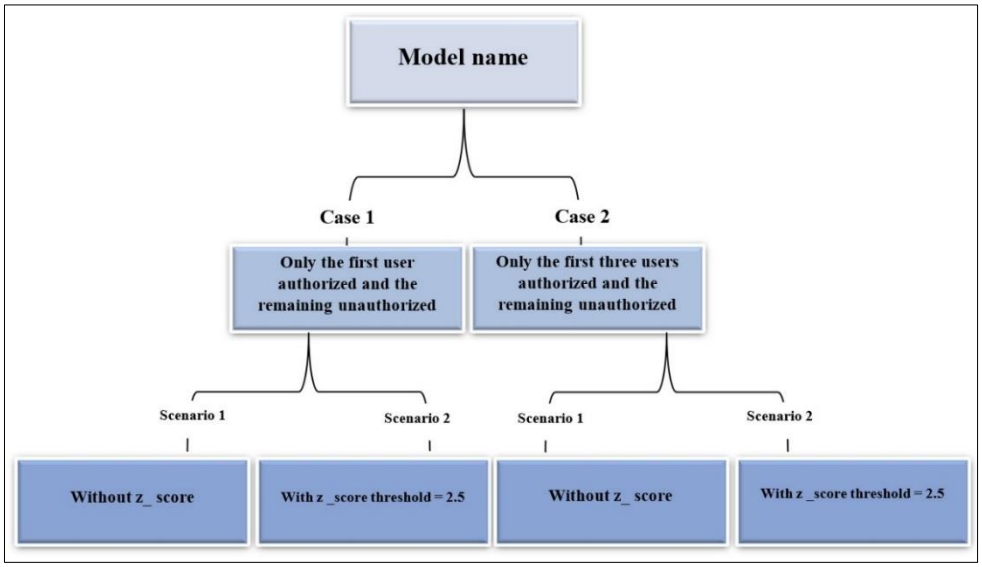

Figure 15. General Structure for each Experiment

In this section discuss the results for all experiments, such that, the number of all experiments implemented in this research as follows: -

$$Total\ Exp. = no.ofmodels * no.of\ cases * no.of\ scenarios$$

$$Total\ Exp. = 7 * 2 * 2 \ \ldots\ldots\ldots\ldots\ldots\ldots > Total\ Exp. = 28\ Experiments$$

Such that in **Experiment-1**, only the original features founded in default dataset (O) are used, these features included (H, U-D, and D-D). This experiment included the following two cases: -

Case-1: this case represented by applying the first strategy of dataset splitting strategies as mentioned previously in sub section 7.4, so, in this strategy suppose only the first user is in Class-1(authorized class) and the rest of 50 users are in Class-2 (un-authorized class). This case applied with each one of the following two scenarios: -

- **Scenario-1**: this scenario applied without using Z-score.
- **Scenario-2**: this scenario applied with 2.5 Z-score threshold.

Case-2: this case represented by applying the second strategy of dataset splitting strategies as mentioned previously in sub section 7.4, so, in this strategy suppose only the first three users are in Class-1(authorized class) and the rest of 48 users are in Class-2 (un-authorized class). This case applied with each one of the following two scenarios: -

- **Scenario-1**: this scenario applied without using Z-score.
- **Scenario-2**: this scenario applied with 2.5 Z-score threshold.

With the same way, all the remaining experiments can be represented according the models' details explained previously in Table (4)**.**

The results for 28 experiments illustrated in Table (5), this table showed that, the results for Experiment-1 are good results but are the lowest comparing with the remaining experiments. In addition, the results indicated to robust and efficient effects of the proposed preprocessing steps, data splitting strategies, and feature engineering.

