Enhanced-TypeNet for Biometric Keystroke Authentication Using Key Embedding

Mahmoud Bahaa
Research and Innovation Department, T2 Company, Riyadh, Saudi Arabia
m.bahaa@t2.sa (corresponding author)

Fahad A. Aloufi
Department of Cybersecurity, College of Computer, Qassim University, Qassim, Saudi Arabia | Research and Innovation Department, T2 Company, Riyadh, Saudi Arabia
faa.alharbi@qu.edu.sa

Received: 13 April 2025 | Revised: 20 May 2025 | Accepted: 6 June 2025

 DOI: https://doi.org/10.48084/etasr.11468

ABSTRACT

Deep learning models have demonstrated significant success across various domains, including authentication tasks such as biometric authentication using keystrokes. This paper introduces an advanced model, named Enhanced-TypeNet, based on Long Short-Term Memory (LSTM) with an embedding layer. This model exhibited exceptional performance in large-scale free-text scenarios. Enhanced-TypeNet is an improved version of the original TypeNet model, which had already achieved state-of-the-art results. Model training employed two distinct learning approaches, distinguished by their loss functions: softmax and triplet loss. Building on this foundation, Enhanced-TypeNet shows notable improvements, especially in scenarios with limited enrollment sequences. In the most challenging scenario, involving only one enrollment sequence, the proposed model demonstrates significant improvements in the Equal Error Rate (EER) of 8% and 2% for softmax and triplet loss, respectively, on physical devices compared to the original model. On touchscreen devices, the model achieves even greater enhancements, with EER improvements of approximately 10% and 9% for softmax and triplet loss, respectively. These findings highlight the efficacy of Enhanced-TypeNet across diverse authentication scenarios, emphasizing its potential for real-world applications.

Keywords-DL; LSTM; embedding; biometric authentication; keystroke authentication

I. INTRODUCTION

Keystroke biometrics is an evolving field that sits at the confluence of cybersecurity and biometric authentication, leveraging the unique patterns in the way an individual types on a keyboard. This innovative modality of biometric systems capitalizes on the rhythm and timing between keystrokes, which are as distinctive as fingerprints [1, 2]. With the proliferation of digital platforms and the growing need for secure authentication mechanisms, keystroke biometrics presents a novel and non-intrusive method to ensure user identity verification and security [3]. The premise of keystroke dynamics lies in the measurement of typing characteristics, such as dwell time (the duration for which a key is pressed) and flight time (the interval between releases of successive keys). These dynamics are not only unique to each individual but are also difficult to replicate or forge, making them a secure form of biometric authentication [4, 5]. Traditional methods to analyze keystroke patterns have relied on statistical and machine learning techniques, which, while effective to some extent, are limited by the complexity and variability of human typing behavior [1, 6].

In recent years, deep learning has emerged as a transformative force in artificial intelligence, offering substantial advances in processing and interpreting complex data. Its application in biometrics, particularly in keystroke dynamics, has shown promise in enhancing the accuracy and reliability of user identification [7]. Deep learning models, characterized by their layered architecture, can extract high-level features from raw keystroke data [8, 9]. Using neural networks, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), deep learning can accommodate the temporal sequence of keystrokes, capturing the temporal dependencies and nuances in typing patterns that are indicative of a user's biometric identity [7].

The field of user authentication based on keystroke dynamics has seen significant advances through the integration of various features and machine learning models, leading to more robust and accurate authentication systems. In [10], a comprehensive study explored the integration of heterogeneous features for mobile device user authentication. This approach combined temporal, spatial, and contextual information to improve accuracy and robustness. By capturing a wide range of keystroke dynamics characteristics, the proposed feature

extraction method reduced the Equal Error Rate (EER) compared to traditional methods relying solely on temporal features. This multifaceted approach demonstrated the potential to leverage diverse feature sets to create more reliable and secure authentication systems for mobile devices. In [11], keystroke dynamics was combined with Hybrid Nano-Generators (HNGs) and artificial intelligence. The HNGs captured detailed mechanical energy signals generated during typing, which were then processed using advanced AI techniques. This hybrid approach significantly improved the accuracy and robustness of user authentication, achieving lower EER compared to conventional methods. This study underscored the potential of combining innovative hardware components with AI-driven analysis to develop more secure and reliable biometric systems.

In [12], a novel three-step authentication scheme was proposed that combined keystroke and motion sensor-based methods. Initially, the model determined the mobile device's orientation (sitting, walking, and relaxing) and identified the user's typing position (landscape or portrait). Authentication was carried out using a classification template based on these factors. The model's performance was evaluated using K-Nearest Neighbors (KNN) and Random Forest (RF) classifiers, with RF outperforming KNN in initial testing. Subsequent optimization with Particle Swarm Optimization (PSO) further improved RF's performance across all metrics, achieving an accuracy of 92.58% and EER of 2.2% in the relaxing landscape position. This study underscored the importance of determining position before authentication, noting that relaxing and walking positions are optimal for keystroke-based authentication.

In [13], the use of transformer architectures in keystroke dynamics was explored. The TypeFormer model leveraged self-attention mechanisms to capture long-range dependencies in keystroke sequences, addressing the sequential and temporal aspects of typing behavior more effectively than traditional models. TypeFormer outperformed existing models, such as CNNs and RNNs, in terms of accuracy and robustness, showcasing its potential for real-world continuous authentication applications. This study highlighted the advantages of transformer architectures in biometric authentication, emphasizing their ability to handle complex temporal patterns and improve security on mobile devices.

