International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 16, Number 2, pages 242–271, June 2026

Large-Scale Study of Controller Operation Dynamics for Player Authentication in Fighting Games[1]

Takeshi Kawamoto
Graduate School of Information Science and Technology, The University of Tokyo
Bunkyo-ku, Tokyo, 113-8654, Japan
kawamoto@yamagula.ic.i.u-tokyo.ac.jp

Maharage Nisansala Sevwandi Perera
Graduate School of Information Science and Technology, The University of Tokyo
Bunkyo-ku, Tokyo, 113-8654, Japan
perera.nisansala@yamagula.ic.i.u-tokyo.ac.jp

Franziska Zimmer
Graduate School of Information Science and Technology, The University of Tokyo
Bunkyo-ku, Tokyo, 113-8654, Japan
zimmer@yamagula.ic.i.u-tokyo.ac.jp

Ryosuke Kobayashi
Graduate School of Information Science and Technology, The University of Tokyo
Bunkyo-ku, Tokyo, 113-8654, Japan
kobayashi@yamagula.ic.i.u-tokyo.ac.jp

Mhd Irvan
Graduate School of Information Science and Technology, The University of Tokyo
Bunkyo-ku, Tokyo, 113-8654, Japan
irvan@yamagula.ic.i.u-tokyo.ac.jp

Rie Shigetomi Yamaguchi
Graduate School of Information Science and Technology, The University of Tokyo
Bunkyo-ku, Tokyo, 113-8654, Japan
yamaguchi.rie@i.u-tokyo.ac.jp

Received: February 20, 2026
Revised: May 5, 2026
Accepted: June 3, 2026
Communicated by Toru Nakanishi

[1]This paper is the extended version of our work presented at The Thirteenth International Symposium on Computing and Networking Workshops (CANDARW).

Abstract

Player authentication is crucial for preserving the integrity of esports competitions, yet current measures cannot entirely prevent impersonation via collusion. While behavioral biometrics offer a promising solution, their application has been largely limited to continuous input devices and homogeneous gameplay states. Taking fighting games as an ideal domain for addressing these gaps, we propose a method that applies keystroke-authentication principles to controller operation dynamics and introduce density-based segmentation to capture context-dependent operation patterns. Experimental results using data from 307 matches by 60 players in *Street Fighter 6* demonstrate the effectiveness of our approach, achieving a PR-AUC of 55.7 % and an EER of 12.5 %. Further analysis revealed that, aggregating decisions using 30 seconds of controller operations, our method can achieve a PR-AUC higher than 90 % and EER lower than 1 %. These findings validate the applicability of keystroke-dynamics principles to controller dynamics and establish an interpretable baseline.

Keywords: esports, player authentication, behavioral biometrics, keystroke dynamics

1 Introduction

Esports has witnessed rapid growth in recent years, evolving into a mainstream form of competitive entertainment with a global audience and substantial economic impact. The gaming industry, including esports, has continued to expand, with the global esports market estimated to reach USD 1.72 billion in 2024 [14]. Furthermore, the number of esports tournaments has been increasing; for example, one-quarter of competitions held in Japan offered prize money [3]. These developments not only reflect growing public interest in esports but also emphasize the need for fairness and competitive integrity in tournaments. Meanwhile, the nature of esports allows for remote participation via the internet, leading to an increase in online tournaments alongside traditional in-person events [3]. Consequently, the esports industry would face the challenge of ensuring competitive integrity while operating within a remote, online environment.

Among various dishonest practices, including those related to technology, players, and organizations [10], player impersonation has emerged as a notable concern in online esports competitions. In this context, impersonation refers to the act of substituting a registered player with another individual. A particular challenge in esports is cooperative impersonation [18], in which the account owner and a substitute player collude, because such collusion is difficult to detect through conventional means. To prevent such cases, organizers typically require verification of account ownership or real-time video feeds of competitors [23]. However, these methods fail in cooperative impersonation scenarios: account credentials can be shared; a video feed cannot guarantee that the person on camera is the actual participant. Given these challenges, the community could benefit from additional security measures. One such measure involves using behavioral logs as credentials, which has been considered a promising approach for authentication, as such logs are hard to forge or transfer.

Existing work has utilized players' behavioral biometrics for authentication across various game genres and modalities, such as open-world games using mouse movements [25], strategy games using mouse and keyboard inputs [30], rhythm games using virtual reality (VR) controller motions [20], and mobile games using touchscreen operations [28]. While these works demonstrated the feasibility of behavioral authentication in games, their scope is limited in two aspects: input devices and data collection situations. First, with respect to input devices, most studies primarily rely on continuous, high-degree-of-freedom interfaces such as a mouse and VR controller, whereas discrete input devices, particularly keyboards, are still underexplored as a source of behavioral biometrics. Second, regarding data collection settings, many studies focus on highly homogeneous behaviors captured during specific tasks, failing to account for the dynamic nature of real-world gameplay, where users frequently transition between diverse states, such as combat and idle periods.

We argue that the fighting game is a promising domain for addressing these gaps. This genre is characterized by fast-paced, expertise-driven actions, yielding distinctive behavioral patterns that are well-suited for authentication. To perform these operations, players use dedicated game controllers, which generate discrete input data, making this genre an ideal candidate for discrete-input behavioral

authentication. In addition, fighting games feature rapid shifts between in-combat and idle periods, which enables us to investigate how different gameplay contexts affect the behavioral signals used for authentication.

In this paper, we propose a player-authentication method for fighting games based on controller operations. To address the issue in input devices, we draw an analogy between controller buttons and keyboard keystrokes, adapting feature-engineering techniques and modeling approaches from keystroke authentication to controller inputs. For the challenge of dynamic context transitions, we are motivated by the notion that separating different gameplay phases can enhance the authentication performance and introduce a context-aware segmentation method.

Our work proceeds in two stages: proof of concept and practical method development. In the proof-of-concept stage, we first verify the validity of the keystroke-based method in controller inputs and separation of gameplays based on the scenes using game-state information. For the evaluation, we collected controller input logs from 99 matches played by nine professional players of the fighting game *Street Fighter 6*. The results demonstrated the effectiveness of our approach, achieving an area under the precision-recall curve (PR-AUC) of 75.7 % and an equal error rate (EER) of 13.8 %. Moreover, we found that the biometric signal is strongest during in-combat periods, where the controller-operation density is higher, which supports our assumption of improvement by scene-wise segmentation.

Based on these findings, we evolved our method by introducing density-based segmentation, which identifies in-combat segments based on the density of controller operations, without relying on explicit scene labels that may not be available in real-world applications. We demonstrated that this method achieved performance comparable to the scene-based segmentation, with 73.0 % PR-AUC and 12.8 % EER. Furthermore, using a larger dataset of 307 matches from 60 players, we assessed the scalability of our approach and achieved 55.7 % PR-AUC and 12.5 % EER. Our additional discussion revealed that aggregating decisions using 30 seconds of controller operations can further enhance the performance, achieving a PR-AUC higher than 90 % and EER lower than 1 %, within the studied population sizes. These results demonstrate the feasibility of player authentication using controller operations only and establish an interpretable baseline for future research in this domain.

Contributions

Our key contributions are as follows:

1. We establish an interpretable baseline method for controller-dynamics authentication in fighting games by adapting keystroke-dynamics techniques and context-aware segmentation (scene-based and density-based). Our approach can serve as a player verifier if only the controller input logs are available.
2. We demonstrate the effectiveness of our approach by collecting and evaluating on two datasets of professional players recorded in real gameplay of *Street Fighter 6*.
3. We analyze practical considerations for deployment from latency and scalability perspectives, notably showing that, regardless of player population size available in this study, our method can achieve reasonable performance with 30-second controller operations.

Organization

The rest of this paper is organized as follows. Section 2 introduces the background of behavioral authentication as fundamentals and reviews its application in esports. Section 3 demonstrates the research gap in existing work and identifies the problem of player authentication in fighting games, and outlines the challenges involved. Section 4 describes the proposed scene-based segmentation method and its evaluation as a proof of concept. Based on the findings in Section 4, we propose a density-based segmentation method in Section 5 and evaluate its performance. Section 6 discusses the comparison of the two methods, and their applicability in actual cases, from latency and scalability perspectives, followed by the acknowledgment of limitations and potential future work. Finally, Section 7 concludes the paper and summarizes our contributions.

2 Related Work

This section first overviews the fundamentals of behavioral authentication (Section 2.1), followed by its representative modalities using computer interactions (Section 2.2). Then it reviews prior works on behavioral authentication in esports using players' operations, and summarizes them from the perspectives of input devices and data collection (Section 2.3).

2.1 Behavioral Authentication

Behavioral authentication is one type of biometric authentication that verifies users based on their distinctive behavioral traits [29]. This section outlines the principles and advantages of biometric authentication.

Traditional authentication methods have relied on knowledge-based approaches, such as passwords, and possession-based approaches, such as security tokens, but these methods are vulnerable to forgetting or theft. Biometric authentication, in contrast, uses unique physical characteristics of individuals, mitigating the risks of conventional methods. Static biometric modalities, such as fingerprint, face, and iris recognition, have achieved high accuracy and practical deployment [16]. However, static biometrics face challenges such as vulnerability to leakage, the need for additional hardware for acquisition, and unsuitability for continuous authentication due to the requirement for explicit user actions [13, 16].

Behavioral authentication, which utilizes dynamic biometric traits, has been proposed to address these issues. Behavioral traits include gait [6], signature [9], keystroke [7], mouse movements [22], VR controller motions [21], and touchscreen operations [15]. In particular, behavioral biometrics in computer interactions can be broadly categorized into discrete inputs, such as keystroke, and continuous inputs, such as mouse and touchscreen operations.

One of the main advantages of behavioral authentication is that it can collect features necessary for authentication from normal user operations. This eliminates the need for additional sensors or devices, reducing implementation costs. Furthermore, this nature of behavioral authentication makes it suitable for continuous authentication, which can verify user identity in real-time and in the background. By performing initial authentication through passwords or static biometrics, and then continuously verifying user identity thereafter, it can detect impersonation or takeover of the authenticated user [15].

2.2 Behavioral Authentication using Computer Interactions

The following section reviews behavioral-authentication modalities in computer interactions. As mentioned above, behavioral biometrics in computer interactions can be categorized into discrete- and continuous-input modalities.

As a representative discrete-input modality, keystroke dynamics authenticates users by analyzing typing patterns on keyboards [7]. It converts sequences of key-types into temporal features such as key-hold times and inter-key intervals, and models their statistical properties. These features provide a unique behavioral signature for each user, and are fed into various machine-learning models such as Random Forest [5] and Support Vector Machine [12]. Instead of calculating temporal features, deep-learning models have also been applied directly to raw keystroke data [1].

On the other hand, mouse dynamics distinguishes users based on mouse movements, such as trajectory, speed, and click duration [22]. In contrast to keystroke dynamics, mouse dynamics captures continuous motion data, and they can generate a variety of features such as velocity, acceleration, and curvature of mouse movements. They can be modeled by various machine-learning techniques [4, 19]. Similarly, VR device motions [21] and touchscreen operations [15] also provide continuous input data as mouse movements, and they are also modeled using similar approaches.

2.3 Behavioral Authentication in Esports

Building on the principles of behavioral authentication discussed above, the following section explores its application to player authentication in video games. We review prior works on behavioral

Table 1: **Behavioral authentication studies in esports.** The performance metric ACC and EER represent the accuracy and equal error rate, respectively.

Game title	Input device	Model	#Players	Performance
League of Legends [8]	Mouse, keyboard	Random Forest	55	ACC 91 %
League of Legends [26]	Mouse, keyboard	Multilayer Perceptron	56	ACC 86 %
Dota 2 [30]	Mouse	Random Forest	93	ACC 95 %
Minecraft [25]	Mouse	LightGBM	10	ACC 92 %
Beat Saber [20]	VR controller	LightGBM	55,541	ACC 94 %
Angry Birds [28]	Touchscreen	Support Vector Machine	30	EER 10 %

authentication in video games using players' operations by game genres, and then summarize them from the perspectives of input devices and data collection.

