키보드 환경에서 키스트로크 인증을 위한 TypeFormer 구조 보완: Soft 마스킹 및 Set-Margin Triplet Loss 적용

(Enhancing TypeFormer for Keyboard-Based Keystroke Authentication through Soft masking and Set-Margin Triplet Loss)

서성찬*, 김수형**

(SeongChan Seo, SooHyung Kim)

요 약

본 연구는 키보드 기반 사용자 인증을 목적으로 TypeFormer를 free-text 환경에 적합하도록 확장하기 위해 마스킹과 Set-Margin Triplet Loss(SM-TL)를 도입한 개선된 학습 구조를 제안한다. 마스킹을 통해 자유 텍스트의 가변 길이로 인해 발생하는 패딩 신호를 효과적으로 제거하여 모델의 안정성을 높였으며, SM-TL은 클래스 단위의 마진 구조를 학습함으로써 Triplet Loss 대비 더욱 응집된 임베딩과 명확한 분리를 제공하였다. Aalto Desktop Dataset 기반 실험에서 제안 모델은 TypeNet과 기존 TypeFormer 모두를 지속적으로 상회하는 EER 성능을 보였다. 이를 통해 개선된 TypeFormer는 데스크톱 free-text 인증 시나리오에서 높은 실용성을 갖는 모델임을 입증하였다.

■ 중심어 : 키스트로크 다이내믹스 ; 트랜스포머 ; 생체인증 ; 손실 함수 ; 마스킹

Abstract

This study proposes an enhanced training framework that extends TypeFormer for keyboard-based free-text keystroke authentication by introducing masking strategies and Set-Margin Triplet Loss (SM-TL). The masking mechanism effectively suppresses padding-related signals caused by variable-length free-text inputs, thereby improving model stability, while SM-TL learns class-level margin structures to produce more compact intra-class embeddings and clearer inter-class separations compared to the conventional Triplet Loss. Experiments conducted on the Aalto Desktop Dataset demonstrate that the proposed model consistently outperforms both TypeNet and the original TypeFormer in terms of EER. These results confirm that the enhanced TypeFormer offers strong practical applicability for desktop free-text authentication scenarios.

■ keywords : Keystroke Dynamics ; Transformer ; Biometrics ; Loss Function ; masking

Ⅰ. 서 론

키스트로크 다이내믹스 기반 사용자 인증 연구에서는 오랫동안 통계적 거리 기반 기법이나 전통적인 머신러닝 모델이 활용되어 왔으며[8], 이후 시계열 특성을 직접 학습하기 위해 RNN 계열 모델(LSTM, GRU)과 CNN - RNN 결합 구조가 주된 접근 방식으로 사용되어 왔다[2, 9]. 특히 타이핑 데이터의 시간적 관계를 모델링하기 위해 순환 신경망 기반 구조가 사실상의 표준으로 자리 잡아 왔다.

RNN 계열 모델은 가변 길이 시퀀스를 처리하는 데 효과적인 구조이지만, 자유 텍스트 환경과

* 준회원, 전남대학교 정보보안융합학과 석사과정
** 종신회원, 전남대학교 AI융합대학 인공지능학부 교수
본 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업 연구 결과로 수행되었음(IITP-2023-RS-2023-00256629). 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터사업의 연구결과로 수행되었음(IITP-2026-RS-2024-00437718). 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2023-00219107).
접수일자 : 2025년 12월 08일
수정일자 : 2025년 01월 13일
게재확정일 : 2026년 02월 19일
교신저자 : 김수형 e-mail : shkim@jnu.ac.kr

같이 입력 길이와 타이핑 패턴의 변동성이 큰 경우에는 시퀀스 전반의 관계를 동시에 고려하는 데 추가적인 보완이 필요할 수 있다[10]. 이러한 맥락에서, 입력 전체의 상호 관계를 모델링할 수 있는 self-attention 기반 구조가 대안으로 주목받고 있다.

대표적으로 Stragapede et al.이 제안한 TypeFormer는 Temporal·Channel 이중 경로 구조를 통해 복잡한 타이핑 패턴을 시간적 흐름과 특징에 따라 효과적으로 학습하여 모바일 환경에서 우수한 성능을 보고한 바 있다[1].

그러나 TypeFormer에는 몇 가지 중요한 한계가 존재한다. 첫째, 입력 시퀀스의 패딩 구간에 대한 마스킹이 적용되지 않아, 모델이 패딩 값을 실제 입력처럼 처리하게 되는 구조적 결함을 갖고 있다. 이는 self-attention을 사용하는 Transformer 구조에서 성능 저하로 이어질 수 있으며, TypeFormer 논문에서도 이러한 문제 가능성이 명시적으로 언급되어 있다. 둘째, TypeFormer는 모바일 환경에서만 검증되었으며, 데스크톱 키보드 기반 환경에 대한 성능은 보고되지 않았다. 사용자 타이핑 환경의 다양성을 고려하면 데스크톱 기반 분석의 필요성이 크지만, 해당 영역은 기존 연구에서 다루어지지 않았다. 마지막으로, TypeFormer의 학습에는 Triplet Loss가 사용되었는데, Morales et al.의 연구에 따르면 Triplet Loss는 개별 샘플(anchor - positive - negative) 단위로만 비교가 이루어지므로 intra-class 구조를 충분히 반영하지 못하고, negative sampling 품질에 따라 학습 안정성이 크게 좌우된다는 문제가 존재한다[6]. 해당 연구에서는 그러한 한계점을 극복하기 위해 클래스 단위의 구조를 학습하도록 설계한 Set-Margin Triplet Loss(SM-TL)를 제안하였다[6]. SM-TL은 intra-class 응집도를 높이고 inter-class 분리를 강화해, 보다 안정적인 결정 경계를 형성하는 것으로 보고되었다.