A notable contribution to continuous authentication was made in [14], presenting an innovative approach to biometric authentication using a combination of CNN and RNN. Traditional keystroke authentication methods often struggle with continuous scenarios, where seamless user verification is required over time. By integrating CNNs for feature extraction and RNNs for temporal pattern recognition, the model captured both spatial and temporal characteristics of keystroke behavior. This hybrid architecture significantly improved authentication accuracy and robustness compared to previous models, making it a promising solution for real-time continuous user verification.

In [15], Partially Observable Hidden Markov Models (POHMM) were used for keystroke dynamics. POHMM effectively captures the sequential and stochastic nature of keystroke patterns, which are inherently noisy and variable. By modeling the hidden states of a user's typing behavior, the POHMM-based approach significantly improves authentication performance compared to traditional Hidden Markov Models (HMMs) and other machine learning techniques. This work underscored the potential of POHMMs to improve the accuracy and reliability of keystroke dynamics authentication systems.

In [16], a dynamic keystroke technique used Deep Belief Networks (DBN), a generative-discriminative hybrid neural network architecture, for secure authentication in e-assessment systems. This approach extracted multiple keystroke features: digraphs (timing patterns between two consecutive keystrokes), trigraphs (patterns among three successive keystrokes), and n-graphs (word-specific keystroke patterns). This comprehensive feature extraction enhances the system's ability to create unique user signatures to distinguish between genuine users and impostors. This study demonstrated the effectiveness of deep learning models for keystroke-based authentication, providing valuable insights that align with this work on neural network architectures for biometric verification.

TypeNet [17] is a deep learning architecture specifically designed for keystroke biometric authentication tasks, celebrated for its impressive results. This study aimed to further improve TypeNet's performance by refining its architecture and optimizing the training procedure. This study presents an improved model, called Enhanced-TypeNet, along with a comprehensive evaluation by comparing it to the original TypeNet model.

II. KEYSTROKE DATASETS

The proposed model was trained and evaluated using the Aalto University Datasets [18, 19]. The dataset in [18] (Dhakal) contains approximately 5 GB of keystroke data from 168,000 participants who used desktop keyboards. On the other hand, the dataset in [19] (Palin) includes almost 4 GB of keystroke data collected from 260,000 participants using mobile devices. Both datasets followed a consistent data collection method based on controlled free text [20]. Participants were instructed to memorize and type English sentences selected randomly from a set of 1,525 examples derived from the Enron mobile email and the Gigaword Newswire corpus. These sentences ranged from 3 words to a maximum of 70 characters, but participants could exceed this limit due to potential typing errors or additions.

In the Dhakal dataset, each participant engaged in 15 sessions, which involved typing a single sentence on a desktop or laptop keyboard. However, in the Palin dataset, only 23% of participants (60,000 out of 260,000) completed at least 15 sessions after starting the typing test. For consistent comparisons, this study focuses on the 60,000 participants in the Palin dataset who completed their initial 15 sessions. Regarding the Palin dataset, it was observed that it contained many missing keycode values, which will affect the training of the proposed model. Therefore, out of the 60,000 participants, 32 K subjects were filtered out, keeping entries that contain at least 5 characters, at least 3 unique characters, and no null keycode values.

III. SYSTEM DESCRIPTION

A. Data Parsing and Feature Extraction

A thorough preprocessing protocol was applied to ensure data suitability. Initially, a discerning approach was employed to selectively retain relevant fields, such as PRESS_TIME, RELEASE_TIME, KEYCODE, PARTICIPANT_ID, and TEST_SECTION_ID, essential for preserving dataset integrity and relevance. PRESS_TIME represents the timestamp when the key was pressed (in ms), RELEASE_TIME is the timestamp when the key was released (in ms), KEYCODE is the keycode of the pressed key, PARTICIPANT_ID includes a unique identifier for each participant, and TEST_SECTION_ID is a unique identifier for the given typing test section.

Following this, a systematic process was employed to iteratively handle individual keystroke files in the Dhakal dataset. Each file was parsed and transformed into structured data frames, retaining only the relevant attributes. Upon completion of this iterative parsing of all files, the datasets were combined into a single data frame, which was then used for feature extraction. In particular, by leveraging the TEST_SECTION_ID column, the data frame was grouped, and features were extracted for each experimental segment. For the Palin dataset, a similar approach was used, but the parsing was done directly from a SQL table.

Given the intrinsic importance of timing nuances in keystroke dynamics, it is vital to extract features that accurately capture the unique typing patterns of individuals. This study meticulously calculates four fundamental features:

- Holding Time (HL): Signifying the duration a key remains depressed.
- Press Latency (PL): Representing the temporal gap between the depression of two consecutive keys.
- Release Latency (RL): Indicating the temporal interval between the release of two consecutive keys.
- Inter-event Latency (IL): Denoting the temporal span between the release of one key and the subsequent key's depression.

Each of these features provides a distinct insight into an individual's typing behavior, collectively offering a holistic portrayal of their typing dynamics. These time-based features (HL, PL, RL, and IL) are scaled by a factor of 1000, converting milliseconds to seconds to keep the time representation consistent throughout the dataset. For the efficient computation of IL, a Just-In-Time (JIT) compiled function was utilized. This function is designed to accept a two-by-two array containing the PRESS_TIME and RELEASE_TIME values corresponding to two consecutive keys.

Upon completion of the extraction process, the sequences corresponding to keycodes, which inherently represent the specific keys being pressed, along with the time features (HL, PL, RL, IL), are turned into arrays of the same size through truncation or zero-padding. A maximum sequence length was established to make all sequences the same length, ensuring smooth integration into the neural network.

In summary, the result of this extraction process is a structured dictionary comprising padded sequences of keycodes and their corresponding time features, meticulously prepared for transmission to the neural model.