2.3.1 Prior Studies by Game Genre

Multiplayer online battle arena (MOBA) games are a genre where multiple players are divided into two teams to compete against each other. This genre is characterized by the combination of mouse-based movement commands and keyboard-based skill activations. As a popular genre in esports, games such as *League of Legends* [8, 26] and *Dota 2* [30] have been studied for behavioral authentication. In *League of Legends*, multimodal behavioral authentication using keyboard and mouse operations has been proposed. From keyboard operations, features such as typing speed of frequently used keys were extracted, and from mouse operations, features like speed and click frequency were used to build models based on Random Forest [8] and Multilayer Perceptron [26]. In the former, using data from 55 players, the accuracy was 28 % for keyboard operations alone and 85 % for mouse operations alone, but combining them improved accuracy to 91 %. The latter successfully identified 56 players with an accuracy of 86 %. In *Dota 2*, mouse operations and match statistics were used as behavioral features to represent players [30]. Physical quantities such as angle, speed, and acceleration were used as features, along with match statistics like kill count, death count, and item purchase and placement counts. Using Random Forest as the model, in a task to verify players from 93 individuals, an accuracy of 95 % was achieved using statistical information and mouse operations.

Open-world games are a genre where players freely explore vast virtual spaces and complete various missions. A representative title is *Minecraft*, where user identification using mouse operation logs has been studied [25]. Physical quantities similar to those in mouse dynamics were used as features, and LightGBM was employed as the model. As a result, it successfully verified 10 players with an accuracy of 92 % and an EER of 0.18 % from their 20 minutes of play.

VR games are a genre where players operate avatars in VR spaces, and a large-scale player authentication study using play data from the rhythm game *Beat Saber* has been conducted [20]. *Beat Saber* is a game where players slash incoming objects with swords in sync with music in a VR space. In this study, physical quantities from players' hand and head movements, as well as state quantities indicating object types and whether they were slashed, were used as features to build a model that identifies players using a LightGBM-based classifier. As a result, it successfully identified players with an accuracy of 94 % from a pool of 55,541 players using 100 seconds of play.

Smartphones are also widely used as gaming platforms. A study attempted behavioral authentication using three different game genres: *Angry Birds*, *Flow Free*, and *Fruit Ninja* [28]. From touchscreen operations in these games, physical quantities such as angle, speed, and acceleration were derived, similar to mouse operations, along with information like touch contact area and finger orientation used as features. Using Support Vector Machine, the study evaluated a task to identify players from a pool of 30 individuals, achieving an area under the receiver operating characteristic curve (ROC-AUC) of 96 % and an EER of 10 % in *Angry Birds*.

2.3.2 Cross-study Summary and Takeaways

Table 1 summarizes the aforementioned works on behavioral authentication using operations in esports. From the perspective of input devices, prior studies have primarily focused on input devices that produce continuous, high-dimensional data, such as mice, touchscreens, and VR controllers, where rich behavioral patterns can be captured by applying their corresponding principles of behavioral biometrics.

Furthermore, from the viewpoint of data collection contexts, it is evident that authentication performance is enhanced by acquiring data with minimal variance in player behavioral traits. Genres such as MOBAs and open-world games, where a single session can last from 30 minutes to over an hour, are considered ideal environments for data collection and authentication due to the large amount of data collected in a single session. Conversely, while VR and mobile games often have shorter play cycles (typically five minutes or less), researchers achieve homogeneous behavioral data by specifically targeting data collection during assigned tasks across multiple sessions.

3 Our Approach

In previous sections, we discussed the necessity and challenges of player authentication in esports. In this section, we outline our approach to address these challenges. We first identify the research gap in existing studies on behavioral authentication in esports (Section 3.1), and then explain why fighting games are a suitable research target (Section 3.2). Next, we describe our research direction to fill the identified gap (Section 3.3). Finally, we formalize the specific task setting for player authentication in this context (Section 3.4).

3.1 Research Gap

In the previous section, we pointed out that behavioral authentication is one of the promising approaches to verify esports players continuously, using information that is difficult to share or impersonate. We also reviewed related works that applied behavioral authentication to esports, focusing on input devices and data collection.

Building upon these discussions, we argue that there is a research gap in applying behavioral authentication to esports from the two perspectives mentioned above. From the perspective of input devices, prior studies have primarily focused on input devices that produce continuous, high-dimensional data, such as mice [25], touchscreens [28], and VR controllers [20], where rich behavioral patterns can be captured. In contrast, devices with discrete inputs like keyboards have primarily been utilized as part of multimodal systems, and their standalone use did not achieve sufficient performance for authentication [8]. However, there are some occasions where only discrete input devices are available, and thus it is important to investigate their potential for behavioral authentication. Regarding data collection, existing studies have mainly focused on scenarios where players are engaged in specific tasks, resulting in highly homogeneous gameplay states. Though this setting simplifies the modeling and provides stable performance, it does not reflect the actual gameplay in esports, where players frequently transition between different states, such as combat and idle periods.

3.2 Fighting Games as a Research Target

We identify the fighting game as both one of the promising candidates for behavioral authentication and a genre that presents unique challenges from the two perspectives above. In this section, we describe the characteristics of fighting games that make them suitable for behavioral authentication and the challenges they pose.

Fighting games are one-on-one competitive games where players control their characters to attack, move, and defend. Most fighting games adopt a round-based system, with rounds and battles as the units of competition. Players win a round by reducing the opponent's health to zero, and the first player to win two rounds wins the battle; one battle consists of up to three rounds. A round

typically lasts between 30 and 60 seconds, with a short break between the end of one round and the start of the next.

Fighting games are promising candidates for behavioral authentication due to the requirement for players to perform fast and precise inputs. Executing attacks in fighting games often involves performing a series of actions, known as combos, by pressing multiple buttons in a specific order and timing. The input timing for combos is extremely tight, with allowable intervals between button presses sometimes as short as one frame (1/60th of a second). Consequently, player inputs in fighting games are expected to be faster and more precise compared to other game genres. This characteristic suggests a strong potential for individual behavioral traits to manifest in player inputs.

Nonetheless, applying behavioral authentication to fighting games presents two main challenges unaddressed in existing studies. From the perspective of input devices, fighting games utilize specialized controllers that provide discrete inputs constrained to a limited set of buttons. This makes it difficult to simply apply existing methods designed for continuous, high-dimensional input devices. As for the aspect of data collection, the round-based structure of fighting games leads to frequent scene transitions between in-combat and idle periods within a single battle, unlike other game genres where players may remain in a single state for extended periods. These two points make player authentication using controller inputs in fighting games a novel area within behavioral authentication research in esports.

3.3 Research Direction

We have identified two main challenges in applying behavioral authentication to fighting games so far. Here, we describe our research direction to address these challenges and demonstrate how our work proceeds.

3.3.1 Idea

For the first challenge, we apply the principles of keystroke dynamics to controller inputs, based on the analogy between controller and keyboard operations. Both types of operations involve discrete inputs from a limited set of buttons, generating time-series data as operation logs. Therefore, we hypothesize that by adapting methods used in keystroke authentication, we can achieve behavioral authentication in fighting games. However, controller inputs differ from keystroke dynamics in that joysticks and buttons operate independently, and the expected input speed is higher. Thus, this application is not straightforward and requires empirical validation.

For the second challenge, we design features and construct models that distinguish between in-combat and idle periods. We assume that players' behaviors differ between these states, and mixing them may degrade authentication accuracy. Therefore, we attempt to separate them to build models that capture the specific input characteristics of each state. We consider two approaches for this separation:

- Use match context data with annotations for in-combat and idle periods to distinguish between them.
- Identify in-combat and idle periods using some method based solely on controller input logs.

The former can reliably separate the states using match context data but requires information beyond controller input logs. The latter uses only controller input logs, making it a more generalizable method. We adopt a staged approach: first, we verify whether handling context changes is feasible using the former, and then we aim for a practical method using the latter.

3.3.2 Two phases of this study

Based on the ideas above, this study proceeds in two phases. The first phase is the proof of concept, where we verify the effectiveness of applying keystroke dynamics to controller inputs and the impact of separating in-combat and idle periods using match context data. We propose a basic method that addresses the two challenges mentioned above and evaluate it on a small-scale dataset of fighting

game players. To address the first challenge, we propose a method that computes features from controller inputs based on keystroke authentication techniques for player authentication. To address the second challenge, we adopt the approach of using match context data to distinguish between in-combat and idle periods. Specifically, we use round start and end timestamps, collected separately from controller inputs, to identify these states and incorporate this information into the features. After proposing this basic method that tackles the challenges, we evaluate its effectiveness on a small-scale dataset of fighting game players.

The second phase pushes the method further based on insights gained from the proof of concept and evaluates it on a larger and more diverse dataset. For the method addressing the second challenge, we improve it to identify in-combat and idle periods using only controller input logs instead of match context data. This allows for authentication using solely controller input logs. We evaluate this method on a larger-scale dataset of fighting game players, using the same game as in the proof of concept.

3.4 Task setting

The primary objective of this study is to evaluate player authentication using controller input logs captured in real-world gameplay environments. This section describes the underlying scenarios and threat models, and formalizes the specific task that constitutes player authentication in this context.

3.4.1 Scenario and Threats

We consider a scenario involving an online esports tournament. During the registration phase, participating players submit their historical gameplay logs, or *enrollment logs*. The tournament organizers then use these logs to construct authentication models for each player. On the day of the tournament, the organizers determine whether the operator of a given match is indeed the registered individual based on the gameplay logs obtained during the session, or *verification logs*. In this study, we define the unit of decision-making as a *battle*, which is a sequence of up to three rounds that determines a winner. Consequently, the entire log captured during a single battle is treated as the verification log for authentication.

As a potential threat, we assume the impersonation scenarios discussed in Section 2.3. These include collaborative impersonation, where a registered player voluntarily lends their account to another person, and adversarial impersonation, where an attacker illicitly acquires a player's credentials to participate. The attackers in these scenarios are not limited to registered players (*internal impostors*) but also include unregistered individuals (*external impostors*). Since the authenticator is constructed solely using the enrollment logs of registered players, external impostors generally possess less information than internal ones, which often makes it more challenging for the system to reject them effectively.

Regarding the attacker model, we assume a *zero-effort attack*, where the attacker does not intentionally attempt to mimic the target player's behavior. As mentioned in Section 3.3.1, this research applies keystroke dynamics methodologies to controller inputs in fighting games. While the possibility of mimicry attacks has been noted in keystroke dynamics, such attacks typically target fixed-text scenarios (e.g., passwords) rather than the unconstrained, free-form inputs characteristic of general computer operations [27]. Given that fighting game inputs require highly flexible responses to dynamic, real-time situations, the effectiveness of mimicry attacks remains unclear; thus, they are beyond the scope of this study.

3.4.2 Formalization of the Authentication Task

The role of the authentication system in the aforementioned scenario can be formalized as follows. Let $\mathcal{P} = \{p_1, p_2, \dots\}$ be the set of registered players. An operation log l is represented as time-series data consisting of the timestamps and types of each controller event, and we denote the universe of all such logs as $\mathcal{L}$. The enrollment data for each player $p \in \mathcal{P}$ is defined as $\mathcal{D}^p = \{l_1^p, l_2^p, \dots\} \subset \mathcal{L}$. The tournament organizer constructs an authenticator using the collection of enrollment logs $\{\mathcal{D}^p\}_{p \in \mathcal{P}}$.