이에 본 연구에서는 (1) 기존 TypeFormer가 키보드 환경에서도 일관되게 좋은 성능을 도출하는지를 실험을 통해 검증하였고, TypeFormer의 구조적 한계와 Triplet 기반 거리 학습의 제약을 해결하고자, (2) 부분적인 마스킹 시스템 적용을 통해 성능 차원에서 향상된 TypeFormer 모델을 제안하였으며, (3) Triplet Loss와 SM-TL을 동일 조건에서 비교하여 TypeFormer에 SM-TL이 Triplet Loss에 비해 더욱 효과적임을 입증하였다. 또한 임베딩 공간을 t-SNE로 시각화하여 두 손실 함수가 만들어내는 클래스 구조 차이를 직관적으로 비교한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 정리하고, 3장에서는 제안 모델과 손실 함수를 설명한다. 4장에서는 데이터셋을 소개하고, 5장에서는 실험 설계, 6장에서는 실험 결과를 제시한다. 마지막으로 7장에서 결론 및 향후 연구 방향을 제안한다.

II. 관련 연구

딥러닝이 시계열·행동 데이터에서 우수한 성능을 보이기 시작한 이후, 키스트로크 다이내믹스 분야에서도 다양한 신경망 기반 모델이 도입되었다. 초기 연구에서는 Deep Belief Network(DBN)와 같은 제한된 딥러닝 구조가 활용되었으나[1], 시퀀스 데이터를 처리하기 위해 LSTM·GRU 기반의 순환 신경망(RNN)이 널리 사용되었다. 이러한 접근은 free-text 환경에서도 안정적인 성능을 보인 바 있다. Lu et al[2]. 은 CNN을 이용해 짧은 구간의 타이밍 패턴을 추출하고, RNN(LSTM/GRU)으로 장기 의존성을 학습하는 CNN - RNN 하이브리드 구조를 제안하여 연속 인증 환경에서 기존 기법 대비 향상된 성능을 보고하였다. 그러나 사용자별 독립 모델을 학습하는 구조라는 것과 장기적인 타이핑 패턴의 연속성과 대규모 실제 환경 적용 측면에서 한계를 가진다. 또한 확률 기반 분류 구조를 사용함으로써 임베딩 공간에서의 intra-class 구조를 직접적으로 제어하지 못한다.

이후 Acien et al.의 TypeNet은 대규모 free-text 데이터셋을 활용해 LSTM 기반 모델이 다양한 디바이스 환경(데스크톱·모바일)에서도 높은 일반화 성능을 달성할 수 있음을 보여주었다[3]. 하지만 모든 과거 정보가 단일 상태 벡터로 순차적으로 요약되는 구조적 특성으로 인해, 시퀀스 전반에 분포한 중요 시점 간의 관계를 명시적으로 선택·조절하는 데에는 한계가 존재한다. 더불어 Triplet Loss 기반 학습은 샘플 단위 비교에 의존하므로 사용자 집합 단위의 분포 안정성을 보장하기에는 한계가 있다[6].

최근에는 Transformer 기반 모델의 확산에 따라, 키스트로크 다이내믹스에서도 self-attention 구조를 활용한 연구가 소개되기 시작했다. 대표적으로 Stragapede et al.이 제안한 TypeFormer는 모바일 free-text 환경에서 우수한 성능을 보고하였다[5]. 이는 Li et al[4].의 연구에서의 Temporal·Channel 두 경로로 학습하는 Transformer와 Hutchins et al[7].의 Block-recurrent transformer를 결합하여 성립된 모델로, 시계열 데이터의 연속적 성질과 데이터의 특징을 병렬적으로 학습하고, Transformer의 장기 의존성 처리 능력과 병렬 연산 효율성, 그리고 RNN의 state 정제 능력을 활용하여 구조적 확장 가능성과 표현 능력 측면에서 잠재력을 보여주었다. 그러나 키보드 환경에서의 검증 결과 부재로 인해 TypeFormer가 다양한 도메인에서 일관되게 효과적인 모델일지에 대해서는 알 수 없는 실정이다. 또한, 저자가 직접 언급한 바로는, 데이터를 시퀀스화 하는 과정에서 필연적으로 발생하는 패딩값에 대한 적절한 조치를 취하지 않아 발생하는 성능 저하 문제가 있다. 뿐만 아니라, Triplet Loss가 야기하는 클래스 분리·응집 차원에서의 문제도 존재한다.

앞서 설명한 TypeFormer가 안고 있는 한계점들을 극복하기 위해 본 연구에서 제안하는 방법에 대해서 3장에서 설명한다.

III. 제안방법

1. 마스킹 적용 전략

TypeFormer는 입력 시퀀스를 Temporal module과 Channel module의 두 분기로 분리하여 학습함으로써, 각각의 분기가 서로 다른 관점에서 타이핑 패턴을 해석하도록 구성되어 있다. 전체 구조는 입력 전처리(GRE 및 전치), self-attention 기반 특징 학습, multi-scale CNN/LSTM 처리, 그리고 block-recurrent 기반 상태 전달 기제가 통합된 형태로 구성된다[4].