B. Model Architecture

Long Short-Term Memory (LSTM) networks are highly effective for sequential data analysis because of their ability to capture temporal dependencies. This makes them ideal for keystroke authentication tasks, where timing and order of keystrokes are crucial. Unlike traditional neural networks, LSTMs have memory cells that retain information over time, enabling them to model sequences with long-range dependencies. This ability allows LSTMs to learn intricate patterns in keystroke data, enhancing the accuracy and robustness of authentication systems. Their ability to handle variable-length input sequences further underscores their suitability for keystroke authentication.

The proposed model architecture introduces a novel approach by incorporating two distinct types of input: the numerical representation of keycodes, represented as integer tokens, and other associated keystroke features, inspired by the TypeNet model [17]. Although keycodes are represented numerically, they inherently possess categorical characteristics, and each serves as a token representing a specific key. Recognizing the inefficiency of directly feeding these numerical representations into the model, a solution was devised utilizing a word embedding layer. Each keycode can be considered a distinct token, allowing the use of NLP-like techniques. In NLP, each word is converted to a unique numerical integer in a process called tokenization, and these numerical tokens are then fed as input to a trainable word embedding layer. This layer transforms these tokens into multidimensional vectors. Using a similar approach, converting each keycode into an embedding vector can effectively capture and utilize the unique characteristics and relationships between keystrokes. In the experiment, this layer transforms each keycode token into a trainable 32-dimensional dense vector, which is then concatenated with other numerical features. By structuring the model pathway to accommodate the inherent nature of these inputs, this architecture follows a systematic approach to processing each input type, ultimately merging their processed representations. This integration facilitates the capture of intricate relationships and contextual information that is collectively presented by these inputs.

The proposed model incorporates two distinct inputs, each tailored to handle specific aspects of keystroke data. The first input is designed to accommodate sequences of keycodes, each sequence having a length of M. These keycodes are transformed by an embedding layer, which maps each keycode to a 32-dimensional dense vector. This embedding operation is crucial to representing categorical variables, such as keycodes, in a dense space. In particular, the embedding layer in this model is initialized to accommodate a maximum of 256 distinct keycodes. Simultaneously, the model processes other keystroke features through the features_input layer (the second input), structured to handle sequences with a shape of $(M, 4)$. Initially, a masking layer is applied to ensure that any padding introduced in the sequences does not affect the model's

computations adversely. Subsequently, the sequence data are fed into two successive LSTM layers, each comprising 64 units, enabling the model to capture temporal dependencies within the input sequences.

Following the parallel processing paths, the model merges the representations by concatenating the embedded keycode vectors with the processed keystroke features. This combined representation is then passed through another LSTM layer aimed at generating a singular output vector that encapsulates the context of the entire input sequence. Subsequently, two models were trained using softmax and triplet losses. In the softmax model, as presented in Figure 1(a), a dense layer is added to classify the input sequences into distinct categories, representing different subjects or typing personalities, with the softmax activation function ensuring the output presentation as class probabilities. In the triplet model, as presented in Figure 1(b), the loss is calculated directly on the output embeddings using the Euclidean distance as a metric.

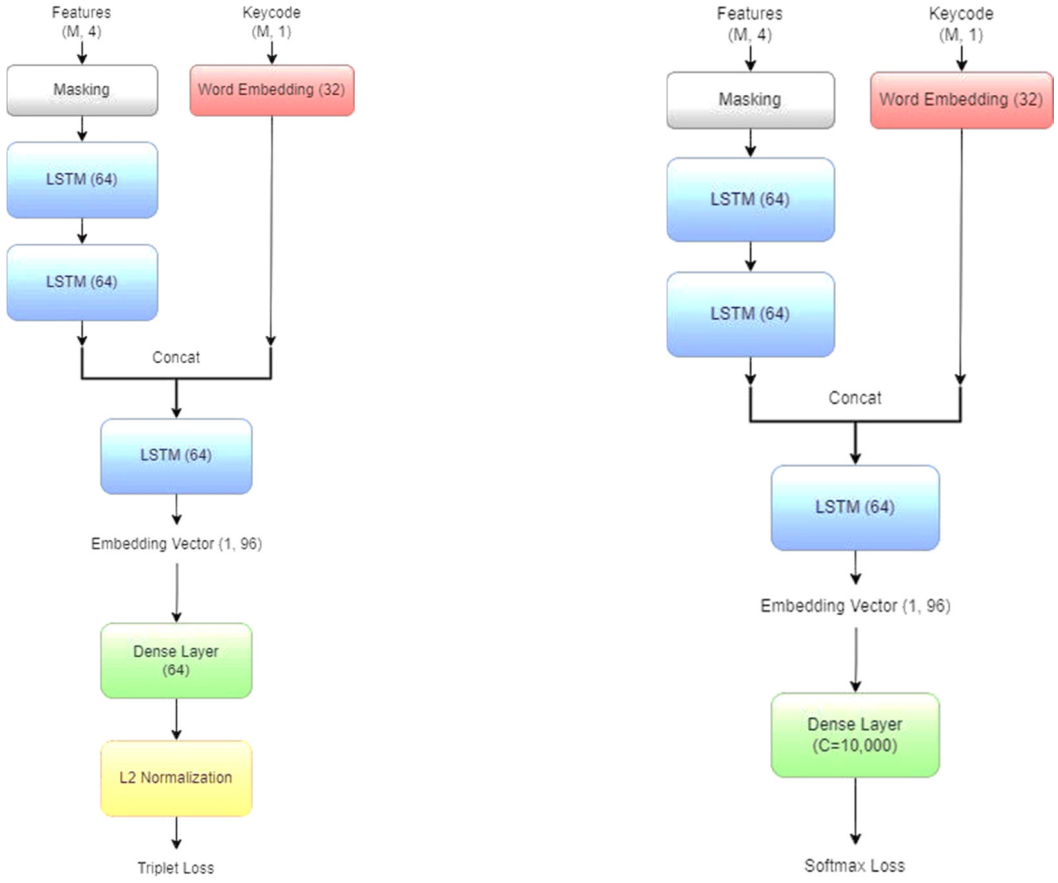

Fig. 1. (a) Model architecture with Triplet loss, (b) Model architecture with Softmax loss.