The authenticator is provided with a claimed identity $p_{\text{claim}} \in \mathcal{P}$ and a verification log $l_{\text{verify}} \in \mathcal{L}$ obtained from an entire battle. We introduce a random variable $P^\star \in \mathcal{P} \cup \mathcal{A}$ to represent the true identity of the operator, where $\mathcal{A}$ denotes the set of unregistered players. The player authentication task is then defined as a binary verification problem to distinguish between the following two hypotheses:

$$H_{\text{genuine}} \colon P^\star = p_{\text{claim}}, \quad H_{\text{impostor}} \colon P^\star \neq p_{\text{claim}},$$

where H_{genuine} represents the case where the operator is the claimed individual p_{claim}, and H_{impostor} represents any other case.

The decision rule f required for the authenticator is expressed as $f \colon \mathcal{L} \times \mathcal{P} \to \{0, 1\}$. The system accepts H_{genuine} if $f(l_{\text{verify}}, p_{\text{claim}}) = 1$ and accepts H_{impostor} if $f(l_{\text{verify}}, p_{\text{claim}}) = 0$. In practice, we define a scoring function $g \colon \mathcal{L} \times \mathcal{P} \to \mathbb{R}$ and a threshold $\kappa \in \mathbb{R}$ such that:

$$f(l_{\text{verify}}, p_{\text{claim}}) = \begin{cases} 1 & \text{if } g(l_{\text{verify}}, p_{\text{claim}}) \geq \kappa, \\ 0 & \text{otherwise.} \end{cases}$$

Under our attacker model, we limit H_{impostor} to zero-effort attacks and do not consider scenarios where the attacker mimics the target p_{claim}. That is, the verification log l_{verify} under H_{impostor} is generated from the attacker's inherent behavioral distribution, and this process does not involve any optimization to approximate the distribution of p_{claim}.

4 Scene-based Approach

In this section, as a proof of concept, we propose a basic method for player authentication in fighting games using controller operation dynamics. Fighting games have two unique characteristics: discrete, specialized input devices of controllers, and frequent scene transitions between in-combat and idle periods within a battle. For the former challenge, we propose a method that adapts model architectures conventionally used in keystroke dynamics authentication, and for the latter, we segment operation logs into specific match contexts using round start and end timestamps, which are then incorporated into the feature engineering process.

4.1 Method

Our objective is to verify a player's identity in fighting games by modeling controller-input dynamics. We propose the method illustrated in Figure 1, which consists of four stages: (a) data collection, (b) preprocessing, (c) feature engineering, and (d) classification.

4.1.1 Game and Data Collection

We selected *Street Fighter 6* (SF6)[2], one of the most popular fighting games in esports, as the target title. In SF6, players use controllers to execute character movements and attacks. These actions require coordinated button presses with precise timing, which makes SF6 highly skill-based and competitive, and thus an appropriate candidate for our study.

We collected controller-operation logs by recording the inputs pressed in each frame using an attached logging device. Since SF6 runs at 60 frames per second, one frame corresponds to 1/60 second. Each log entry consists of a timestamp and the set of controller inputs pressed at that frame. The inputs include four directional inputs from the joystick (↑, ↓, ←, →) and nine action buttons (○, ×, △, □, L1, L2, R1, R2, option), as shown in Figure 1(a). The directional inputs generally correspond to jumping (↑), crouching (↓), and horizontal movement (←, →), and are also used for defensive actions and command inputs. The action buttons are used to trigger character actions, such as attacks, throws, special moves, and auxiliary actions, depending on the character and the control settings. Directional inputs and action buttons can be pressed simultaneously, so a single log entry may contain multiple input labels, as in "←↑R2" in Figure 1(a).

[2] `https://www.streetfighter.com/6/en-us`

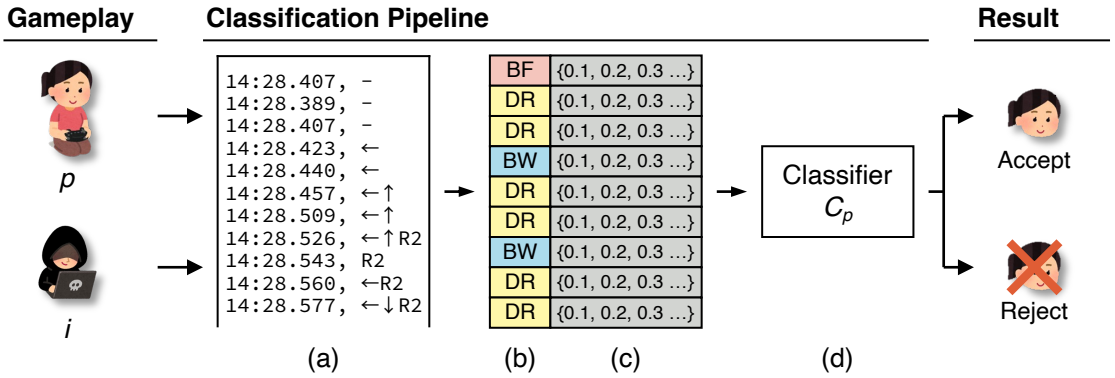


Figure 1: **Overall architecture of the scene-based approach.** Left: the legitimate player p and an impostor i play the game. Middle: (a) we record controller operation logs, where each row consists of a timestamp and the controller inputs pressed at that frame; (b) we split the logs into scene-based segments; (c) for each segment, we compute temporal features; and (d) we classify the segment with C_p, trained for target player p, to decide whether p produced the segment. Right: we accept p as genuine and reject i as an impostor.

4.1.2 Preprocessing

An SF6 battle comprises up to three rounds: a round is won by reducing the opponent's health to zero, and the first player to win two rounds wins the match. A round typically ends in approximately 30–60 seconds, and players have a short preparation period between rounds.

To deal with these frequent context changes, we segmented the controller operation logs based on game scenes. We partitioned a battle into three scenes based on the match procedure: *before-round* (BF) from battle start to the first round; *during-round* (DR) from round start to round end; and *between-round* (BW) from round end to the next round start. For example, a three-round battle yields the scene sequence: BF, DR, BW, DR, BW, DR.

After scene annotation, we split the logs in each scene into *segments* starting from the beginning of the scene as depicted in Figure 1(b). We denote the length of a segment by θ frames (equivalent to $\theta/60$ seconds); the final segment in a scene may be shorter. Accordingly, a scene lasting n frames yields $\lceil n/\theta \rceil$ segments. We set $\theta = 500$ frames (approximately 8.3 seconds) as the default, since this value corresponds to the median segment length across scenes. We also conducted a parameter study to evaluate different θ values.

4.1.3 Feature Engineering

From each segment, we extracted two types of information: the scene label and key-timing events. We used the scene label (BF, DR, or BW) to capture context-dependent operation trends even when segments are mixed with segments from other scenes.

We also computed temporal features inspired by keystroke dynamics, as illustrated in Figure 1(c). Keystroke dynamics are analogous to controller-operation dynamics, as described in Section 3.3.1, and we thus adapted the feature engineering principles of keystroke dynamics to our controller data. Specifically, we computed five temporal metrics: one holding time and four interval times, as depicted in Figure 2. A keypress event K is defined as a tuple of a key name k, pressed-down time d, and released-up time u: $K = (k, d, u)$. Using this notation, the sequence of keypress events in a segment is denoted as

$$\mathcal{K} = \{K_1, K_2, \dots\} = \{(k_1, d_1, u_1), (k_2, d_2, u_2), \dots\}.$$

For each keypress K_i ($i \in \{1, 2, \dots, |\mathcal{K}|\}$), we calculated the holding time H_i as the duration between the pressed-down time d_i and released-up time u_i: $H_i = u_i - d_i$. In addition, for each consecutive

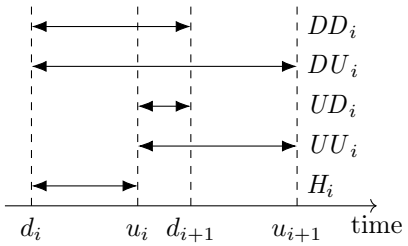


Figure 2: **Controller input features**. d_i, u_i are pressed-down and released-up time of the i-th keypress, and d_{i+1}, u_{i+1} are those of the $i+1$-th keypress. H_i is the holding time of the i-th key. UU_i, UD_i, DU_i, and DD_i are the time intervals between two keys.

pair of keypresses (K_i, K_{i+1}) $(i \in \{1, 2, \ldots, |\mathcal{K}| - 1\})$, we computed four interval metrics:

$$DD_i = d_{i+1} - d_i, \quad UU_i = u_{i+1} - u_i, \quad DU_i = u_{i+1} - d_i, \quad UD_i = d_{i+1} - u_i.$$

For the holding times H_i, we computed the sum, mean, variance, and standard deviation by key. Let $\mathcal{C}$ be the set of all keys; for each key $c \in \mathcal{C}$, we computed these statistics for the subset of holding times $\{H_i \mid k_i = c\}_{i=1}^{|\mathcal{K}|}$. Similarly, for the interval times DD_i, UU_i, DU_i, and UD_i, we computed the same statistics for each key combination: for each pair of keys $(c, c') \in \mathcal{C} \times \mathcal{C}$, we computed these statistics for the subset of intervals

$$\{DD_i \mid k_i = c, k_{i+1} = c'\}_{i=1}^{|\mathcal{K}|-1}, \quad \{UU_i \mid k_i = c, k_{i+1} = c'\}_{i=1}^{|\mathcal{K}|-1},$$
$$\{DU_i \mid k_i = c, k_{i+1} = c'\}_{i=1}^{|\mathcal{K}|-1}, \quad \{UD_i \mid k_i = c, k_{i+1} = c'\}_{i=1}^{|\mathcal{K}|-1}.$$

These statistics constitute the feature vector for each segment.

4.1.4 Classifiers

Consistent with our task definition (Section 3.4), we employed a one-vs-rest classification strategy. One-vs-rest is a common approach for multi-class classification problems, where a separate binary classifier is trained for each class to distinguish it from all other classes. For each player p, we trained a binary classifier C_p using the positive samples from player p and negative samples from all other players. The classifier C_p takes the feature vector extracted from a segment as input and outputs the probability $q_p(s) \in [0, 1]$ that the segment s was produced by player p.

As base learners, we considered four commonly used algorithms: Random Forest, Support Vector Machine, XGBoost, and LightGBM. These algorithms represent complementary modeling families and provide robust, interpretable performance. We evaluate their comparative performance to identify the most effective learner for our task.

4.2 Experiment

The goal of this experiment was to evaluate the feasibility of controller-dynamics authentication in a fighting game. We provide details of the experimental setup and design in the following.

4.2.1 Dataset

For this experiment, we collected a dataset called *SF6-small* that contains controller operation logs from professional players of SF6. The SF6-small dataset comprised controller logs from 99 battles played by nine professional players, labeled A–I. All players selected the same character and participated in multiple one-on-one matches. The data collection was conducted with the cooperation of JCG Co., Ltd.[3]. This dataset was collected with the approval of the Ethics Review Committee of the Graduate School of Information Science and Technology, The University of Tokyo.

[3] `https://jcg.co.jp/`

4.2.2 Evaluation

We evaluated the proposed method using the following experimental setup. Cross-validation method and metrics are designed to ensure robust and reliable performance estimation, while the two decision levels allow us to assess both the classifier's performance on individual segments and its practical performance in real-world scenarios.

Cross Validation To mitigate occasional instability, we employed five-fold stratified group cross-validation: stratification preserves class balance for one-vs-rest evaluation, and grouping by battle prevents data leakage. In each split, four folds form the training set, and the remaining fold forms the test set. During training, we used Optuna [2] to optimize hyperparameters for each model, holding out one fold for validation; the search ranges are detailed in Appendix A.

Metrics We chose the area under the precision-recall curve (PR-AUC) and the equal error rate (EER) as primary evaluation metrics. PR-AUC measures the trade-off between precision and recall and is particularly informative for imbalanced datasets in our one-vs-rest strategy. EER denotes the point at which the false acceptance rate (FAR) and false rejection rate (FRR) are equal, providing a balanced measure of performance between security and usability. Formal definitions are given in Appendix B.