TypeFormer 모델은 자유 텍스트 기반 키스트로크 시퀀스를 일정한 길이의 입력으로 변환하는 과정에서, 시퀀스 간 길이 차이를 보정하기 위해 제로 패딩을 사용한다. 그러나 원 제안자인 Stragapede et al[5].은 TypeFormer 구조 내에 마스킹 처리 모듈이 존재하지 않으며, 이로 인해 패딩 구간까지 self-attention 연산에 노출되는 구조적 결함이 성능 저하의 원인이 될 수 있음을 언급하였다. 이는 Transformer 기반 구조가 입력 시퀀스의 모든 위치를 동일하게 처리하는 특성상, 의미 없는 패딩 값까지 학습에 활용되는 부작용을 초래할 수 있음을 시사한다.

이러한 구조적 한계를 해결하기 위해, 본 연구에서는 TypeFormer 아키텍처에 마스킹 기법을 도입하였다. 마스킹은 패딩 값이 모델의 각 연산 단계에 영향을 주지 않도록 제어하는 방식으로, 적용 범위에 따라 전체 모델, Temporal 모듈, Channel 모듈로 세분화되었으며, 연산 시점에 따라 Soft 마스킹과 Hard 마스킹으로 구분하여 다양한 조합을 구성하였다.

Soft 마스킹은 self-attention, LSTM, pooling 연산 과정에서 패딩 시점이 모델 출력에 영향을 미치지 않도록 가중치를 조절하는 방식으로, 가변 길이 입력에서 발생하는 패딩 기반 잡음을 완화하는 데 목적이 있다[11].

이를 적용하기 위해 본 연구에서는 입력 시퀀스 내 유효 타임스텝을 구분하는 마스크 $m_{b,t}$를

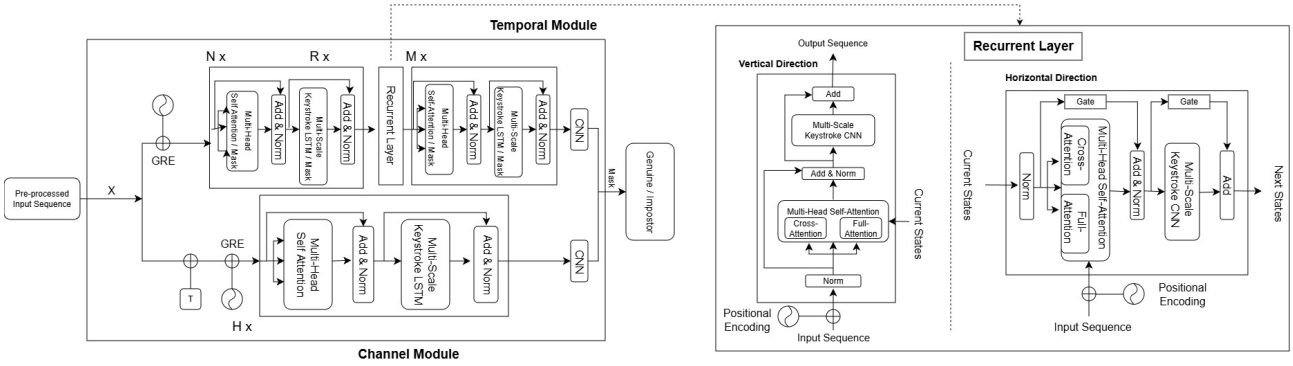


그림 1. Temporal module에 soft masking이 적용된 TypeFormer 아키텍처

정의한다. 배치 b의 시퀀스에서 실제 입력 길이를 ℓ_b라 할 때, 마스크는 다음과 같이 정의된다.

$$m_{b,t} = \begin{cases} 1, t \le \ell_b, \\ 0, t > \ell_b, \end{cases} \quad \ell_b = \sum_{t=1}^{T} m_{b,t} \tag{1}$$

Self-attention에서는 패딩 위치가 key로 선택되지 않도록 attention score에 마스크를 반영한다. 구체적으로, attention 가중치는

$$A_b = softmax\left(\frac{Q_b K_b^T}{\sqrt{d}} + B_b\right) \tag{2}$$

여기서 $B_b(t,s) = -\infty$ if $m_{b,s} = 0$, and 0 otherwise이다.

Recurrent 연산에서는 유효 길이 ℓ_b에 대해서만 LSTM을 적용하며,

$$h_{b,1:\ell_b} = \mathrm{LSTM}(x_{b,1:\ell_b}) \tag{3}$$

이후 출력을 원래 시퀀스 길이 T로 복원한다.

마지막으로 pooling 단계에서는 패딩 시점이 통계 연산에 영향을 주지 않도록

$$\tilde{z}_{b,d,t} = \begin{cases} z_{b,d,t}, m_{b,t} = 1 \\ -\infty \ m_{b,t} = 0 \end{cases} \tag{4}$$

를 적용한 후 global max pooling을 수행한다.

hard 마스킹은 soft 마스킹이 적용된 이후에도 GRE, Positional Encoding, Residual Connection과 같은 연산 과정에서 패딩 시점의 값이 변형되어 이후 계층으로 전파될 수 있다는 점에 착안하여 도입된 방식이다. 최근 연구에서는 패딩 토큰의 영향을 분석하기 위해, 해당 토큰을 의미 정보가 제거된 clean padding 표현으로 치환하여 패딩 시점의 기여를 제거하는 실험적 조작이 사용된 바 있다 [12]. 본 연구에서는 이러한 분석적 관찰에서 착안하여, 일시적인 실험 조작이 아닌 모델 설계 차원의 대응으로서, 각 연산이 완료된 직후 패딩 시점에 해당하는 값을 다시 0으로 재설정하는 hard 마스킹을 적용한다.

hard 마스킹은 다음과 같이 표현할 수 있다.

$$\tilde{x}_{b,t} = m_{b,t} \cdot x_{b,t}, \quad m_{b,t} \in 0,1 \tag{5}$$

여기서 $m_{b,t} = 0$은 패딩 시점을, $m_{b,t} = 1$은 유효한 입력 시점을 의미한다. 이 연산은 각 주요 연산 계층 이후에 적용되며, 패딩 위치에서 발생한 비의미적인 활성값이 이후 계층으로 누적·전파되는 것을 효과적으로 차단한다.