1) Softmax Loss

In keystroke authentication, the softmax function serves a critical role in classification tasks. It transforms the output of the preceding layer, which captures essential features of keystroke patterns, into a probability distribution over classes. Notably, during inference, the embedding output from the layer preceding the final classification layer is utilized. This embedding represents a condensed version of the input data, enabling efficient computation and accurate classification for authentication purposes. The softmax function $\sigma(z)_i$ for the i-th class, where z is the input vector and i ranges over all classes, is defined as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \quad (1)$$

where e is Euler's number, z_i represents the raw score for the i-th class, and N is the total number of classes. During inference, the embedding output is fed into the softmax function to generate probabilities, facilitating the classification of keystroke patterns.

2) Triplet Loss

In the realm of keystroke authentication, the triplet loss function assumes paramount importance for metric learning tasks. Its objective is to generate embeddings that effectively encapsulate the distinctive characteristics of keystroke patterns, facilitating precise authentication. This embedding serves as a condensed representation of keystroke data, enabling efficient computation and accurate similarity comparisons for authentication purposes.

The triplet loss function $L(A, P, N)$ operates by evaluating triplets of embeddings: an anchor (representing a genuine user's keystroke pattern), a positive example (denoting another instance of the same genuine user's keystroke pattern), and a negative example (signifying an impostor's keystroke pattern). During inference, the embedding output is utilized within the triplet loss function to compute the loss, facilitating the assessment of similarity between keystroke patterns for authentication. This process aids in effectively discerning genuine user keystroke patterns from impostor keystroke patterns, ensuring robust authentication performance.

$$L(A, P, N) = \max\{d(A, P) - d(A, N) + \alpha, 0\} \qquad (2)$$

where $d(.)$ represents a distance metric (e.g., Euclidean distance) between two embeddings, and α is a margin parameter that controls the degree of separation between positive and negative examples.

C. Implementation Details

Model training was carried out using the TensorFlow framework and a single RTX 4080 GPU. The AdamW optimizer was employed with a learning rate set to 0.005 and a weight decay parameter of 0.1. AdamW optimization is a variant of Adam that is enhanced with weight decay regularization techniques, as described in [21].

In the initial training phase, the softmax model was trained for 200 epochs, incorporating early stopping if no improvement was observed after 10 epochs. During this training phase, no additional regularization techniques were applied, apart from the weight decay inherent in AdamW optimization. Upon reaching convergence, a recurrent dropout rate of 0.2 was introduced to the LSTM layers, while a dropout rate of 0.5 was applied to the inputs of the second LSTM layer. This step aims to alleviate potential overfit concerns. Subsequently, the model underwent a second training episode, in which regularization techniques were implemented. In particular, the application of these regularizations yielded additional accuracy improvements in softmax training. Figure 2 presents the detailed model architecture.

In the subsequent phase, the triplet model underwent full training, employing weight decay as the sole regularization technique. Training with triplet loss has its challenges, as it heavily relies on the proper selection of positive and negative triplets within each minibatch. To address this, semi-hard negative mining was utilized for negative triplets and hard positive mining for positive triplets. For each batch, an equal number of classes and an equal number of examples per class were crafted. Initially, a small batch size of 625 was used, progressively scaling up in multiples of 2 after each convergence. This escalation poses a challenge to the model, as larger batch sizes inherently introduce more complex negative triplets. Batch size augmentation continued to 10,000, the maximum capacity accommodated by GPU memory.

For each test subject, a subset of keystroke samples was designated as the gallery set, comprising G sequences used for enrollment, while Q query samples were selected from the remaining sequences for authentication testing. The experiment varied parameters such as the number of gallery embeddings G and query embeddings per subject Q to evaluate performance under different conditions. Impostor scores were generated by comparing the query embeddings of each subject against the gallery embeddings of others, simulating real-world scenarios with potential impostor inputs.

EER is a crucial metric in biometric systems, representing the point where the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR) on the ROC curve, providing a balanced measure of the system's accuracy. To calculate the EER, the true labels of test subjects were compared with test scores generated by the recognition system, computed as the average Euclidean distance between each query embedding and the gallery embeddings. The ROC curve visualizes the system's performance across various threshold values.

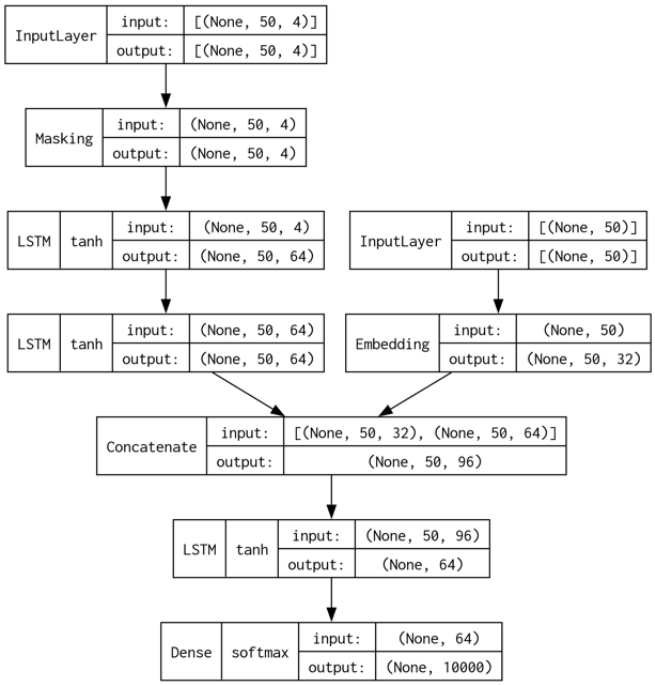

Fig. 2. Detailed model of softmax architecture.