Decision level We evaluated authentication performance at two decision levels: segment-level and battle-level. A classifier of a claimed player p outputs a probability $q_p(s)$ for a segment s indicating the likelihood that s was produced by player p. At the segment-level evaluation, we computed the metrics directly from these segment probabilities, which enables the pure assessment of classifier performance on individual segments. For battle-level evaluation, we aggregated segment-level probabilities within a battle using majority voting. Given a battle comprising segments $\mathcal{S} = \{s_1, s_2, \dots\}$, we computed the battle-level probability $\hat{q}_p = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} q_p(s)$. We then computed the metrics based on these battle-level probabilities. This way of evaluation reflects practical scenarios where authentication decisions are made over the course of an entire battle rather than on isolated segments.

4.2.3 Experimental Design

Our experiment investigates the validity of controller-input dynamics for authentication. We first measure the performance of various settings on a simple verification task to determine the most suitable model, and then evaluate the chosen model on more challenging settings.

Closed-set Evaluation To select the most suitable type of segments and settings for our task, we compared verification performance across different scenes and base learners. For scene comparisons, we formed four distinct data groups (G1–G4) as detailed in Table 2. Group G1, G2, and G3 correspond respectively to all data, idle-time data, and in-combat data, enabling comparisons across gameplay situations. Because BF segments were, unlike DR and BW segments, typically too short to evaluate independently, we prepared G4 (between-round data) to isolate the effect of BF segments when comparing with G2. In addition, we evaluated four base learners (Random Forest, Support Vector Machine, XGBoost, and LightGBM) and compared their results to identify the most suitable algorithm for our model and task.

To select the best model effectively, we evaluated the performance of models in different conditions to detect inside impostors. Specifically, the set of players in both the training and testing datasets is $\mathcal{P} = \{\mathrm{A}, \mathrm{B}, \dots, \mathrm{I}\}$. For each target player $p \in \mathcal{P}$, we trained a classifier C_p using training data from all players in $\mathcal{P}$ and evaluated its performance on all players' data. We repeated this with varying data groups and base learners and compared their results.

Table 2: **Data groups**. Group G1–G4 consists of specific segment types with ✓ (BF: before-round, DR: during-round, BW: between-round) for their representing scenes. The number of segments under their length $\theta = 500$ is shown for each group.

Group	Scenario	Constituent segments			#Segments
		BF	DR	BW	
G1	All data	✓	✓	✓	1529
G2	Idle-time data	✓		✓	390
G3	In-combat data		✓		1139
G4	Between-round data			✓	298

Open-set Evaluation To test applicability to outside impostors unseen during training, we conducted an open-set verification using the best data and models identified in the closed-set evaluation. Here, the testing dataset includes players excluded from the training dataset. We selected the target player $p \in \mathcal{P}$ and outside impostor $i \in \mathcal{P} \setminus \{p\}$. We trained classifier C_p using training data of $\mathcal{P} \setminus \{i\}$, and evaluated its performance on data from all players, including the held-out player i as the outside impostor. We repeated this for all combinations of target players and outside impostors and reported the average performance, which was across 72 combinations.

4.3 Results

We present the results of experiments on player authentication using controller operations. Briefly, the Random Forest classifier trained on in-combat segments achieved the best closed-set performance and also performed acceptably in open-set verification.

4.3.1 Closed-set Verification

Table 3 summarizes the performance for closed-set player authentication across four segment groups (G1–G4) and four learners, at the segment-level and battle-level, respectively. The best result was obtained using G3 (in-combat) with the Random Forest classifier. This combination consistently outperformed the others, reaching 75.7 % PR-AUC and 13.8 % EER. This performance is substantially better than random guessing for a nine-player verification task, which would yield 11.1 % PR-AUC and 50.0 % EER.

Table 3 reveals a clear performance hierarchy across game scenes and base learners. The model trained exclusively on in-combat data (G3) performed best, followed by the model using all data (G1), and then the models trained primarily on idle-time data (G2 and G4). Specifically, the PR-AUC of the in-combat model (G3: 75.7 %) was higher than that of the idle-time models (G2: 66.7 %; G4: 67.2 %). Across all experimental conditions, Random Forest was the most effective classifier, followed by XGBoost.

Table 4 provides the battle-level verification results. Via majority voting, the battle-level performance improved to nearly perfect performance, reaching 99.9 % PR-AUC and 0.7 % EER for the best model. The best condition was different from the segment-level evaluation, with the model trained on all data (G1) outperforming the in-combat model (G3).

4.3.2 Open-set Verification

Based on the closed-set findings, we conducted open-set verification experiments using Random Forest with the two best data groups (G1 and G3). Table 5 summarizes performance for each verification case; the open-set evaluation reports the average performance across all combinations of the target player and outside impostor. For both metrics, open-set verification yielded lower performance than closed-set verification, yet still outperformed random guessing.

Table 3: **Closed-set verification performance (segment-level)**. The table shows the mean and standard deviation of PR-AUC and EER of four base learners of Random Forest (RF), Support Vector Machine (SVM), XGBoost (XGB), and LightGBM (LG) across four data groups (G1–G4). The best performance in each group among base learners is highlighted in **bold**, and the overall best performance across groups is underlined.

(a) PR-AUC [%] ↑

Group	RF	SVM	XGB	LG
G1 (All data)	**72.0 ± 15.3**	53.8 ± 17.0	68.9 ± 17.5	68.3 ± 16.8
G2 (Idle time)	**66.7 ± 20.2**	50.3 ± 26.0	61.9 ± 19.2	60.4 ± 21.0
G3 (In-combat)	**75.7 ± 13.5**	54.2 ± 22.7	71.7 ± 15.9	70.8 ± 17.1
G4 (Between rounds)	**67.2 ± 24.8**	52.1 ± 28.7	58.1 ± 25.7	56.6 ± 24.6

(b) EER [%] ↓

Group	RF	SVM	XGB	LG
G1 (All data)	**15.1 ± 5.4**	20.6 ± 5.7	**15.1 ± 5.2**	15.2 ± 5.2
G2 (Idle time)	**19.6 ± 10.7**	28.3 ± 18.4	20.0 ± 11.4	20.5 ± 10.8
G3 (In-combat)	**13.8 ± 6.1**	22.6 ± 9.4	14.7 ± 5.7	14.9 ± 6.3
G4 (Between rounds)	**16.0 ± 13.5**	27.0 ± 19.2	19.9 ± 14.1	17.5 ± 12.3

4.4 Discussion

We interpret our experimental findings and discuss their implications, including the validity of our approach, the challenges of open-set verification, and the impact of game context on authentication performance. We also conduct parameter studies to analyze the effects of segment length.

4.4.1 Validity of Our Approach

Our method successfully distinguished the legitimate player from both other registered players and unknown impostors. This result implies the effectiveness of our approach in capturing unique behavioral patterns of players via keystroke-inspired features and context-aware segmentation.

Moreover, aggregating segment-level predictions to the battle level significantly improved performance, suggesting that while individual segments may be noisy, their collective patterns provide a robust signal for authentication. The difference in the best-performing data groups can be attributed to the difference in the amount of data: G1 (all) had more segments than G3 (in-combat), resulting in better battle-level performance for G1. Though this aggregation takes more time to make a decision, it shows the potential for practical applications, where decisions are often made over a battle rather than on isolated segments.

4.4.2 Outside Impostor Detection

Open-set verification showed lower performance than closed-set verification (Table 5). This trend was consistent across all players' classifiers, as demonstrated in Figure 3, indicating generally weaker performance in open-set scenarios.

This degradation occurred because the model tended to fail to reject outside impostors. The detection error trade-off (DET) curves in Figure 4 compare FAR for inside and outside impostors; they show that, for the same FRR of genuine users, FAR was higher for outside impostors.

This failure in rejection was due to the model's reliance on training data, which limited its ability to generalize to unseen players. Nevertheless, in both scenarios, the model performed player verification with reasonable accuracy, exceeding random guessing. These results indicate that our model captured features unique to genuine players relative to others, even when those others were

Table 4: **Closed-set verification performance (battle-level)**. The mean and standard deviation of PR-AUC and EER is reported. The best performance in each group among base learners is highlighted in **bold**, and the overall best performance across groups is underlined.

(a) PR-AUC [%] ↑

Group	RF	SVM	XGB	LG
G1 (All data)	98.7 ± 7.0	97.3 ± 9.2	**99.9 ± 0.7**	99.0 ± 4.6
G2 (Idle time)	**91.4 ± 15.6**	67.3 ± 30.1	82.4 ± 23.0	84.1 ± 23.5
G3 (In-combat)	**98.0 ± 5.9**	92.1 ± 21.5	**98.0 ± 9.3**	95.0 ± 18.1
G4 (Between rounds)	**82.8 ± 25.8**	63.9 ± 34.7	72.1 ± 30.5	73.7 ± 28.8

(b) EER [%] ↓

Group	RF	SVM	XGB	LG
G1 (All data)	0.19 ± 0.73	0.33 ± 1.2	**0.07 ± 0.47**	0.20 ± 1.0
G2 (Idle time)	**4.0 ± 11.2**	20.8 ± 27.8	7.0 ± 13.8	7.3 ± 15.9
G3 (In-combat)	0.37 ± 1.1	3.4 ± 14.2	**0.25 ± 0.99**	2.5 ± 14.5
G4 (Between rounds)	**4.85 ± 10.1**	26.3 ± 31.7	10.6 ± 15.2	8.0 ± 13.5

Table 5: **Open-set verification performance.** The table shows the mean and standard deviation of metrics in each player's classifiers applying Random Forest on two data groups.

Group	PR-AUC [%] ↑		EER [%] ↓	
	Closed-set	Open-set	Closed-set	Open-set
G1 (All)	72.0 ± 15.3	64.1 ± 19.0	15.1 ± 5.4	16.2 ± 6.0
G3 (In-combat)	75.7 ± 13.5	66.5 ± 19.6	13.8 ± 6.1	14.5 ± 6.0

excluded from training. This finding supports the potential of controller operation dynamics as a biometric modality in fighting games, even against unknown adversaries.

4.4.3 The Impact of Game Context

Our results in Table 3 show a clear performance gap between in-combat (DR) and idle-time (BW, BF) segments: in-combat segments yielded higher authentication performance. While this aligns with the general biometric observation that more constrained tasks usually yield higher accuracy [17, 24], we argue that the primary driver here is input density.

Table 6 shows classifier characteristics with respect to inputs. The input ratio is defined as the proportion of frames with at least one button press. Idle-time data are sparse, showing long inactive periods, whereas in-combat segments are dense with continuous actions. Table 6 also compares the most important features for each classifier, measured by Gini importance from Random Forest. The top features for classifiers using G3 were consistently related to actual button operations, while classifiers trained on idle-time data tended to rely on inactivity.

These results, enabled by our interpretable models, suggest that segmenting matches from the entire gameplay to isolate in-combat periods improves player-authentication performance in fighting games. This improvement can be attributed to the high density of controller operations during combat, which yields distinctive, player-specific patterns.

4.4.4 Parameter Study

To investigate the effect of segment length on model performance, we conducted a parameter study by varying segment length θ and evaluating the resulting metrics (Figure 5). For both G1 (all)

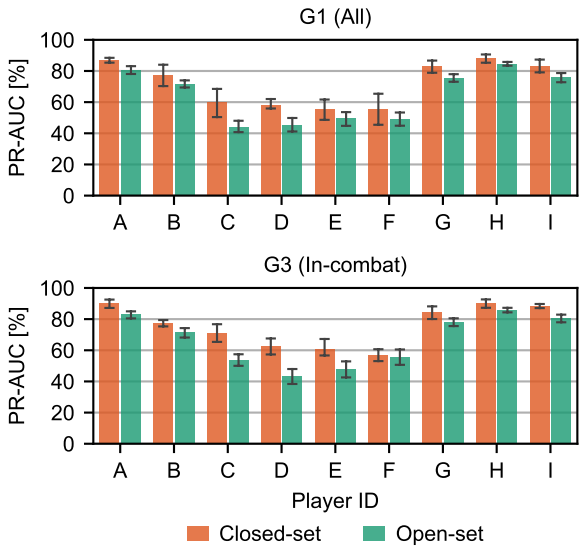


Figure 3: **Closed- and open-set performance per player**. The performance of each player's classifier with G1 (above) and G3 (below) is measured with PR-AUC. Each bar represents the mean performance, with error bars indicating the 95 % confidence interval.