이러한 마스킹 전략은 이후 실험을 통해 free-text 환경에서 가장 효과적인 적용 방식이 분석된다.

2. Set-Margin Triplet Loss (SM-TL)

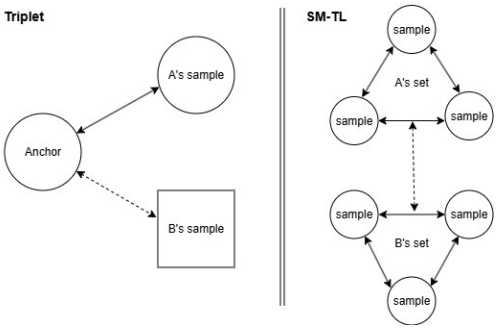


그림 2. Triplet과 SM-TL의 학습 방식 차이

그림 2는 SM-TL이 Triplet과는 다르게 샘플 단위가 아닌 집합 단위로의 학습을 지향한다는 것을 표현하고 있다.

SM-TL의 수식 표현은 다음과 같다.

$$L_{SM-TL} = \sum_{a \in S^+} \max(0, |f(a) - \mu^+|^2 - |f(a) - \mu^-|^2 + \alpha) \quad (6)$$

식 (6)에서 a는 집합 S^+에 속하는 anchor 샘플을 의미하며, μ^+와 μ^-는 각각 anchor와 동일 사용자로 구성된 positive 집합과 타 사용자로 구성된 negative 집합의 중심을 나타낸다. 이는 각 사용자를 하나의 샘플이 아니라 여러 샘플로 구성된 집합으로 바라보는 것으로, 두 사용자에 대응되는 두 집합 $S_i = \{x_1^i \cdots x_G^i\}$와 $S_i = \{x_1^j \cdots x_G^j\}$이 주어졌을 때, SM-TL은 두 집합 내부의 거리 구조를 압축하는 동시에, 두 집합 간 거리를 마진 이상으로 확보하도록 학습을 수행한다. SM-TL은 집합 내부의 샘플 쌍 (x_k^i, x_q^i)과 집합 간 샘플 쌍 (x_k^i, x_l^j)을 기반으로 다수의 Triplet을 구성하여 손실을 계산한다.

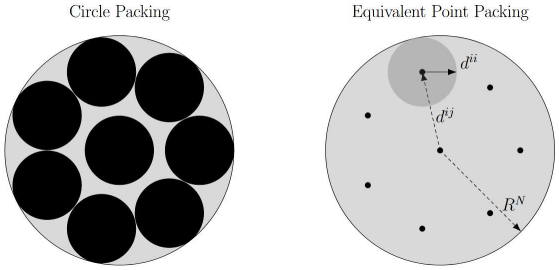


그림 3. Circle Packing의 시각적 설명

이러한 학습 방식은 각 클래스가 응집된 군집을 형성하면서 클래스 간 마진을 유지하는 Circle Packing 개념과 일치한다.

SM-TL 또한 집합 내부의 최대 거리를 줄여 집합을 컴팩트하게 만들고, 서로 다른 집합 간 최소 거리를 넓혀 클래스 간 분리를 극대화한다. 이를 통해 임베딩 공간은 margin-consistent 구조를 형성하며, open-set 환경에서도 새로운 사용자가 안정적으로 배치될 수 있는 일반화 성능을 제공한다.

Ⅳ. 데이터셋

본 연구에서는 자유 텍스트 기반 키스트로크 인증 모델의 일반화 성능을 검증하기 위해 Aalto University에서 공개한 Aalto Desktop DataseT을 사용하였다. 이 데이터셋은 온라인 타이핑 플랫폼을 통해 전 세계 다양한 사용자로부터 자연스럽게 수집된 약 168,000명, 약 1억 3천 6백 85만 건(약 136M)의 키 입력 이벤트로 구성된 대규모 타이핑 데이터셋이다.

표 1. Aalto Desktop Dataset의 구조

Aalto Dataset Example

SENTENCE	PRESS	RELEASE	KEY	KC
We don't recommend McNair this week as a result.	1471934395787	1471934395901	w	87
We don't recommend McNair this week as a result.	1471934395929	1471934396061	e	69
We don't recommend McNair this week as a result.	1471934396096	1471934396208		32
We don't recommend McNair this week as a result.	1471934396286	1471934396422	d	68
We don't recommend McNair this week as a result.	1471934396507	1471934396647	i	73
We don't recommend McNair this week as a result.	1471934397105	1471934397220	n	78
We don't recommend McNair this week as a result.	1471934397233	1471934397337	t	84

표 1은 본 연구에서 사용된 Aalto Desktop Dataset 구조의 일부분을 나타낸다. SENTENCE는 사용자에게 입력하라고 요구되는 문장, PRESS는 사용자가 특정 키를 눌렀을 때의 타임스탬프, RELEASE는 눌린 키가 떼어졌을 때의 타임스탬프, KEY는 눌리거나 떼어진 키의 이름, KC는 Keycode를 의미한다.