IV. EXPERIMENTAL METHOD

Consistent with the approach adopted in the TypeNet model, the experiments were segmented into two procedures: authentication and identification.

A. Authentication Procedure

Gallery samples $x_{i,g}$ from subject i in the test set were compared with a query sample $x_{j,q}$. The comparison is either a genuine match if $i = j$ or an imposter match if $i \neq j$. The test score is determined by averaging the Euclidean distances between each gallery embedding vector $f(x_{i,g})$ and the query embedding vector $f(x_{j,q})$:

$$TestScore = \frac{1}{G} \sum_{g=1}^{G} \left\| f(x_{i,g}) - f(x_{j,q}) \right\| \qquad (3)$$

where G is the number of sequences in the gallery (i.e., the enrollment samples), and q is the query sample from subject j.

Each subject has 15 sequences, with 5 sequences retained for testing, providing 5 genuine test scores per subject. To assess performance relative to the number of enrollment sequences, G varied between 1 and 10. To generate impostor scores, a test sample from each remaining subject was selected for each enrolled subject. Following the procedure of TypeNet, the number of enrolled subjects k was set to 1000 for the desktop and mobile models. Consequently, each subject has 5 genuine scores and $k-1$ impostor scores. In keystroke dynamics authentication, it is typical to have a higher number of impostor scores compared to genuine scores. The results presented in the next section are expressed as the EER, where the FAR equals the FRR. These error rates are calculated for each subject and then averaged over all k subjects [22].

B. Identification Procedure

In forensic applications, identification scenarios are commonly encountered, where the final decision depends on a compilation of evidence. Biometric recognition technology is vital in these situations, as it provides a list of potential candidates termed the background set B in this study. The Rank-1 identification rate evaluates the system's ability to unambiguously identify the target individual from all subjects in the background set. Rank-n represents the accuracy when a ranked list of n profiles is considered, with the final determination made manually or automatically based on additional evidence.

The dataset contained 15 sequences per test subject, which were then divided into two categories: Gallery (comprising 10 sequences $G = 10$) and Query (comprising 5 sequences $Q = 5$). Evaluation of the identification rate involved comparing the Query set samples $x_{j,q}^{Q}$, where $q = 1, \dots, 5$ represents the test subject j, against the Background Gallery set $x_{i,g}^{G}$, where $g = 1, \dots, 10$ represents all background subjects. The distance between each gallery embedding vector $f(x_{i,g}^{G})$ and each query embedding vector $f(x_{j,q}^{Q})$ was calculated using the average Euclidean distance formula:

$$Avg.Dist. = \frac{1}{G \times Q} \sum_{g=1}^{G} \sum_{q=1}^{Q} \| f(x_{i,g}^{G}) - f(x_{j,q}^{Q}) \| \quad (4)$$

V. EVALUATION RESULTS

This section presents the comprehensive evaluation results for the keystroke biometric task, encompassing both authentication and identification aspects. Each subsection provides a detailed analysis of the model's performance in the respective domains, shedding light on its efficacy and robustness in verifying user authenticity and identifying individuals based on their keystroke dynamics. The results were compared with the performance of the TypeNet model. To ensure robustness and generalization, the dataset was divided into separate training, validation, and testing splits, ensuring no overlap between subjects across all groups. In the desktop dataset, 10,000 subjects were used for softmax training and 60,000 for triplet training. Similarly, in the mobile dataset, 10,000 subjects were assigned for softmax training, while 30,000 were designated for triplet training. All subjects had a minimum of 15 input entries, ensuring adequate representation for each individual. To maintain consistency between experiments, the number of subjects in the testing set was fixed at 1000. This careful data splitting allows for a thorough model evaluation and ensures a reliable assessment of authentication and identification capabilities.

A. Authentication Results

In the realm of free-text keystroke authentication algorithms, a crucial factor influencing performance is the volume of keystroke data collected per subject for enrollment [23]. This section explores the impact of varying the number of gallery samples on authentication accuracy and model performance when dealing with a limited number of gallery samples. Moreover, the length of the keystroke sequence is also a crucial factor to consider. Drawing from the insights provided in [17], where sequence lengths M of 50 and 70 demonstrated optimal results, these experiments were based on a sequence length $M = 50$ to ensure robust performance evaluation.

The evaluation was carried out using a dataset comprising 1000 test subjects ($k = 1000$), each associated with a set of behavior embeddings generated by the proposed recognition system. For each subject, 5 query embeddings ($Q = 5$) were randomly selected to assess the system's performance. The evaluation was performed with varying numbers of gallery embeddings per subject, specifically 1, 2, 5, 7, and 10.

Genuine scores were calculated as the average Euclidean distance between gallery and query embeddings from the same subject, while impostor scores were calculated using embeddings from different subjects. Test scores were normalized to ensure a consistent similarity scale using the formula:

$$y_{score} = \frac{1}{1+distance} \quad (5)$$

EER was determined by finding the threshold at which the False Positive Rate (FPR) equals 1 minus the True Positive Rate (TPR) using the ROC curve. Figure 3 and Figure 4 illustrate the ROC curves with the $x + y = 1$ line, and highlight the EER intersection point.