Table 6: **Top feature buttons**. Density is the number of active frames divided by idle frames. The symbols for each group and player represent the buttons that comprise the top feature measured by Gini importance in Random Forest. The cases where inactivity "–" is considered the most important are shown in gray cells.

Group	Density	Player								
		A	B	C	D	E	F	G	H	I
G1	60.5	–, –	×	–, –	△	–, –	↑	↑	–, –	→
G2	39.4	↓	△, □	–, –	→	□	–, –	–, –	–	–
G3	69.0	□, △	R1	◯, ◯	–	△, R1	↑	R1, ↓	R2	□
G4	42.0	op, ×	△, □	–	→	↓, □	–	↓, →	–	↓, ←

and G3 (in-combat), increasing segment length improved most metrics up to $\theta = 1000$, after which performance plateaued and became unstable. The superior performance of G3 over G1 persisted across segment lengths, with only a few exceptions.

4.5 Implications and Next Steps

We proposed a method for player authentication in fighting games using controller-input dynamics. Our experimental results demonstrated the feasibility of this approach, comprising two key components: gameplay segmentation based on game scenes and feature extraction inspired by keystroke dynamics. The best model, using Random Forest on in-combat segments, achieved 75.7 % PR-AUC and 13.8 % EER in closed-set verification, and 66.5 % PR-AUC and 14.5 % EER in open-set verification in segment-level evaluation. These findings suggest that controller-input dynamics can serve as a viable biometric modality for player authentication in fighting games.

However, we should note the main limitation of this method, which lies in its reliance on manual scene annotation. For practical, real-time deployment, such annotation is not feasible, and should be replaced with automatic detection of game-state transitions. Our analysis in Section 4.4.3 indicates that input density may serve as an effective signal for this purpose, as in-combat segments exhibited

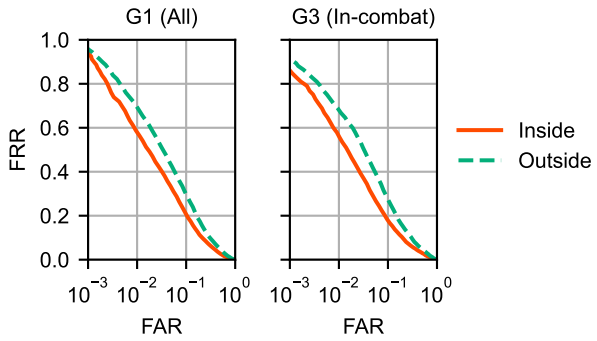


Figure 4: **Performance for inside and outside impostors**. FRRs of genuine players are plotted against FARs of inside and outside impostors at different thresholds. Each player's model is trained using G1 (left) and G3 (right) datasets, and the curve represents the average performance across players, with shaded areas indicating the 95 % confidence intervals.

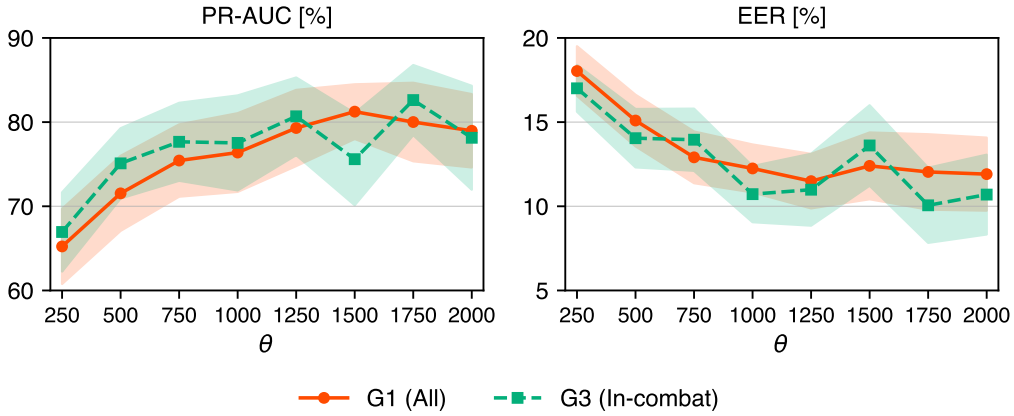


Figure 5: **Segment length and performance.** We measured the performance of classifiers for closed-set verification with varying segment lengths θ. The classifiers were based on Random Forest and used G1 and G3 datasets for each curve, with shaded areas indicating the 95 % confidence intervals.

higher input density than idle-time segments. In the next section, we propose a density-based segmentation approach to automatically identify in-combat periods, which we believe will enhance the practicality of our authentication method.

5 Density-based Approach

In the previous section, we investigated the validity of controller-based player verification by a keystroke-dynamics-inspired approach and context-aware segmentation. We demonstrated that using only in-combat data yields high verification accuracy, attributing this to the higher frequency of controller inputs during combat, which strongly reflects individual player characteristics. However, extracting in-combat data requires auxiliary information indicating the match progress, which may be difficult to obtain in practical applications.

Therefore, in this section, we focus on the input density of controller operations. To achieve verification without the need for auxiliary information such as match timing, we propose a method to automatically classify whether the player is in combat or resting based solely on controller inputs

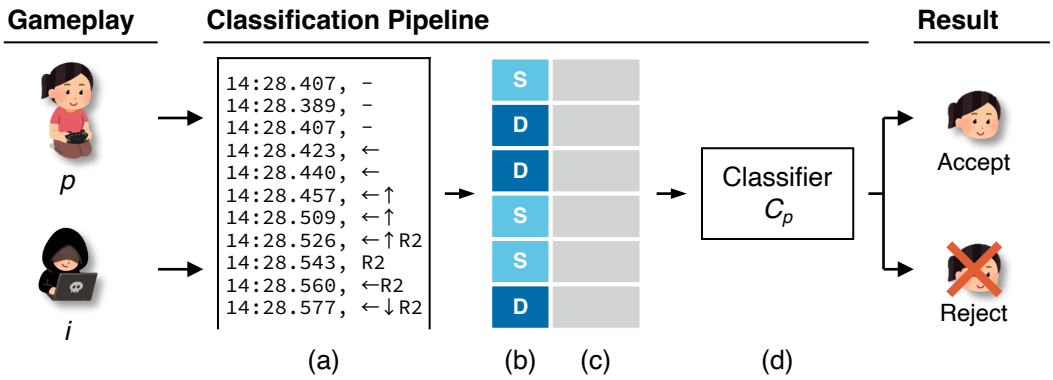


Figure 6: **Overall architecture of the density-based approach.** Left: the legitimate player p and an impostor i play the game. Middle: (a) we record controller operation logs, where each row consists of a timestamp and the controller inputs pressed at that frame; (b) we split the logs regardless of scenes and classify each segment based on density; (c) for each segment, we compute temporal features; and (d) we classify the segment with C_p, trained for target player p, to decide whether p produced the segment. Right: we accept p as genuine and reject i as an impostor.

and extract important data. We then apply the proposed method to a dataset collected from actual tournaments and evaluate its effectiveness.

5.1 Method

We propose a density-based segmentation method to extract segments with high controller activity and leverage them for player verification, as depicted in Figure 6. This method mainly differs from the scene-based method described in Section 4.1 in the previous section in the segment classification step, where we classify segments into dense and sparse based on input density, rather than classifying them based on match states.

In preprocessing, we split the data into segments of a fixed length θ frames, regardless of the start and end times of rounds, by simply cutting out data every θ frames from the beginning. We then classify each segment as *dense* or *sparse*. We calculate the input density as $N_{\text{active}}/N_{\text{all}}$, where N_{all} is the number of frames in the segment and N_{active} is the number of frames with inputs. If the ratio exceeds a threshold ρ, the segment is classified as dense; otherwise, it is classified as sparse. Subsequently, we compute temporal features for each segment in the same manner as in Section 4.1.3 and use their statistics as features. Using these features, we train a classifier for each player, and use it to verify whether the player is genuine or an impostor based on the segment's features.

5.2 Experiment

In this section, we explain the experimental setup and design for evaluating the proposed density-based method, and then present the results and discussion. The difference from the scene-based method described in Section 4.2 lies in the larger dataset and experimental design.

5.2.1 Dataset

In addition to the SF6-small dataset for which the participants are limited to 9 players, we also collected a larger dataset named *SF6-large*. The SF6-large dataset consists of controller operation logs from 307 battles played by 60 players during actual tournaments. The collection of this dataset was approved by the Ethics Review Committee of the Graduate School of Information Science and Technology, The University of Tokyo.

5.2.2 Evaluation

As in Section 4.2.2, we evaluated the proposed method using PR-AUC and EER using five-fold stratified group cross-validation. We computed these metrics at both the segment and battle levels. At the segment level, we calculated the metrics based on the classifier's output for each segment. At the battle level, we applied majority voting to the segment-level predictions for each battle and computed the metrics based on the battle-level predictions.

5.2.3 Experimental Design

We evaluated the proposed method in order to verify the validity of the density-based segmentation to extract important segments for player verification. To this end, we evaluate the proposed method on the SF6-large dataset using the following three types of segment sets.

Dense Only segments classified as dense are used for training and testing.

Sparse Only segments classified as sparse are used for training and testing.

All All segments are used for training and testing, without classification.

Comparison between the Dense and All settings allows us to verify the effectiveness of the density-based segmentation in extracting important segments for player verification.

We evaluated the proposed method under these protocols for both closed-set and open-set scenarios. For the closed-set scenario, we used all battles in the dataset for both training and testing. For the open-set scenario, we included battles from players outside the enrolled set as impostor data in the test set only. For open-set evaluation using the SF6-large dataset, for each target player $p \in \mathcal{P}$, we separated the resting players $\mathcal{P} \setminus \{p\}$ into three groups, and used two groups for training and one group for testing, rotating the groups in a three-fold cross-validation manner. This procedure is different from the one used for the SF6-small dataset, because of the long experiment time due to the larger number of players.

5.3 Results

Table 7 and Table 8 show the closed-set and open-set evaluation results on the SF6-small dataset, respectively. In the segment-level, which reflects the model's distinguishability, the Dense model achieved the highest performance in both closed-set and open-set evaluations at the segment-level, with PR-AUC of 73.0 % and EER of 12.8 % in the closed-set evaluation, and PR-AUC of 65.6 % and EER of 13.5 % in the open-set evaluation. Metric values increased at the battle-level, where majority voting was applied, with the Dense model achieving PR-AUC of 97.0 % and EER of 0.13 % in the closed-set evaluation, and PR-AUC of 95.7 % and EER of 0.24 % in the open-set evaluation. The Dense model outperformed other models in all evaluations, except for a few cases in battle-level evaluations.

Table 9 and Table 10 show the closed- and open-set evaluation results on the SF6-large dataset, which consists of 60 players' data. Similar to the results on the SF6-small dataset, the Dense model achieved the highest performance in both closed-set and open-set evaluations at the segment-level, with PR-AUC of 55.7 % and EER of 12.5 % in the closed-set evaluation, and PR-AUC of 48.0 % and EER of 13.0 % in the open-set evaluation, and all the results improved at the battle-level.

5.4 Discussion

Based on the experimental results above, we discuss the validity of the proposed density-based method and the effects of its parameters. In addition, we analyze the authentication performance for players using the same character to examine the influence of character-specific group behavior.