사용자들은 웹 인터페이스를 통해 제시된 15개의 영어 문장을 암기하여 전사하는 방식으로 입력하였으며, 각 문장은 Enron Mobile Mail Corpus와 Gigaword Newswire Corpus에서 추출된 1,525개 문장 중 무작위로 선택되었다. 문장들은 최소 세 단어 또는 70자 이상의 길이를 갖도록 구성되었다[13].

모든 입력 이벤트는 key-down 및 key-up 시점을 1ms로 기록하여 키 누름 시간, 키 사이 간격 등 미세한 타이밍 정보를 정밀하게 측정할 수 있다. 또한 각 세션의 오류율이 25% 이상일 경우 데이터 품질 향상을 위해 제외하였다.

Ⅴ. 학습 및 검증 프로토콜

1. 전처리 및 특징 추출

본 연구에서는 TypeNet에서 제안된 절차를 기

반으로 원시 키 입력 로그를 5차원 특징 벡터로 변환하였다. 데이터셋은 PRESS, RELEASE 타임스탬프 및 KEYCODE 정보를 포함하고 있으며, 각 키 입력에 대해 다음 특성을 도출하였다.

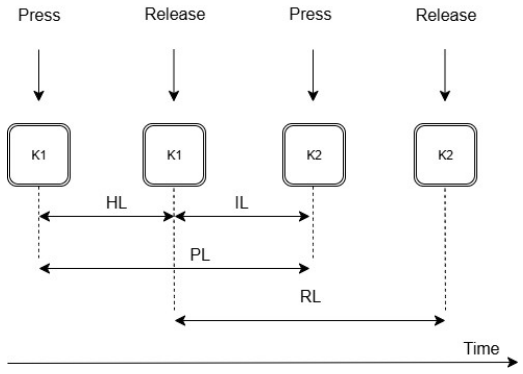


그림 4. 키스트로크 데이터에서 추출 가능한 특징들

- Hold Latency (HL): release - press
- Inter-key Latency (IL): 이전 key의 release와 다음 key의 press 사이의 시간
- Press Latency (PL): 연속된 두 press 사이의 시간
- Release Latency (RL): 연속된 두 release 사이의 시간
- Keycode (KC): ASCII 기반 key ID

시간 기반 특징(HL, IL, PL, RL)은 밀리초(ms) 단위로 측정되므로, 모든 값에 대해 1000으로 나누어 초(s) 단위로 변환하였다.

Keycode 특징은 ASCII 값 범위를 가지므로, 0~1 구간으로 정규화하기 위해 min - max 정규화를 적용하였다. 구체적으로, 각 keycode 값 k에 대해

$$k' = \frac{k - k_{\min}}{k_{\max} - k_{\min}} \quad (7)$$

를 사용하였으며, 여기서 $k_{\min}$과 $k_{\max}$는 각각 최소 및 최대 ASCII 값을 의미한다.

또한, 입력 시퀀스의 길이를 고정하기 위해 모든 시퀀스를 L=50으로 정규화되었는데, 이는 선행연구들로부터의 경험적 근거를 토대로 설정되었다[3, 5, 6]. 길이가 50을 초과하는 시퀀스는 앞에서부터 잘라내고, 50에 미치지 못하는 경우에는 제로 패딩을 적용하였다. 패딩된 시점은 이후 연산에서 마스킹을 통해 제외된다.

2. 학습 절차 및 하이퍼파라미터

본 연구에서 제안하는 모델은 자유 텍스트 키스트로크 시퀀스를 고정 길이의 행렬 입력으로 정규화한 뒤, 이를 거리 기반 비교가 가능한 고정 차원의 임베딩 공간으로 사상하는 방식으로 학습된다.

전처리 과정에서 각 타이핑 이벤트는 HL, IL, PL, RL, Keycode로 구성된 5차원 특징 벡터로 표현되며, 하나의 입력 시퀀스는 길이 L=50의 시계열로 구성된다. 이에 따라 단일 입력은 (L,5) 형태의 행렬로 표현되며, 학습 시에는 이를 배치 크기 B 단위로 묶어 (B,L,5) 형태의 미니배치로 모델에 입력한다. 길이가 L에 미치지 못하는 시퀀스는 제로 패딩으로 보정되며, 해당 패딩 위치는 이후 연산에서 마스킹 처리된다.

모델은 입력 시퀀스 $X \in \mathrm{R}^{\mathrm{L}\times 5}$를 임베딩 함수 $f_\theta(\cdot)$를 통해 고정 차원의 벡터 $z \in R^d$로 변환한다. 여기서 θ는 학습 가능한 모델 파라미터 전체를 의미하며, 본 연구에서는 임베딩 차원을 d=128로 설정하였다.

모든 실험에서 제안 모델(TypeFormer + SM-TL)은 동일한 하이퍼파라미터 설정으로 학습하였다. 최적화 알고리즘으로는 Adam을 사용하였으며, 초기 학습률은 1×10^{-3}으로 두었다. 학습은 총 250 에폭(epoch)에 걸쳐 수행되었고, 각 에폭은 150개의 미니배치(mini-batch) 업데이트로 구성되도록 정의하여, 데이터 양에 따라 에폭 길이가 변하지 않도록 고정하였다.

손실 함수인 Set-Margin Triplet Loss의 마진 상수 α는 선행 연구에 근거해 초기에는 1.5로 설정하였으나, 이후 ablation study를 통해 키보드 환경에서의 TypeFormer는 마진 값을 1.4로 설정하였을 때 가장 좋은 성능을 도출한다는 것을 확인하였다. 이에 대해서는 6장에서 보다 자세히 설명하였다.