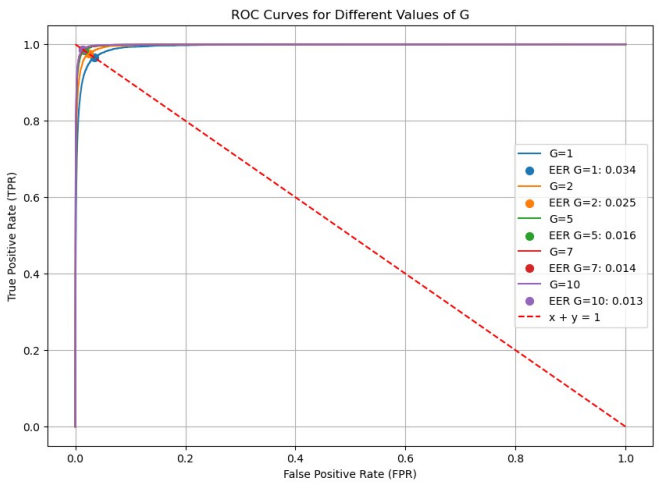

Fig. 3. Desktop triplet EER.

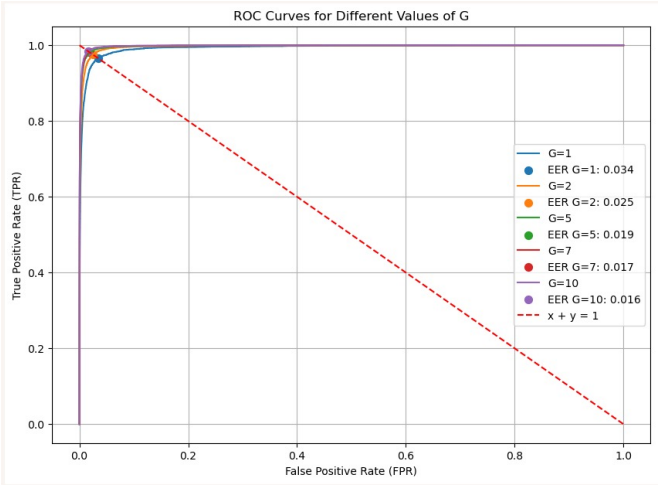

Fig. 4. Mobile triplet EER.

The experiments showed that the proposed Enhanced-TypeNet significantly outperforms TypeNet. This improvement is evident in both desktop and mobile scenarios, regardless of the number of gallery samples, as shown in Tables I and II. In the most challenging one-shot authentication case ($G = 1$), compared to TypeNet, the proposed model reduces EER by approximately 8% in the softmax model and 2% in the triplet model for the desktop scenario. For the mobile scenario, the results show an improvement of about 10% in the softmax model and around 9% in the triplet model. In the particular case of Rank-1, which is the most challenging scenario, it is obvious that the proposed model significantly improved the results, particularly the triplet version. In the desktop scenario, the accuracy increased by roughly 30% in the softmax version and by about 23% in the triplet version. In the mobile scenario, Enhanced-TypeNet demonstrated substantial improvements, with accuracy increasing from 23.5% to 71.2% in the softmax version and from 25.5% to 92.3% in the triplet version.

TABLE I. EER (%) ACHIEVED USING THE DESKTOP TEST SET

Model	**Enrolment sequences per subject *G* (Desktop)**				
	1	**2**	**5**	**7**	**10**
TypeNet (softmax) [17]	16.8	13.1	10.8	9.2	8.8
Enhanced-TypeNet (softmax)	8.72	6.92	5.16	5.04	4.7
TypeNet (triplet) [17]	5.4	3.6	2.2	1.8	1.6
Enhanced-TypeNet (triplet)	3.4	2.5	1.6	1.4	1.3

G represents the number of enrollment sequences per subject.

TABLE II. EER (%) ACHIEVED USING THE MOBILE TEST SET

Model	**Enrolment sequences per subject *G* (Mobile)**				
	1	***2***	***5***	***7***	***10***
TypeNet (softmax) [17]	17.2	15.4	13.8	13.4	12.7
Enhanced-TypeNet (softmax)	7.33	5.59	4.56	4.04	3.93
TypeNet (triplet) [17]	12.6	10.7	9.2	8.5	8.0
Enhanced-TypeNet (triplet)	3.4	2.3	1.8	1.7	1.5

G represents the number of enrollment sequences per subject.

B. Authentication: Generalization to Different Scenarios

This section explores the extent to which trained models can be generalized in different scenarios. The experiments used the best model (triplet), trained separately on the desktop and mobile datasets, and a third model, trained on a mixed dataset. Then, these models were nested to determine their adaptability and robustness under varying conditions. The results in Table III show that the Enhanced-TypeNet models trained in both mobile and mixed datasets demonstrated effective generalization across different scenarios. However, the model trained on the desktop achieved worse results when tested on the mobile dataset. This behavior likely occurs due to changes in the location of keys between the mobile and desktop keyboards. In particular, the TypeNet model trained on the mixed dataset achieved an EER of 17.9% on desktop testing data and 12.6% on mobile testing data. In contrast, Enhanced-TypeNet significantly improved these results, achieving EERs of 1.8% on desktop and 2.12% on mobile testing data, highlighting a substantial performance improvement.

In addition, the use of an input flag was investigated as an indicator of input type (desktop or mobile) during training. Although this approach resulted in only slight improvements, it suggests that the trained model was already capable of generalizing across different scenarios, as shown in Table III.