Table 5: Results for all Experiments

Dataset	No. of Users	Z-score Threshold	F1	Acc	EER	Recall	Precision	ROC
O	1	Without	87.96	99.31	7.58	81.24	98.58	97.23
		2.5	88.83	99.07	6.91	83.72	95.89	98.11
	3	Without	90.52	98.33	4.81	85.88	96.77	98.90
		2.5	91.32	98.28	5.64	88.37	94.84	98.95
O+2	1	Without	80.94	99.19	10.38	73.57	95.48	96.62
		2.5	91.08	99.23	3.61	85.60	98.70	98.80
	3	Without	89.30	98.21	4.97	84.45	95.99	98.72
		2.5	93.48	98.40	5.46	91.10	96.19	99.13
O+U-U	1	Without	85.19	99.19	9.33	77.77	98.38	96.83
		2.5	90.57	99.20	4.11	86.23	96.15	99.56
	3	Without	89.96	98.23	4.81	85.38	96.12	98.78
		2.5	92.08	98.40	5.10	89.32	95.32	99.37
O+D-U	1	Without	85.19	98.99	9.41	77.77	98.38	96.90
		2.5	91.03	99.26	4.11	86.92	96.23	98.74
	3	Without	89.85	98.13	4.47	85.37	95.83	98.80
		2.5	92.48	98.67	4.89	91.78	93.20	99.15
O+3	1	Without	88.68	99.33	7.46	82.61	97.65	97.35
		2.5	88.97	99.10	8.80	83.07	97.62	98.25
	3	Without	89.75	98.21	5.30	84.93	96.35	98.70
		2.5	91.94	98.40	5.62	88.50	96.15	98.97
O+2+3	1	Without	84.88	99.16	10.90	77.75	97.22	96.80
		2.5	91.55	99.30	2.05	86.28	98.73	98.75
	3	Without	89.35	98.13	5.17	84.67	95.73	98.73
		2.5	94.20	98.80	4.46	91.81	96.91	99.35
2+3	1	Without	76.71	98.87	10.28	68.74	97.66	97.14
		2.5	92.06	99.35	2.93	87.00	98.82	99.00
	3	Without	86.16	97.67	5.61	80.67	94.48	98.44
		2.5	92.06	98.36	4.81	89.26	95.36	99.26

For discussing and analyzing the obtained results of all implemented experiments, several important sides must be displayed as follows: -

1- Preprocessing

The proposed preprocessing steps reflects very good results specially by using Z-score concept, so the results obtained by using Z-score with threshold=2.5 are higher than without Z-score for all cases.

2- Feature engineering

The proposed methods of feature engineering in this research produced the highest results, such that the best results obtained from the proposed model- (O+2+3), that used the combination of original dataset features in addition to new generated diagraph features and new generated trigraph (H, D -U, D-D, U-D, U-U, TH). These results give the indication of enhancement the efficiency of keystroke dynamics system that depended on the original dataset only, as illustrated in Table (V). Several combinations of features applied, some of these implemented by using only the new generated features and others implemented by using the original dataset features in addition to new generated features in different cases, for all applied combinations, the results are better than used the original dataset only.

3- dataset splitting strategies

In proposed models using two strategies for dataset splitting, the results obtained for all experiments indicated that SVM produced the best results when considered the trails for the first three users as Class-1(authorized users) and all the remaining trails for Class-2(un-authorized users). In addition, the training and testing time for all proposed model by using Z-score is less than other models, that means by using Z-score concept reduce the training and testing time and enhance the efficiency of SVM-KD. See Table (6).

Table 6: Training and Testing Times for all Experiments of SVM-KD

Model	**No. of Users (Cases)**	**Z-score Th. (Scenarios)**	**Training Time\s**	**Testing Time\s**
O	1	without	0.6651seconds	0.3280
		2.5	0.3170seconds	0.1779
	3	without	1.3851seconds	0.7110
		2.5	0.6780seconds	0.3530
O+2	1	without	0.7860seconds	0.4020
		2.5	0.4050seconds	0.1950
	3	without	1.7061seconds	0.8240
		2.5	0.8370seconds	0.4180
O+U-U	1	without	0.7971seconds	0.3660
		2.5	0.3920seconds	0.1809
	3	without	1.7001seconds	0.8200
		2.5	0.8340seconds	0.4220
O+D-U	1	without	0.7941seconds	0.3759
		2.5	0.4170seconds	0.1789
	3	without	1.7101seconds	0.8060
		2.5	0.8430seconds	0.3850

O+3	1	without	0.7790seconds	0.3700
		2.5	0.3800seconds	0.1709
	3	without	1.5901seconds	0.7780
		2.5	0.7730seconds	0.3459
O+2+3	1	without	0.8920seconds	0.4309
		2.5	0.4590seconds	0.2069
	3	without	1.8881seconds	0.8590
		2.5	0.9310seconds	0.4090
2+3	1	without	0.7540seconds	0.3699
		2.5	0.4720seconds	0.2379
	3	without	1.6210seconds	0.8030
		2.5	0.9390seconds	0.4760