학습률 스케줄링은 검증 세트에서의 EER(Equal Error Rate) 변화를 기준으로 동적으로 조정하였다. 일정 에폭 동안 검증 EER이 더 이상 개선되지 않을 경우, 현재 학습률을 절반으로 감소시키는 방식으로 세밀한 수렴을 유도하였다.

훈련 종료 후에는 검증 세트에서 가장 낮은 EER을 기록한 시점의 모델 파라미터를 최종 모델로 선택하였다. 이후의 모든 성능 평가는 이 최종 모델을 기준으로 수행하여, 비교 실험 간 일관성을 확보하였다.

본 연구는 실험을 위해서 학습에 68,000명, 검증에 400명, 테스트에 1,000명의 데이터를 사용하였다.

3. 실험 설계: 인증 프로토콜

인증 실험은 TypeNet에서 사용된 프로토콜을 기반으로 구성하였다. 테스트 단계에는 학습 과정에서 등장하지 않은 사용자만 포함되며, 이는 실제 서비스 환경과 동일한 open-set 인증 시나리오를 재현하기 위함이다.

가. 등록(Enrollment) 단계

각 사용자는 사전에 일정 개수의 입력 시퀀스를 제공하여 갤러리 세트(galleryset)를 형성한다. 본 연구에서는 등록 시퀀스 수를 1회, 2회, 5회, 7회, 10회로 변화시키며 성능 변화를 분석하였다. 등록 시퀀스의 개수는 실제 시스템에서 사용자가 최초 로그인 시, 몇 번의 키 입력을 요구받는가에 대응되는 중요한 설계 변수이다.

나. 질의(Query) 단계

평가 시에는 각 사용자로부터 별도로 수집된 테스트 시퀀스를 질의로 사용한다. 질의 시퀀스는 해당 사용자의 갤러리와 비교하여 본인 여부를 판별하는 데 활용된다. 본 연구에서는 모든 사용자에 대해 동일한 개수(예: 5개)의 질의 시퀀스를 배정하여 통계적 일관성을 확보하였다.

질의 데이터는 등록 데이터와는 다른 수집 구간에서 분리 추출하였다.

다. 스코어링 및 판별 방식

각 질의 시퀀스는 동일 사용자의 등록 시퀀스들과 임베딩 공간에서 유클리드 거리를 계산하여 유사도를 평가한다. 한 사용자의 등록 데이터가 n_G개, 질의 데이터가 n_P개인 경우, 각 질의 시퀀스는 모든 등록 시퀀스와의 거리를 계산한 뒤 이를 평균하여 하나의 점수를 산출하며, 이 과정을 통해 사용자별로 n_P개의 Genuine 점수가 생성된다.

Impostor 점수는 특정 사용자를 기준으로, 해당 사용자를 제외한 모든 타 사용자의 질의 시퀀스를 이용하여 계산된다. 이때 각 타 사용자로부터 하나의 질의 시퀀스를 선택하고, 이를 검증 대상자의 등록 데이터 n_G개와 각각 거리 계산한 후 평균하여 Impostor 점수를 산출한다. 본 연구에서는 이러한 스코어링 방식을 TypeNet의 평가 프로토콜과 동일하게 적용하여, 모델 간 직접적인 성능 비교가 가능하도록 하였다.

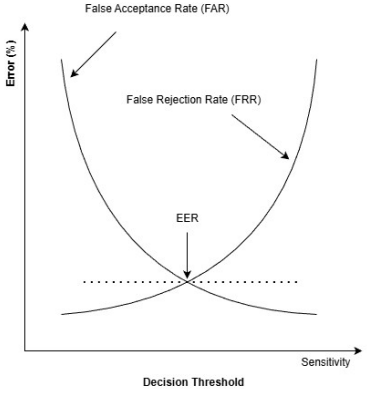


그림 5. EER 개념 도식화

그림 5에는 본 연구에서 사용한 Equal Error Rate(EER) 성능 지표에 대한 개념이 도식화되어 있는데, 앞서 설명한 스코어 산출 이후, Genuine 점수 분포와 Impostor 점수 분포를 구성하고 임계값을 변화시키며 FAR과 FRR을 계산한다. 두 오차율이 일치하는 지점의 값은 EER로 정의되며, 본 연구에서는 사용자별로 계산된

EER을 평균하여 전체 시스템의 최종 성능을 평가하였다.

VI. 실험 및 결과

1. 마스킹 적용에 따른 성능 비교

표 2. 마스킹 적용에 따른 성능 변화 비교표. 표 내의 값=EER, 값의 단위=%, E=등록 데이터 개수

Mask	E=1	E=2	E=5	E=7	E=10
미적용	9.58	7.49	6.20	5.88	5.90
Whole	9.49	7.38	6.06	5.87	5.89
soft Temporal	4.67	3.01	2.18	2.22	2.13
hard Temporal	4.81	3.20	2.37	2.40	2.32
soft Channel	9.69	7.66	6.27	5.97	6.07
hard Channel	9.59	7.47	6.12	5.93	5.73

TypeFormer는 시퀀스를 고정 길이로 변환하는 과정에서 패딩이 포함되며, 이 패딩이 모델 내부 계산에 노출될 경우 성능 저하를 유발할 수 있다. 이를 해결하기 위해 본 연구에서는 Soft 마스킹과 Hard 마스킹 전략을 도입하여 그 효과를 비교하였다.

실험 결과, 마스킹 적용 방식 및 적용 위치에 따라 성능 차이가 뚜렷하게 나타났다.