TABLE III. EER (%) FOR THE FOUR MODELS

Test dataset	**Model**	**Model yype**			
		Desktop	**Mobile**	**Mixture**	**Mixture (with flag)**
Aalto desktop [17]	[17]	2.2	21.4	17.9	–
	Proposed	1.6	3.01	1.8	1.86
Aalto Mobile	[17]	13.7	9.2	12.6	–
	Proposed	18.59	1.8	2.12	2.02
Aalto Mixture	Proposed	–	–	1.97	1.9

C. Identification Results

A rigorous evaluation method was used to measure the effectiveness of the biometric recognition system in identification scenarios. Drawing inspiration from forensic practices, the evaluation protocol aimed to determine the system's proficiency in identifying a specific subject from a group of subjects, relying on a collection of evidence. The test data for each subject was divided into two parts: the Gallery set and the Query set. For every test subject, a Query set was randomly selected, containing a set number of embeddings ($Q = 5$). Furthermore, a matching set of Gallery embeddings was selected from the remaining embeddings for that subject in the dataset ($G = 10$).

The Euclidean distances between each Query embedding and all Gallery embeddings were used to calculate the distance between inputs. These distances formed the basis for subsequent identification evaluations. The Rank-n metric was used to assess the system's identification accuracy. For each Query, the distance to each gallery was determined, and then all distances for each subject against all others were averaged. The results were sorted based on the n smallest distances, indicating the closest matches in the Gallery set. A match was

considered correct if the actual subject was among the n smallest distances. The rank-n accuracy was calculated as the ratio of correctly identified subjects to the total number of subjects. This metric provided a detailed insight into the system's performance across various identification scenarios.

Table IV presents the identification accuracy results for TypeNet and enhanced-TypeNet in desktop and mobile scenarios. The results indicate that Enhanced-TypeNet performs more efficiently than TypeNet. In the particular case of Rank-1, which is the most challenging scenario, it is obvious that the proposed model significantly improved the results, particularly the triplet version. In the desktop scenario, the accuracy increased by roughly 30% in the softmax version and by about 23% in the triplet version. In the mobile scenario, Enhanced-TypeNet demonstrated substantial improvements, with accuracy increasing from 23.5% to 71.2% in the softmax version and from 25.5% to 92.3% in the triplet version.

TABLE IV. IDENTIFICATION ACCURACY (%)

Method	Scenario	Rank		
		1	50	100
TypeNet (softmax) [17]	D	47.5	96.3	98.7
Enhanced TypeNet (softmax)	D	78.5	99.9	99.9
TypeNet (Triplet) [17]	D	67.4	99.8	99.9
Enhanced TypeNet (Triplet)	D	90.5	100	100
TypeNet (softmax) [17]	M	23.5	82.6	91.4
Enhanced TypeNet (softmax)	M	71.2	99.9	100
TypeNet (Triplet) [17]	M	25.5	87.5	94.2
Enhanced TypeNet (Triplet)	M	92.3	100	100

D refers to the desktop scenario and M refers to the mobile scenario

D. Comparison with Recent Transformer-Based Models

Recent advances in keystroke dynamics authentication have seen the emergence of transformer-based architectures as powerful alternatives to traditional RNN models. To evaluate the performance of the Enhanced-TypeNet model in the context of these cutting-edge approaches, it was compared directly with TypeFormer [13], a recently proposed transformer architecture specifically designed for mobile keystroke biometrics. Table V presents the comparative results of Enhanced-TypeNet against TypeFormer and the original TypeNet model on the standard sequence length of 50 keystrokes. The Enhanced-TypeNet model consistently outperforms both the original TypeNet and the more recent TypeFormer architecture in all enrollment session configurations. The superior performance of Enhanced-TypeNet can be attributed to its key embedding approach, which captures the categorical nature of keystroke data more effectively.

TABLE V. COMPARISON WITH OTHER MODELS

Model	Enrolment sequences per subject G (Mobile)				
	1	2	5	7	10
TypeNet (triplet) [17]	12.6	10.7	9.2	8.5	8.0
TypeFormer [13]	6.17	4.57	3.25	2.86	2.54
Preliminary Transformer [13]	6.99	-	3.84	-	3.15
Enhanced TypeNet (triplet)	**3.4**	**2.3**	**1.8**	**1.7**	**1.5**

VI. CONCLUSION

This study presented a free-text keystroke biometrics system that uses RNN architectures and different training strategies. The Enhanced-TypeNet model, an improved version of TypeNet, showed superior performance on a comprehensive set of authentication and identification experiments that included various datasets from desktop keyboards and mobile devices. The results demonstrate that the Enhanced-TypeNet models trained using triplet loss consistently outperformed the original TypeNet models, especially in scenarios with a large number of subjects but limited enrollment samples per subject. This highlights the effectiveness of the proposed enhancements and training techniques in improving the robustness and generalization capabilities of the keystroke biometrics system.

The large-scale nature of the experiments and the results achieved underscore the practical applicability and potential of the developed free-text keystroke biometric systems. These systems can contribute to enhanced user authentication and identification in a variety of security-critical applications, making important strides toward more reliable and convenient user verification. Overall, this study presents significant advancements in the field of keystroke biometrics, providing valuable insights and a solid foundation for further research and real-world deployments of such technologies.

ACKNOWLEDGMENT

This work was funded and supported by the Research and Innovation Department of the T2 company (www.t2.sa).

REFERENCES

[1] S. P. Banerjee and D. Woodard, "Biometric Authentication and Identification Using Keystroke Dynamics: A Survey," *Journal of Pattern Recognition Research*, vol. 7, no. 1, pp. 116–139, 2012, https://doi.org/10.13176/11.427.

[2] A. Kolakowska, "A review of emotion recognition methods based on keystroke dynamics and mouse movements," in *2013 6th International Conference on Human System Interactions (HSI)*, Sopot, Poland, Jun. 2013, pp. 548–555, https://doi.org/10.1109/HSI.2013.6577879.

[3] D. Buschek, A. De Luca, and F. Alt, "Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, Seoul Republic of Korea, Apr. 2015, pp. 1393–1402, https://doi.org/10.1145/2702123.2702252.