9. Conclusion

In this section, from the results of this research several points can be concluded, the first point is related to preprocessing phase. This phase produced good effects on the results comparing with the results without preprocessing. In addition to the feature-engineering phase generated good new keystroke dynamics features namely U-D, U-U, and H-T that strength the default features, which founded in original data set, therefore, these additional features, could be used to strengthen password security by differentiating between authorized and unauthorized users. Applying SVM classifiers with various data splitting techniques for training and testing success with keyboard dynamics for authentication fields is an example of deep learning. The provided research's accuracy with various suggested models is nearly at the acceptable error levels needed for a system with a certain level of security. Data splitting strategy for training and testing is an important issue in user authentication based on keystroke dynamics. The results of second strategy of splitting data when consider the first three users as authorized in all experiments are better than results obtained from first strategy.

Conflicts of Interest: The authors declare that there is no conflict of interest.

Author Contributions: Conceptualization Mays M. Hoobi; methodology Rasha Khalid Ibrahim; analysis Mays M. Hoobi, and writing original draft preparation Rasha Khalid Ibrahim, Mays M. Hoobi; validation and reviewing the manuscript by both authors, who then approved the final version of the manuscript.

Acknowledgments: The authors would like to thank Baghdad University in Baghdad, Iraq, for their operation with (http://uobaghdad.edu.iq).

References

[1] O. G. Olufemi and R. D. Alimi, "Authenticating Device Users via Keyboard Strokes," *International Journal of Computer Applications*, vol. 175, no. 17, p. 8887, 2020.

[2] N. Ali, M. M. Hoobi and D. F. Saffo, "Development of Robust and Efficient Symmetric Random Keys Model based on the Latin Square Matrix," *Mesopotamian Journal of Cybersecurity*, vol. 4, no. 3, pp. 203-215, 2024.

[3] A. J. Smith and B. L. Johnson, "A Comprehensive Review of Cryptographic Techniques for Secure Communication," *Journal of Information Security and Applications*, vol. 65, pp. 102-114, 2023. doi: 10.1016/j.jisa.2023.102114.

[4] M. M. Hoobi, "Enhanced rail-fence cryptography algorithm using hybrid models," in *AIP*, Erbil, Iraq, 2025.

[5] R. M. Al-Amri, D. N. Hamood and A. K. Farhan, "Theoretical background of cryptography," *Mesopotamian Journal of CyberSecurity*, vol. 2023, pp. 7-15, 2023.

[6] X. Lu, S. Zhang, P. Hui and P. Lio, "Continuous authentication by free-text keystroke based on CNN and RNN," *Computers & Security*, vol. 96, p. 101861, 2020.

[7] A. S. Elmaghraby and Y. Zheng, "Cybersecurity enhancement using recurrent neural networks and keystroke dynamics," in *SPIE*, online, 2021.

[8] Olga-Dimitra Asvesta, Eleni Vrochidou and George A. Papakostas, "IKDD: A Keystroke Dynamics Dataset for User Classification," *Information*, vol. 15, p. 511, 2024.

[9] A. A. Wahab, D. Hou, S. Schuckers and A. Barbir, "Utilizing Keystroke Dynamics as Additional Security Measure to Protect Account Recovery Mechanism," in *ICISSP*, online, 2021.

[10] A. Arsh, N. Kar, S. Das and S. Deb, "Multiple approaches towards authentication using keystroke dynamics," in *Procedia Computer Science*, India, 2024.

[11] A. M. Gedikli and M. O. Efe, "A simple authentication method with multilayer feedforward neural network using keystroke dynamics," in *Springer International Publishing*, Istanbul, Turkey, 2020.

[12] E. Maiorana, H. Kalita and P. Campisi, "Deepkey: Keystroke dynamics and CNN for biometric recognition on mobile devices," in *IEEE*, Roma, Italy, 2019.