먼저, 마스킹을 전혀 적용하지 않았을 때와 모델 전 구간에 일괄적으로 적용했을 때(E=1일 때, EER 9.58 → 9.49)는 성능 차이가 매우 미미하여, 단순한 마스킹 적용만으로는 실질적인 효과를 기대하기 어려움을 보여준다.

반면, Channel 모듈에만 마스킹을 적용한 경우 오히려 오류율이 증가하였다. soft 방식의 경우 EER이 9.69로 비적용 조건보다도 높았으며, hard 방식 또한 9.59로 비슷한 경향을 보였다. 이는 Channel 영역에 대한 마스킹이 모델의 핵심적인 특징 학습을 방해하거나, 불필요한 정보 제거로 이어졌을 가능성을 시사한다.

반대로, Temporal 모듈에 마스킹을 국한하여 적용한 실험에서는 매우 유의미한 성능 향상이 관측되었다. soft 방식의 경우 EER이 4.67(E=1 기준)로 크게 낮아졌으며, hard 방식 역시 4.81로 전 구간 혹은 Channel 모듈 적용 대비 뚜렷한 개선을 보였다. 특히 soft 방식이 모든 E 값에서 가장 낮은 오류율을 기록하며 가장 우수한 결과를 나타냈다.

이러한 결과는 Temporal 모듈에 대한 soft 마스킹이 free-text 키스트로크 인증에서 가장 효과적인 전략임을 시사한다.

2. TypeNet과 TypeFormer 비교

표 3. TypeNet vs TypeFormer (Triplet and SM-TL). 표 내의 값=EER, 값의 단위=%, E=등록 데이터 개수

Model	E=1	E=2	E=5	E=7	E=10
TypeNet	5.40	3.60	2.20	1.80	1.60
TypeFormer with Triplet	11.28	9.09	7.90	7.83	7.88
TypeFormer with SM-TL	4.41	2.87	2.14	2.06	2.10

기존 TypeFormer 논문은 모바일 환경에서만 TypeNet과 성능 비교를 수행하였으며, 실제 키보드 기반 입력에 해당하는 데스크톱 free-text 환경에서는 충분히 검증되지 않은 상태였다. 본 연구는 동일한 Aalto Desktop Dataset과 인증 프로토콜을 적용해 기존 TypeFormer(Triplet Loss), 개선된 TypeFormer(SM-TL), 그리고 TypeNet을 공정하게 비교하였다.

먼저 TypeNet과 기존 TypeFormer(Triplet Loss)를 비교한 결과, 키보드 환경에서는 기존 TypeFormer가 패딩 처리 부재 및 손실 함수 구조적 한계로 인해 TypeNet 대비 불안정한 성능을 보였다. 이는 기존 TypeFormer가 모바일 기반 데이터 특성에 최적화된 구조였음을 의미한다.

이어 개선된 TypeFormer + SM-TL을 평가한 결과, 등록 시퀀스 개수 E=1, 2, 5 구간에서 TypeNet과 기존 TypeFormer 대비 일관적으로 낮은 EER을 기록하였다. 특히 E가 작아(1~2개) 등록 정보가 부족한 환경에서조차 개선된 TypeFormer는 강한 구별 능력을 유지하였으며,

이는 SM-TL이 사용자별 분포적 변동성을 적절히 통제하고 클래스 간 마진 구조를 안정적으로 확보한 결과로 해석된다.

종합적으로, 기존 TypeFormer가 모바일 환경에서만 검증되었다는 한계를 넘어, 개선된 TypeFormer + SM-TL은 키보드 환경에서도 강력한 인증 모델로 기능함을 실험적으로 확인하였다.

3. 마진 값 변화에 따른 성능 분석

표 4. 마진 값에 따른 성능 변화 비교표. 표 내의 값=EER, 값의 단위=%, E=등록 데이터 개수

Margin	E=1	E=2	E=5	E=7	E=10
2.0	9.70	7.68	6.24	6.02	5.97
1.6	4.70	3.12	2.43	2.35	2.19
1.5	4.67	3.01	2.18	2.22	2.13
1.4	4.41	2.87	2.14	2.06	2.10
1.3	4.68	3.26	2.48	2.27	2.19

SM-TL의 핵심 제어 요소는 클래스 중심 간 최소 분리 거리를 결정하는 마진 값이다. 본 연구에서는 다양한 마진 값을 변화시키며 SM-TL의 민감도를 분석하였다.

실험 결과, 마진 값이 지나치게 작을 경우 genuine - impostor 간 경계가 충분히 확보되지 않아 EER이 증가했으며, 반대로 마진 값을 지나치게 크게 설정하면 negative 클래스 분리가 과도해져 학습이 불안정해지는 경향을 확인하였다.

가장 낮은 EER은 margin=1.4에서 관찰되었으며, 이를 통해 SM-TL의 마진 값은 단순한 튜닝 변수라기보다 임베딩 공간 구조를 결정하는 핵심 파라미터임을 확인할 수 있었다.

VII. 결론 및 향후 연구

본 연구에서는 자유 텍스트 기반 키스트로크 다이내믹스 인증에서 TypeFormer 모델이 갖는 구조적 한계를 재검토하고, 이를 보완하기 위한 세 가지 측면의 개선 방향을 제시하였다. 첫째, 패딩 처리 부재로 인해 발생할 수 있는 비의미적 활성값 전파 문제를 해결하기 위해 soft 마스킹을 도입함으로써, 가변 길이 입력 환경에서도 모델이 패딩 시점을 안정적으로 처리할 수 있음을 보였다. 이러한 결과는 자유 텍스트 환경의 시계열 인증 모델 설계에서 패딩 처리 및 마스킹 전략이 모델 안정성과 성능에 중요한 요소로 작용할 수 있음을 시사한다.