[4] J. Hernandez-Ortega, R. Daza, A. Morales, J. Fierrez, and J. Ortega-Garcia, "edBB: Biometrics and Behavior for Assessing Remote Education." arXiv, Dec. 10, 2019, https://doi.org/10.48550/arXiv.1912.04786.

[5] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Benchmarking desktop and mobile handwriting across COTS devices: The e-BioSign biometric database," *PLOS ONE*, vol. 12, no. 5, May 2017, Art. no. e0176792, https://doi.org/10.1371/journal.pone.0176792.

[6] S. F. N. Sadikan, A. A. Ramli, and M. F. Md. Fudzee, "A survey paper on keystroke dynamics authentication for current applications," presented at the Advances in Electrical Engineering and Electronic Engineering From Theory to Applications (Series 2): Proceedings of the International Conference of Electrical and Electronic Engineering (ICon3E 2019), Putrajaya, Malaysia, 2019, Art. no. 020010, https://doi.org/10.1063/1.5133925.

[7] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometrics recognition using deep learning: a survey," *Artificial*

Intelligence Review, vol. 56, no. 8, pp. 8647–8695, Aug. 2023, https://doi.org/10.1007/s10462-022-10237-x.

[8] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, "DeepSign: Deep On-Line Signature Verification," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 229–239, Apr. 2021, https://doi.org/10.1109/TBIOM.2021.3054533.

[9] A. Maalej and I. Kallel, "Does Keystroke Dynamics tell us about Emotions? A Systematic Literature Review and Dataset Construction," in *2020 16th International Conference on Intelligent Environments (IE)*, Madrid, Spain, Jul. 2020, pp. 60–67, https://doi.org/10.1109/IE49459.2020.9155004.

[10] J. Kim and P. Kang, "Freely typed keystroke dynamics-based user authentication for mobile devices based on heterogeneous features," *Pattern Recognition*, vol. 108, Dec. 2020, Art. no. 107556, https://doi.org/10.1016/j.patcog.2020.107556.

[11] P. Maharjan *et al.*, "Keystroke Dynamics based Hybrid Nanogenerators for Biometric Authentication and Identification using Artificial Intelligence," *Advanced Science*, vol. 8, no. 15, Aug. 2021, Art. no. 2100711, https://doi.org/10.1002/advs.202100711.

[12] B. S. Saini *et al.*, "A Three-Step Authentication Model for Mobile Phone User Using Keystroke Dynamics," *IEEE Access*, vol. 8, pp. 125909–125922, 2020, https://doi.org/10.1109/ACCESS.2020.3008019.

[13] G. Stragapede, P. Delgado-Santos, R. Tolosana, R. Vera-Rodriguez, R. Guest, and A. Morales, "Mobile Keystroke Biometrics Using Transformers," in *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, Waikoloa Beach, HI, USA, Jan. 2023, pp. 1–6, https://doi.org/10.1109/FG57933.2023.10042710.

[14] L. Xiaofeng, Z. Shengfei, and Y. Shengwei, "Continuous authentication by free-text keystroke based on CNN plus RNN," *Procedia Computer Science*, vol. 147, pp. 314–318, 2019, https://doi.org/10.1016/j.procs.2019.01.270.

[15] J. V. Monaco and C. C. Tappert, "The partially observable hidden Markov model and its application to keystroke dynamics," *Pattern Recognition*, vol. 76, pp. 449–462, Apr. 2018, https://doi.org/10.1016/j.patcog.2017.11.021.

[16] A. O. Aljahdali, F. Thabit, H. Aldissi, and W. Nagro, "Dynamic Keystroke Technique for a Secure Authentication System based on Deep Belief Nets," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10906–10915, Jun. 2023, https://doi.org/10.48084/etasr.5841.

[17] A. Acien, A. Morales, J. V. Monaco, R. Vera-Rodriguez, and J. Fierrez, "TypeNet: Deep Learning Keystroke Biometrics," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 1, pp. 57–70, Jan. 2022, https://doi.org/10.1109/TBIOM.2021.3112540.

[18] V. Dhakal, A. M. Feit, P. O. Kristensson, and A. Oulasvirta, "Observations on Typing from 136 Million Keystrokes," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Montreal, Canada, Apr. 2018, pp. 1–12, https://doi.org/10.1145/3173574.3174220.

[19] K. Palin, A. M. Feit, S. Kim, P. O. Kristensson, and A. Oulasvirta, "How do People Type on Mobile Devices?: Observations from a Study with 37,000 Volunteers," in *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, Taipei Taiwan, Oct. 2019, pp. 1–12, https://doi.org/10.1145/3338286.3340120.

[20] A. Wahab, D. Hou, S. Schuckers, and A. Barbir, "Utilizing Keystroke Dynamics as Additional Security Measure to Protect Account Recovery Mechanism:," in *Proceedings of the 7th International Conference on Information Systems Security and Privacy*, 2021, pp. 33–42, https://doi.org/10.5220/0010191200330042.

[21] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization." arXiv, Jan. 04, 2019, https://doi.org/10.48550/arXiv.1711.05101.

[22] A. Morales, J. Fierrez, and J. Ortega-Garcia, "Towards Predicting Good Users for Biometric Recognition Based on Keystroke Dynamics," in *Computer Vision - ECCV 2014 Workshops*, vol. 8926, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Springer International Publishing, 2015, pp. 711–724.

[23] J. Huang, D. Hou, S. Schuckers, and Z. Hou, "Effect of data size on performance of free-text keystroke authentication," in *IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2015)*, Hong Kong, Mar. 2015, pp. 1–7, https://doi.org/10.1109/ISBA.2015.7126361.