Table 7: **Closed-set evaluation using SF6-small dataset.** The mean and standard deviation of PR-AUC and EER are shown for segment-level and battle-level evaluations. **Bold** indicates the best performance in each decision level.

(a) PR-AUC [%] ↑

	Dense	Sparse	All
Segment-level	**73.0 ± 20.1**	51.7 ± 22.7	70.1 ± 17.5
Battle-level	97.0 ± 15.2	76.4 ± 25.4	**97.7 ± 8.2**

(b) EER [%] ↓

	Dense	Sparse	All
Segment-level	**12.8 ± 6.5**	25.1 ± 12.7	14.5 ± 5.7
Battle-level	**0.13 ± 0.63**	9.9 ± 15.3	0.37 ± 1.4

Table 8: **Open-set evaluation using SF6-small dataset.** The mean and standard deviation of PR-AUC and EER are shown. **Bold** indicates the best performance in each decision level.

(a) PR-AUC [%] ↑

	Dense	Sparse	All
Segment-level	**65.6 ± 23.1**	44.3 ± 23.2	62.4 ± 20.5
Battle-level	95.7 ± 17.2	65.9 ± 30.6	**96.3 ± 12.3**

(b) EER [%] ↓

	Dense	Sparse	All
Segment-level	**13.5 ± 7.0**	26.6 ± 12.4	15.8 ± 6.2
Battle-level	**0.24 ± 1.1**	10.9 ± 15.6	0.35 ± 1.2

5.4.1 Validity of Density-based Method

We proposed a density-based method that classifies segments based on input density. We have evaluated this method on two datasets of SF6-small with 9 players and SF6-large with 60 players, under two scenarios of closed-set and open-set evaluations. The results showed that our model, especially the Dense model, can verify players with high accuracy in both datasets and scenarios.

For datasets, evaluation on SF6-large showed lower performance compared to SF6-small, as shown in Table 7 to Table 10, where PR-AUC decreased by approximately 17 % and EER increased by around 0.3 %. This is due to the number of players increasing from 9 to 60, making the verification task more difficult. Random guessing from n players is expected to yield a PR-AUC of $1/n$ and an EER of 50 %. Therefore, the observed performance degradation is reasonable given the increased number of players.

Regarding scenarios, while the open-set evaluation is generally more difficult than the closed-set evaluation, the Dense model consistently achieved reasonably high performance in both scenarios. For instance, in the SF6-large dataset at the battle-level, the Dense model achieved PR-AUC of 95.8 % and EER of 0.14 % in the closed-set evaluation, and PR-AUC of 94.1 % and EER of 0.44 % in the open-set evaluation. This indicates that the Dense model effectively captures individual-specific input patterns, enabling robust verification even when encountering unseen impostors.

These findings indicate that segments with high input density contain more individual-specific and effective information for player verification, confirming the notion we discussed in Section 4.4 regarding in-combat segments. Throughout this method, we can achieve high verification accuracy without relying on auxiliary information, such as match progress, by focusing on input density alone.

Table 9: **Closed-set evaluation using SF6-large dataset.** The mean and standard deviation of PR-AUC and EER are shown. **Bold** indicates the best performance in each decision level.

(a) PR-AUC [%] ↑

	Dense	Sparse	All
Segment-level	**55.7 ± 22.1**	18.6 ± 17.3	45.1 ± 19.2
Battle-level	**95.8 ± 17.5**	47.5 ± 40.9	94.8 ± 18.6

(b) EER [%] ↓

	Dense	Sparse	All
Segment-level	**12.5 ± 8.5**	36.5 ± 12.5	19.7 ± 8.1
Battle-level	**0.14 ± 1.4**	10.1 ± 21.7	**0.14 ± 0.87**

Table 10: **Open-set evaluation using SF6-large dataset.** The mean and standard deviation of PR-AUC and EER are shown. **Bold** indicates the best performance in each decision level.

(a) PR-AUC [%] ↑

	Dense	Sparse	All
Segment-level	**48.0 ± 22.4**	15.7 ± 16.7	38.5 ± 18.6
Battle-level	**94.1 ± 20.8**	41.6 ± 41.6	92.5 ± 22.6

(b) EER [%] ↓

	Dense	Sparse	All
Segment-level	**13.0 ± 9.1**	36.8 ± 13.2	19.9 ± 8.4
Battle-level	0.44 ± 5.3	11.0 ± 23.0	**0.17 ± 0.79**

5.4.2 Parameter Study

We investigated the impact of density threshold ρ and segment length θ on verification performance using the SF6-large dataset, changing one parameter while keeping the other fixed as follows:

- Density threshold ρ: $\{0.2, 0.4, 0.6, 0.8\}$, with θ fixed at 250 frames
- Segment length θ: $\{250, 500, 750, 1000\}$ frames, with ρ fixed at 0.5

In the experiments involving ρ, the segment length θ was set to 250 to ensure a sufficient volume of training data. By utilizing shorter segments, we increased the total number of samples, thereby securing enough data even for extreme cases such as $\rho = 0.2$ or 0.8.

The evaluation results are illustrated in Figure 7. As ρ increases, a decrease in performance is observed for the Dense model, whereas the Sparse model shows an improvement. A higher density threshold ρ implies that only segments with higher input density are retained in the Dense segments. While this has a negligible effect when $\rho < 0.4$, values exceeding this threshold lead to the loss of unique, individual-specific input patterns. This loss results in the performance degradation of Dense and the corresponding improvement in Sparse. These findings suggest that in the SF6-large dataset, segments with an input density of 0.4 or higher significantly contribute to the authentication process.

Furthermore, an increase in θ leads to a performance improvement in all the models, but the Sparse model's improvement is marginal compared to the others. Generally, a larger θ increases the number of frames per segment but reduces the total number of segments obtainable from a single battle. For the Dense category, the increased information density per segment enhances performance. Conversely, for the Sparse category, the reduction in the total number of segments outweighs the

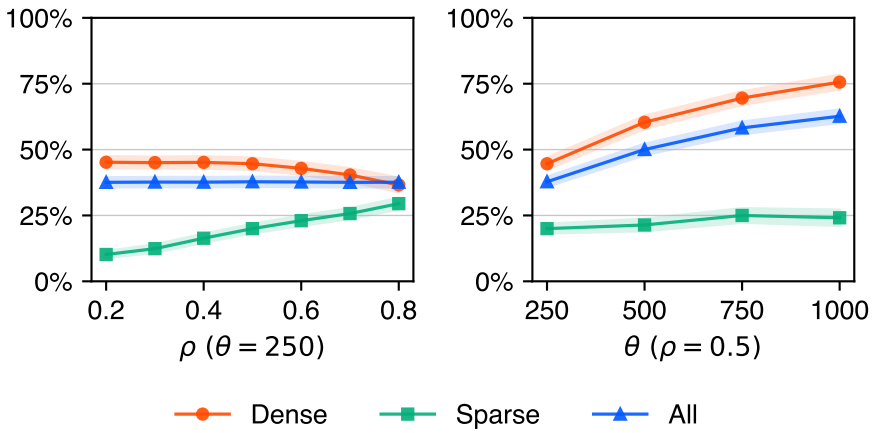


Figure 7: **Performance with different parameters.** PR-AUCs are shown for different density thresholds ρ with segment length θ fixed at 250 (left), and for different segment lengths θ with density threshold ρ fixed at 0.5 (right).

benefits of increased segment length, leading to a slight performance improvement. However, as the number of segments increases, the valid data with sufficient segments for evaluation decreases, and the training of classifiers becomes unstable. Therefore, it is necessary to select an appropriate θ that balances segment length and the number of segments.

5.4.3 Character-Specific Group Behavior

In fighting games, controller inputs are inherently dependent on the specific command sequences required for character-specific techniques. Consequently, players utilizing the same character may exhibit similar input patterns, potentially posing a challenge for behavioral authentication.

To investigate this impact, we conducted training and evaluation focused on the three characters with the largest player populations in our dataset: Gouki (11 players), Mai (9 players), and Ryu (8 players). For this analysis, the models were trained and tested exclusively using data from players associated with each respective character.

The PR-AUC for each character is presented in Figure 8. At the segment level, while the result for Ryu is lower compared to other characters, the overall results remain robust. At the battle level, high PR-AUC was achieved for all three characters in Dense and All settings, with values exceeding 90 %. These results demonstrate that player authentication based on controller dynamics remains highly effective, even when the pool of users is restricted to those playing the same character.

5.5 Implications and Next Steps

We introduced a density-based segmentation method in player authentication that classifies segments based on input density, and evaluated its effectiveness on two datasets of SF6-small and SF6-large under closed-set and open-set scenarios. The results demonstrated the effectiveness of the proposed method, particularly the Dense model, in achieving high verification accuracy.

So far in this paper, we have investigated controller operation dynamics using two segmentation approaches: scene-based and density-based methods. We observed their effectiveness throughout the experiments using two datasets of actual gameplays in SF6, which implies the feasibility of player authentication in fighting games using only controller operation data. In the following section, we will discuss these two approaches and their applicability in more detail.

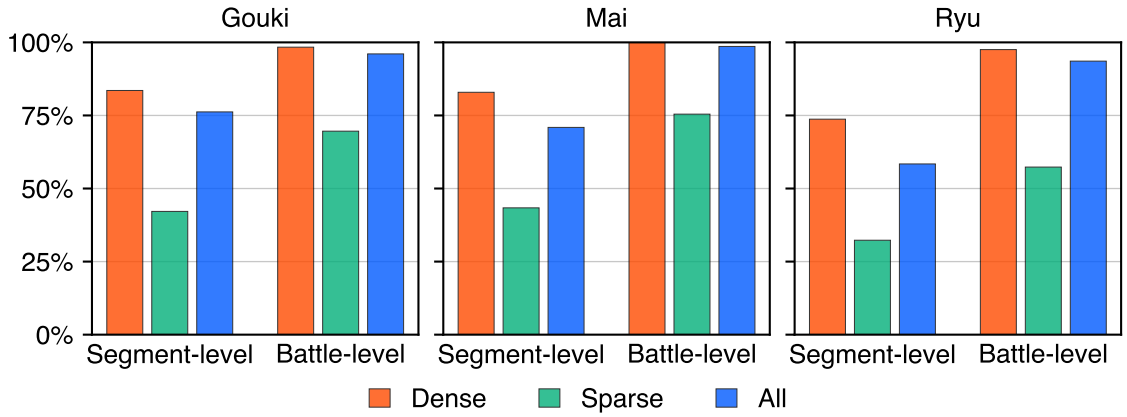


Figure 8: **Closed-set performance per character.** PR-AUCs calculated for the set of players using Gouki (left), Mai (middle), and Ryu (right) in the SF6-large dataset are shown.

Table 11: **Comparison between the scene-/density-based method**. The table shows the mean and standard deviation of PR-AUC and EER for each method on the SF6-small dataset.

Protocol		PR-AUC [%] ↑		EER [%] ↓	
Scene	Density	Scene	Density	Scene	Density
In-combat	Dense	75.7 ± 13.5	73.0 ± 20.1	13.8 ± 6.1	12.8 ± 6.5
Idle time	Sparse	66.7 ± 20.2	51.7 ± 22.7	19.6 ± 10.7	25.1 ± 12.7
All	All	72.0 ± 15.3	70.1 ± 17.5	15.1 ± 5.4	14.5 ± 5.7

6 Discussion

In this section, we discuss the main aspects of the two proposed methods, scene-based and density-based methods, which were confirmed to be effective through the experiments in Section 4.2 and Section 5.2, respectively. First, we compare the experimental results of the two methods in order to verify whether density-based segmentation can replace scene-based segmentation. Next, we discuss the applicability of the proposed method from the perspectives of authentication latency and population scalability. Finally, we note the limitations of this study and future work.