둘째, Triplet Loss와 Set-Margin Triplet Loss(SM-TL)를 동일한 조건에서 비교 분석함으로써, 클래스 집합 단위의 마진을 고려하는 SM-TL이 소수 등록 샘플 조건에서도 보다 안정적인 성능을 도출한다는 것을 확인하였다.

셋째, Aalto Desktop Dataset을 활용하여 TypeFormer를 데스크톱 환경에 직접 적용한 결과, 기존 구조를 그대로 사용할 경우 전반적인 성능 저하가 발생함을 확인하였다. 또한 마스킹을 적용하더라도 이러한 성능 저하를 충분히 해소하기 어려웠다.

이후 손실 함수를 Set-Margin Triplet Loss(SM-TL)로 대체하여 실험을 수행한 결과, 마스킹과 SM-TL을 함께 적용한 TypeFormer는 데스크톱 환경에서 기존 TypeNet과의 비교에서 등록 시퀀스 수가 제한적인 조건(E=1~5)에서 더 우수한 성능을 달성하였다. 이러한 결과는 Transformer 기반 키스트로크 인증 모델이 충분한 등록 데이터가 확보되지 않은 현실적인 환경에서도 경쟁력 있는 성능을 발휘할 수 있음을 시사하며, 손실 함수 설계가 데스크톱 키보드 환경에서의 성능 안정성에 중요한 역할을 함을 보여준다.

그러나 본 연구에서의 단일 데이터셋에서 검증된 성능을 통해 제안하는 방법론이 다양한 데이터셋에서 일관된 성능을 도출할 수 있다고 단언하기에는 어려움이 있다. 따라서 모델의 일반화 가능성을 논하기 위해 Clarkson II, Buffalo 등 다양한 데이터셋에서의 추가적인 검증 과정이 이루어져야 한다.

REFERENCES

[1] Deng, Y., & Zhong, Y. Keystroke dynamics user authentication based on gaussian mixture model and deep belief nets. *International Scholarly Research Notices*, 2013(1), 565183. (2013).

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

[3] Acien, A., Morales, A., Monaco, J. V., Vera-Rodriguez, R., & Fierrez, J. TypeNet: Deep learning keystroke biometrics. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 1, pp. 57-70, 2021.

[4] Li, B., Cui, W., Wang, W., Zhang, L., Chen, Z., & Wu, M. Two-stream convolution augmented transformer for human activity recognition. In *Proceedings of the AAAI conference on artificial intelligence,* Vol. 35, No. 1, pp. 286-293, May, 2021.

[5] Stragapede, G., Delgado-Santos, P., Tolosana, R., Vera-Rodriguez, R., Guest, R., & Morales, A. TypeFormer: Transformers for mobile keystroke biometrics. *Neural Computing and Applications*, vol. 36, no. 29, pp. 18531-18545, 2024.

[6] Morales, A., Fierrez, J., Acien, A., Tolosana, R., & Serna, I. SetMargin loss applied to deep keystroke biometrics with circle packing interpretation. *Pattern Recognition*, 122, 108283. 2022.

[7] Hutchins, D., Schlag, I., Wu, Y., Dyer, E., & Neyshabur, B. Block-recurrent transformers. *Advances in neural information processing systems*, 35, 33248-33261. 2022.

[8] Shadman, R., Wahab, A. A., Manno, M., Lukaszewski, M., Hou, D., & Hussain, F. Keystroke dynamics: Concepts, techniques, and applications. *ACM Computing Surveys*, vol . 57, no. 11, pp. 1-35, 2025.

[9] Li, J., Chang, H. C., & Stamp, M. Free-text keystroke dynamics for user authentication. In *Artificial Intelligence for Cybersecurity,* Cham: Springer International Publishing, pp. 357-380, 2022.

[10] Stragapede, G., Delgado-Santos, P., Tolosana, R., Vera-Rodriguez, R., Guest, R., & Morales, A. Mobile keystroke biometrics using transformers. In *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG),* IEEE. pp. 1-6, Jan. 2023.

[11] Wu, Y., Fang, K., Zhang, D., Wang, H., Zhang, H., & Chen, G. TLM: token-level masking for transformers. arXiv preprint arXiv:2310.18738. 2023.

[12] Toker, M., Galil, I., Orgad, H., Gal, R., Tewel, Y., Chechik, G., & Belinkov, Y. Padding tone: A mechanistic analysis of padding tokens in t2i models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies,* Vol. 1: Long Papers, pp. 7618-7632, Apr. 2025.

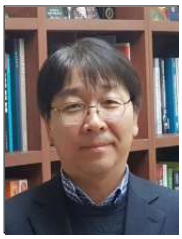
[13] Dhakal, V., Feit, A. M., Kristensson, P. O., & Oulasvirta, A. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI conference on human factors in computing systems,* pp. 1-12, Apr. 2018.

저 자 소 개

서성찬(준회원)

2020년 감리교신학대학교 종교철학과 학사 졸업.
2022~현재 전남대학교 정보보안융합학과 석사 재학

<주관심분야 : 정보보안, 인공지능 기반 보안, 패턴인식>



김수형(종신회원)

1986년 서울대학교 컴퓨터공학과 학사 졸업.
1988년 KAIST 전산학과 석사 졸업.
1993년 KAIST 전산학과 박사 졸업.
1997년~현재 전남대학교 AI융합대학 인공지능학부 교수

<주관심분야 : 인공지능, 자연영상 패턴인식, 감정인식, 정밀의료, 문서영상처리>