[13] G. Zhao, J. Yang, J. Chen, G. Zhu, Z. Jiang, X. Liu and B. Zhang, "Keystroke dynamics identification based on triboelectric nanogenerator for intelligent keyboard using deep learning method," *Advanced Materials Technologies*, vol. 4, no. 1, p. 1800167, 2019.

[14] K. Elliot, J. Graham, Y. Yassin, T. Ward, J. Caldwell and T. Attie, "A comparison of machine learning algorithms in keystroke dynamics," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, 2019.

[15] T. Ramu, K. Suthendran and T. Arivoli, "Machine learning based soft biometrics for enhanced keystroke recognition system," *Multimedia Tools and Applications*, vol. 79, no. 15, pp. 10029-10045, 2020.

[16] K. W. Tse and K. Hung, "User behavioral biometrics identification on mobile platform using multimodal fusion of keystroke and swipe dynamics and recurrent neural network," in *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Malaysia, 2020.

[17] L. A. Gabralla, "Dense Deep Neural Network Architecture for Keystroke Dynamics Authentication in Mobile Phone," *Adv. Sci. Technol. Eng. Syst. J.*, vol. 5, no. 6, pp. 307-314, 2020.

[18] A. Landowska and A. Kołakowska, "Keystroke dynamics patterns while writing positive and negative opinions," *Sensors*, vol. 21, no. 17, p. 5963, September 2021.

[19] A. Thakare, S. Gondane, N. Prasad and S. Chigale, "A machine learning-based approach to password authentication using keystroke biometrics," in *Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication: Proceedings of MDCWC 2020*, Singapore, 2021.

[20] A. Rahman, M. E. Chowdhury, A. Khandakar, S. Kiranyaz, K. S. Zaman, M. Reaz and M. A. Kadir, "Multimodal EEG and keystroke dynamics based biometric system using machine learning algorithms," *IEEE Access*, vol. 9, pp. 94625-94643, 2021.

[21] H. C. Chang, J. Li and M. Stamp, "Machine Learning-Based Analysis of Free-Text Keystroke Dynamics," in *Artificial Intelligence for Cybersecurity*, Cham: Springer, Springer International Publishing, 2022, pp. 331--356.

[22] S. Kar, A. Bamotra, B. Duvvuri and R. Mohanan, "KeyDetect--Detection of anomalies and user based on Keystroke Dynamics," *arXiv*, vol. 1, p. 2304.03958, 2023.

[23] T. Xi, I. Kuzminykh, B. Ghita and T. Bakhshi, "Evaluating Learning Algorithms for Keystroke Based User Authentication," in *2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, Istanbul, Turkiye, 2023.

[24] S. Roy, J. Pradhan, A. Kumar, D. Adhikary, U. Roy, D. Sinha and R. K. Pal, "A systematic literature review on latest keystroke dynamics based models," *IEEE Access*, vol. 10, pp. 92192-92236, 2022.

[25] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, Lisbon, Portugal, 2009.

[26] S. Szeghalmy and A. Fazekas, "A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning," *Sensors*, vol. 23, no. 4, p. 2333, 2023.

[27] S. Sulaiman, I. Ibraheem and S. Hameed, "Credit Card Fraud Detection Using Improved Deep Learning Models," *Computers, Materials & Continua*, vol. 78, no. 1, pp. 1049-1069, 2024.

[28] Y. Zheng, L. Yu, S. Haque, P. Zhang and A. S. Elmaghraby, "Improving cybersecurity through deep learning on keystroke and mouse dynamics," in *Multimodal Image Exploitation and Learning 2022*, Florida, 2022.

[29] S. Haboubi and O. B. Salem, "Energy Consumption Prediction of Smart Buildings by Using Machine Learning Techniques," *Iraqi Journal of Science*, vol. 64, no. 12, pp. 6509-6521, 2023.

[30] M. Al-jumaili and J. Bazzi, "Cyber-Attack Detection for Cloud-Based Intrusion Detection Systems," *Mesopotamian Journal of CyberSecurity*, vol. 2023, pp. 170-182, 2023.

[31] M. Aljabri, A. Shaahid, F. Alnasser, A. Saleh, D. Alomari, M. Aboulnour and A. Althubaity, "IoT Attacks Detection Using Supervised Machine Learning Techniques," *HighTech and Innovation Journal*, vol. 5, no. 3, pp. 534-550, 2024.