6.1 Comparison between Two Approaches

Table 11 compares the performance between the scene-based and the density-based method on the SF6-small dataset. When using data during idle periods (Idle and All), the density-based method shows lower performance than the scene-based method. On the other hand, when using only data during active periods (Active), the PR-AUC slightly improved from 75.7 % to 75.9 %, and the EER improved from 13.8 % to 12.7 %.

Regarding our proposal to replace the information about the match situation with input density, in the segment classification, the method in Section 4 used the start and end times of each round, while the proposed method used input density to replace that. When comparing with the conventional method using the SF6-small dataset (Table 11), almost the same performance was obtained in the case of using only data during active periods in the conventional method and using only dense data in the proposed method. This fact suggests that in player authentication, even without information about match situation data such as the start and end times of rounds, authentication is possible as long as controller operations are available.

6.2 Practical Considerations

We argue for the practical applicability of the proposed method from two perspectives: authentication latency and population scalability. The former discusses how long it takes to achieve high-confidence authentication, which is crucial for the realization of continuous player authentication during competitive matches. The latter examines the scalability of the proposed method when the number of enrolled players increases, which is important for applications in large-scale online gaming environments.

6.2.1 Authentication Latency

The practical utility of the proposed method lies in its ability to achieve high-confidence authentication within the operational constraints of a competitive match. To assess the authentication latency, we analyzed the relationship between the number of segments in a battle used for majority voting and the resulting battle-level authentication performance. Formally, as described in Section 4.2.2, given the battle comprising M segments $\{s_1, s_2, \ldots, s_M\}$, we computed the cumulative probability $\hat{q}_p^m$ for $m = 1, 2, \ldots, M$ as $\hat{q}_p^m = \frac{1}{m}\sum_{i=1}^{m} q_p(s_i)$, where $q_p(s_i)$ is the probability for segment s_i. By varying m from 1 to M, we evaluated how the authentication performance evolves as more segments are incorporated into the decision-making process.

Figure 9 illustrates the relationship between the number of segments used for majority voting and the battle-level authentication performance on the SF6-large dataset. The Dense model achieves a performance saturation point, which is defined as a PR-AUC exceeding 90 % and an EER below 1 %, at $m = 4$ (PR-AUC) and $m = 3$ (EER). Other models also exhibit performance improvements by number of segments, reaching as high as the dense model at last, but they require more segments to reach the saturation point (e.g, for EER, Sparse: $m = 13$; All: $m = 9$). We should note that the convergence of the Sparse model in the highest performance is due to the limited number of battles containing sufficient sparse segments.

Given that a typical round in *Street Fighter 6* lasts approximately 30 to 60 seconds, the Dense model can achieve reliable authentication within a single round. This demonstrates the feasibility of implementing the proposed method for real-time player identification without awaiting the conclusion of a battle.

6.2.2 Population Scalability

To evaluate the practical applicability of the proposed method in large-scale online environments, we investigated how the authentication performance changes as the number of enrolled players increases from 10 to 60. We randomly selected 10 players from the SF6-large dataset, and incrementally increased the number of enrolled players N from 10 (the initial players only) to 60 by adding 10 new players at a time. We repeated this process three times with different random seeds and averaged the results to ensure robustness.

Figure 10 shows the relationship between the number of enrolled players N and the closed-set evaluation results at both the segment-level and battle-level on the SF6-large dataset. At the segment-level (Figure 10a), although a natural decline in PR-AUC is observed as N increases due to the increased complexity of the verification task, the proposed Dense models consistently maintain superior performance compared to other models. Specifically, even at $N = 60$, the Dense model preserves a significantly higher PR-AUC and lower EER than the random selection, suggesting that density-based modeling effectively captures player-specific characteristics that remain discriminative even in larger populations. In battle-level judgments, the performance degradation with increasing N is slight for all models except the Sparse model, indicating that aggregating segment-level scores into battle-level decisions enhances robustness against the challenges posed by larger player pools.

These results demonstrate that the proposed method possesses sufficient scalability for real-world applications, such as identifying unauthorized players in online ranked matches. The aggregation of temporal operation density information allows the system to mitigate segment-level uncertainties, achieving highly reliable authentication even when the pool of potential impostors is large.

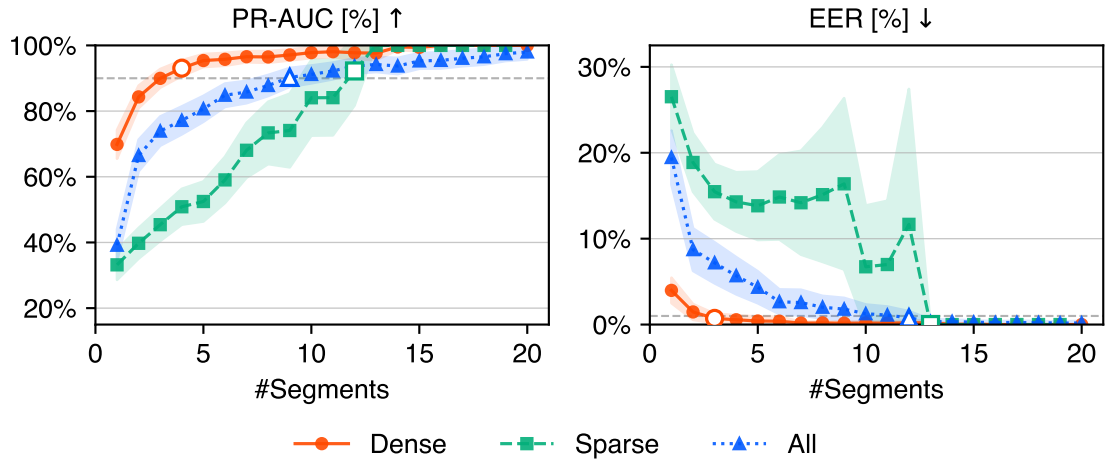


Figure 9: **The number of segments and performance**. The figure shows the relationship between the number of segments used for majority voting and the battle-level authentication performance (PR-AUC, EER) on the SF6-large dataset. Each point represents the mean value over all players, and the shaded area indicates the 95 % confidence interval. Gray dashed line indicates the performance saturation point (PR-AUC $> 90\,\%$, EER $< 1\,\%$), and open markers indicate the first points that exceed the saturation point for each model.

6.3 Positioning with Respect to Prior Work

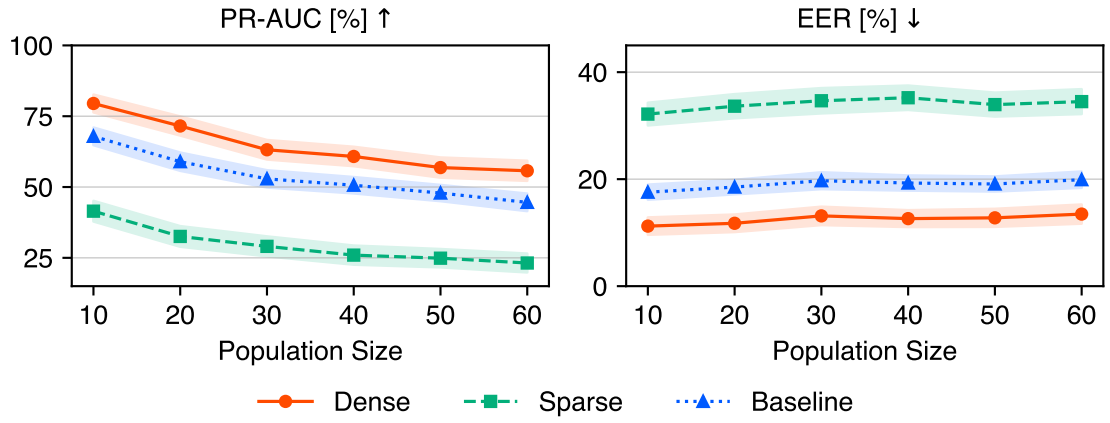
As discussed in Section 3, we address a gap in behavioral authentication for discrete input devices under rapidly changing gameplay contexts. Here, we position our work relative to prior game-focused behavioral authentication studies and summarize its implications.

With regard to input modality, we extend behavioral biometrics from continuous, high-degree-of-freedom interfaces to discrete controller inputs. Prior work reports strong performance with mouse, VR-controller, and touchscreen signals [20, 25, 28, 30], whereas keyboard-only inputs can be less discriminative in MOBA settings [8]. Our results show that controller operations can still provide competitive authentication performance. Although it is difficult to directly compare with prior work [8], this contrast suggests the role of feature design. Our method computes more specific features from shorter time windows than the previous work. These results suggest that discrete controller inputs can yield authentication-relevant behavioral signatures, which expands the applicability of behavioral authentication across game genres and devices.

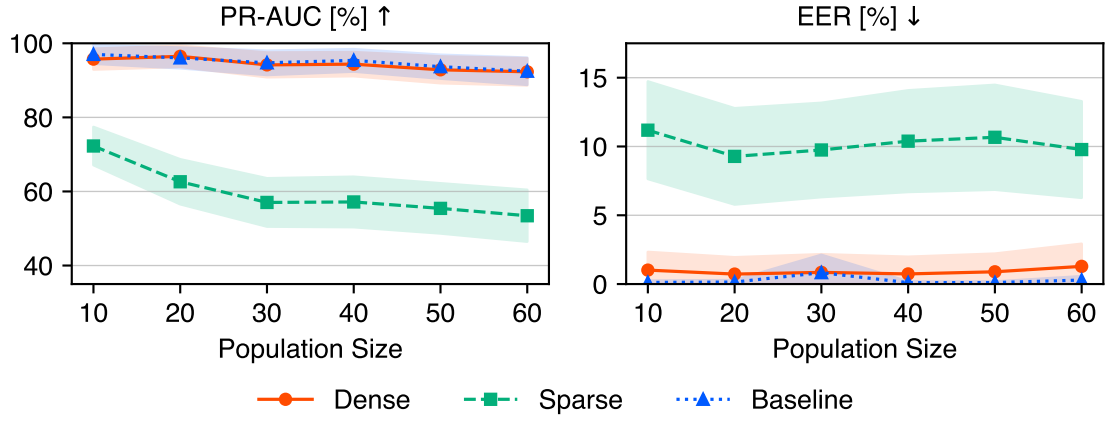
In terms of frequent context changes, high-input-density periods reliably capture informative activity, such as in-combat phases. Though this approach seems to align with prior works that aggregate short segments corresponding to specific tasks [20, 28], our method differentiates itself by its independence from explicit task labels. Density-based segmentation only requires full controller input logs: it identifies high-density segments from input timing alone, without game-state signals that may be unavailable in practice. Using density as a proxy for gameplay context may enable more deployable authentication systems for dynamic esports environments.

6.4 Limitations and Future Work

Finally, we acknowledge what this study does not cover as a preliminary investigation into the feasibility of controller-based behavioral authentication in fighting games. Such limitations include the threat model assumption, the dataset scale and diversity, and the generalizability of the proposed method. We also give a discussion of potential future work to address these limitations toward practical deployment in competitive environments.



(a) Segment-level



(b) Battle-level

Figure 10: **Scalability evaluation results**. The figures show the relationship between the number of enrolled players and the authentication performance (PR-AUC, EER) for (a) segment-level and (b) battle-level authentication. Each point represents the mean value of over trials selecting 10 players randomly, and the shaded area indicates the 95 % confidence interval.

Advanced threat models. Although this study focused on zero-effort attacks, more sophisticated impersonation attempts may arise in practical esports environments, including manual mimicry and attacks assisted by machine learning. Given prior access to the target player's controller operation data, an attacker may attempt to imitate their operation tendencies manually or utilize generative models to synthesize inputs that closely resemble the target player's behavior. Despite several studies investigating these attacks in the context of keystroke authentication [11,27], their applicability to fighting games is not straightforward; the attacker must not only reproduce plausible input patterns but also choose appropriate actions in response to the dynamic game situation under the objective of winning. We thus leave the feasibility of these advanced attacks, as well as the robustness of our method against them, for future work toward practical deployment in competitive environments.

Dataset diversity and longitudinal stability. Our datasets are constrained by their scale and diversity, as they primarily consist of players with high expertise. Furthermore, data collection for both the small and large datasets was conducted over a short duration, which may not capture long-term behavioral changes or the evolution of a player's style over time. Collecting

larger-scale datasets that include a broader range of skill levels and span extended time periods will be essential to validate the longitudinal robustness of our findings.

Generalizability across titles. This research utilized SF6 as the primary evaluation platform. While the proposed density-based segmentation proved effective for this title, fighting games with different command structures, pacing, or input modalities may yield different results. Evaluating our method across a wider variety of fighting games is necessary to establish its utility as a universal security measure within the broader esports ecosystem.

7 Conclusion

We explored the feasibility of player authentication in fighting games using controller operations, in order to address the issue of player impersonation in online esports tournaments. Fighting games are characterized by the discrete input modality of gaming controllers and the dynamic transitions between in-combat and idle scenes, which has not been explored in existing work on behavioral authentication in games.

To deal with these characteristics, we proposed a method that applies feature engineering techniques in keystroke authentication to controller operations, and two segmentation methods that separate different gameplay phases of in-combat and idle periods: scene-based segmentation relying on game state information, and density-based segmentation that identifies high-density input segments without game state information. Experimental results using data from nine professional players of *Street Fighter 6* demonstrated the effectiveness of our approach, achieving a PR-AUC of 75.7 % in closed-set verification and 66.5 % in open-set verification. The effectiveness of our method was further validated using larger dataset of 60 players, achieving a PR-AUC of 55.7 % in closed-set evaluation and 48.0 % in open-set evaluation. We also analyzed the latency and scalability of our method, and found that it can achieve reasonable performance with 30 seconds of controller operations, regardless of player population size available in this study.

Throughout this study, we have established an interpretable method for controller-dynamics authentication in fighting games that can perform if only controller input logs are available, and demonstrated its effectiveness using real gameplay data of professional players. We believe this study provides a foundation for further research into player authentication in fighting games and other esports genres, contributing to the integrity and security of competitive gaming.

Acknowledgments This work was supported by JST Moonshot R&D Grant Number JPMJMS2215.

References

[1] 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*, 4(1):57–70, 2022. doi:10.1109/TBIOM.2021.3112540.

[2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, pages 2623–2631, 2019. doi:10.1145/3292500.3330701.

[3] AKRacing. esports world, 2024. Available at https://esports-world.jp/ (accessed on December 12, 2024).

[4] M. Antal and E. Egyed-Zsigmond. Intrusion detection using mouse dynamics. *IET Biometrics*, 8(5):285–294, 2019. doi:10.1049/iet-bmt.2018.5126.

[5] B. Ayotte, M. Banavar, D. Hou, and S. Schuckers. Fast free-text authentication via instance-based keystroke dynamics. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(4):377–387, 2020. doi:10.1109/TBIOM.2020.3003988.

[6] C. BenAbdelkader, R. Cutler, and L. Davis. Stride and cadence as a biometric in automatic person identification and verification. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 372–377, 2002. doi:10.1109/AFGR.2002.1004182.

[7] F. Bergadano, D. Gunetti, and C. Picardi. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security*, 5(4):367–397, 2002. doi:10.1145/581271.581272.

[8] I. D. S. Beserra, L. Camara, and M. D. Costa-Abreu. Using keystroke and mouse dynamics for user identification in the online collaborative game League of Legends. In *Proceedings of the 7th International Conference on Imaging for Crime Detection and Prevention (ICDP 2016)*, 2016. doi:10.1049/ic.2016.0076.

[9] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a "Siamese" time delay neural network. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, pages 737–744, 1993.

[10] P. K. Czegledy. Esports integrity policies. *Gaming Law Review*, 25(4):161–170, 2021. doi:10.1089/glr2.2020.0017.

[11] I. Eizagirre, L. Segurola, F. Zola, and R. Orduna. Keystroke presentation attack: Generative adversarial networks for replacing user behaviour. In *Proceedings of the 2022 European Symposium on Software Engineering*, pages 119–126, 2023. doi:10.1145/3571697.3571714.

[12] T. Eude and C. Chang. One-class SVM for biometric authentication by keystroke dynamics for remote evaluation. *Computational Intelligence*, 34(1):145–160, 2018. doi:10.1111/coin.12122.

[13] O. L. Finnegan, J. W. White, B. Armstrong, E. L. Adams, S. Burkart, M. W. Beets, S. Nelakuditi, E. A. Willis, L. von Klinggraeff, H. Parker, M. Bastyr, X. Zhu, Z. Zhong, and R. G. Weaver. The utility of behavioral biometrics in user authentication and demographic characteristic detection: a scoping review. *Systematic Reviews*, 13(1), 2024. doi:10.1186/s13643-024-02451-1.

[14] Fortune Business Insights. eSports market size, share, and industry analysis, by streaming type (live and on-demand), by revenue streaming (media rights, advertisement, sponsorship, ticket & merchandise, game publisher fees, and others), by gaming genre (real-time strategy games, first person shooter games, fighting games, multiplayer online battle arena games, mass multiplayer online role-playing games, and others), and regional forecast, 2025-2032, 2025. Available at https://www.fortunebusinessinsights.com/esports-market-106820 (accessed on December 12, 2025).

[15] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, 2013. doi:10.1109/TIFS.2012.2225048.

[16] A. K. Jain, P. Flynn, and A. A. Ross, editors. *Handbook of Biometrics*. Springer, New York, NY, 2007. doi:10.1007/978-0-387-71041-9.

[17] S. Khan, C. Devlen, M. Manno, and D. Hou. Mouse dynamics behavioral biometrics: A survey. *ACM Computing Surveys*, 56(6), 2024. doi:10.1145/3640311.

[18] R. Kobayashi, M. Irvan, F. Zimmer, M. N. S. Perera, and R. S. Yamaguchi. Log-based authentication as a promising method for detecting cooperative impersonation from temporal and spatial perspectives. In *2026 18th International Conference on COMmunication Systems*

and NETworks (COMSNETS), pages 579–584, 2026. doi:10.1109/COMSNETS67989.2026.11418176.

[19] S. Mondal and P. Bours. Continuous authentication using mouse dynamics. In *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*, pages 1–12, 2013.

[20] V. Nair, W. Guo, J. Mattern, R. Wang, J. F. O'Brien, L. Rosenberg, and D. Song. Unique identification of 50,000+ virtual reality users from head & hand motion data. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, pages 895–910. USENIX Association, 2023.

[21] K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt. Behavioural biometrics in VR: Identifying people from body motion and relations in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–12, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3290605.3300340.

[22] M. Pusara and C. E. Brodley. User re-authentication via mouse movements. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, VizSEC/DMSEC '04, pages 1–8, New York, NY, USA, 2004. Association for Computing Machinery. doi:10.1145/1029208.1029210.

[23] Rays Baseball Club LLC, Esports Foundry, and Rare Drop Co. THE 2025 RAYS GAMING 'SUNBURST INVITATIONAL' OFFICIAL RULES, 2025. Available at https://www.mlb.com/rays/fans/rays-gaming/rules (accessed on May 28, 2025).

[24] R. Shadman, A. A. Wahab, M. Manno, M. Lukaszewski, D. Hou, and F. Hussain. Keystroke dynamics: Concepts, techniques, and applications. *ACM Computing Surveys*, 57(11), 2025. doi:10.1145/3733103.

[25] N. Siddiqui, R. Dave, and N. Seliya. Continuous user authentication using mouse dynamics, machine learning, and minecraft. In *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages 1–6, 2021. doi:10.1109/ICECET52533.2021.9698532.

[26] V. R. d. Silva and M. D. Costa-Abreu. An empirical biometric-based study for user identification with different neural networks in the online game League of Legends. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2018. doi:10.1109/IJCNN.2018.8489164.

[27] C. M. Tey, P. Gupta, and D. Gao. I can be you: Questioning the use of keystroke dynamics as biometrics. In *Proceedings of the 20th Annual Network and Distributed System Security Symposium (NDSS '13)*, pages 1–16, 2013.

[28] J. Voris. Measuring how we play: Authenticating users with touchscreen gameplay. In K. Murao, R. Ohmura, S. Inoue, and Y. Gotoh, editors, *Mobile Computing, Applications, and Services*, pages 144–164, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-90740-6_9.

[29] R. Yampolskiy and V. Govindaraju. Behavioural biometrics: A survey and classification. *International Journal of Biometrics*, 1(1):81–113, 2008. doi:10.1504/IJBM.2008.018665.

[30] S. Yuen, J. D. Thomson, and O. Don. Automatic player identification in Dota 2. *arXiv preprint arXiv:2008.12401*, 2020. doi:10.48550/arXiv.2008.12401.

A Hyperparameter Search Range

Table 12 lists the hyperparameter search ranges for each base learner used for optimization using Optuna [2].

Table 12: **Hyperparameter search ranges.** The table defines the search spaces for each algorithm where numerical intervals represent ranges for integer or floating-point sampling, while bracketed lists denote categorical selections, thereby determining the specific sampling behavior using Optuna.

Base learner	Parameter name	Searching range
Random Forest	n_estimators	100 – 300
	max_depth	3 – 20
	min_samples_split	2 – 10
	min_samples_leaf	1 – 5
	max_features	[sqrt, log2]
SVM	kernel	[linear, poly, rbf, sigmoid]
	C	0.1 – 100.0
	gamma	1e-4 – 1.0
XGBoost	n_estimators	100 – 300
	learning_rate	0.01 – 0.2
	max_depth	3 – 10
	subsample	0.6 – 1.0
	colsample_bytree	0.6 – 1.0
	reg_alpha	0.0 – 1.0
	reg_lambda	0.0 – 5.0
LightGBM	n_estimators	100 – 300
	learning_rate	0.01 – 0.2
	num_leaves	16 – 128
	min_child_samples	5 – 30
	colsample_bytree	0.6 – 1.0
	subsample	0.6 – 1.0

B Evaluation Metrics

Let η denote the decision threshold on the classifier's output probability, where samples with score larger than η are classified as positive. Based on this threshold, we can count the number of true positives $\mathrm{TP}(\eta)$, true negatives $\mathrm{TN}(\eta)$, false positives $\mathrm{FP}(\eta)$, and false negatives $\mathrm{FN}(\eta)$.

With these values, we can define two commonly used metrics:

$$\mathrm{Precision}(\eta) = \frac{\mathrm{TP}(\eta)}{\mathrm{TP}(\eta) + \mathrm{FP}(\eta)}; \quad \mathrm{Recall}(\eta) = \frac{\mathrm{TP}(\eta)}{\mathrm{TP}(\eta) + \mathrm{FN}(\eta)}.$$

The precision-recall curve is then defined as the plot of $\mathrm{Precision}(\eta)$ against $\mathrm{Recall}(\eta)$ as η varies from 0 to 1, and the area under this curve is called the PR-AUC, which indicates the performance under the situation with imbalanced classes.

With $\mathrm{FP}(\eta)$, $\mathrm{TN}(\eta)$, $\mathrm{TP}(\eta)$, and $\mathrm{FN}(\eta)$, we can also define the false acceptance rate $\mathrm{FAR}(\eta) = \frac{\mathrm{FP}(\eta)}{\mathrm{FP}(\eta)+\mathrm{TN}(\eta)}$ and the false reject rate $\mathrm{FRR}(\eta) = \frac{\mathrm{FN}(\eta)}{\mathrm{FN}(\eta)+\mathrm{TP}(\eta)}$. There is a trade-off between these two rates as the threshold η varies, leading to the definition of the equal error rate (EER) as the point where the FAR and FRR are equal at a specific threshold $\eta^\star$: $\mathrm{EER} = \mathrm{FAR}(\eta^\star) = \mathrm{FRR}(\eta^\star)$. The EER can be interpreted as the performance that strikes a balance between security and applicability.