Multimedia Tools and Applications
https://doi.org/10.1007/s11042-022-13464-0

TRACK 4: DIGITAL GAMES, VIRTUAL REALITY, AND AUGMENTED REALITY

AI-enabled prediction of video game player performance using the data from heterogeneous sensors

Anton Smerdov[1] · Andrey Somov[1] · Evgeny Burnaev[1] · Anton Stepanov[1]

Received: 8 December 2021 / Revised: 9 June 2022 / Accepted: 6 July 2022

Abstract

The emerging progress of video gaming and eSports lacks the tools for ensuring high-quality analytics and training in professional and amateur eSports teams. We report on an Artificial Intelligence (AI) enabled solution for predicting the eSports player in-game performance using exclusively the data from sensors. For this reason, we collected the physiological, environmental, and the smart chair data from professional and amateur players. The player performance is assessed from the game logs in a multiplayer game for each moment of time using a recurrent neural network. We have investigated an attention mechanism improves the generalization of the network and provides a straightforward feature importance as well. The best model achieves Area Under the Receiver Operating Characteristic Curve (ROC AUC) score 0.73 in predicting whether a player will perform better or worse in the next 240 seconds based on in-game metrics. The prediction of the performance of a particular player is realized although their data are not utilized in the training set. The proposed solution has a number of promising applications for professional eSports teams and amateur players, such as a learning tool or performance monitoring system.

Keywords eSports · Machine learning · Neural networks · Sensor network · Video gaming

1 Introduction

Esports is a competitive video gaming where the single players or teams contend with each other to achieve a specific goal by the end of the game. The eSports industry has progressed a lot within the last decade [5]: a number of professional and amateur teams take part in numerous competitions where the prize pools achieve tens of millions of US dollars. Its global audience has already reached 474 mln. in 2021 and is expected to reach more than

✉ Andrey Somov
a.somov@skoltech.ru

[1] CDE, Skolkovo Institute of Science and Technology (Skoltech), Moscow, Russia

Published online: 23 August 2022

577 mln. by 2024 [49]. eSports industry includes so far a number of promising research and commercial directions, e.g., streaming, hardware, game development, connectivity, analytics, and training.

Apart from the growing audience, the number of eSports players and ‘Pro’-players (professional players or athletes), the players with a contract, has grown in the last few years [49]. It made the competition among the players and teams even harder, attracting extra funding for training process and analytics. The opportunity to win a prize pool playing a favorite game is very tempting for amateur players, and most of them consider a professional eSports career in the future. At the same time, *analytics and training* direction [27] is recognized as the most promising one as it includes the innovative research and business in artificial intelligence,[1] data/video processing, and sensing.

Although eSports is recognised in a number of countries as sport, it is still in infancy period: there is a lack of holistic training methodologies and widely accepted data analytics tools [48]. It makes it unclear how to improve the particular game skill except for spending the lion share of time in the game and watching how the popular streamers perform, and participating in trainings. Currently, there is a lack of tools providing feedback about the player performance and advising how to perform better [51]. It creates a huge potential for the eSports research in order to understanding the factors essential to win in a game. Considering the abundance of data available through the game replays, so-called ‘demo’ files, allow for replicating the game and performing fundamental analysis.

In terms of prediction and analytics, which is relevant to the research reported in this work, most of the current works in eSports rely exclusively on the in-game data analysis. However, using only in-game data for estimating the players’ performance is a limiting factor for providing helpful feedback to the team and players. While it can provide the primary information about the gamer’s traits and behavior, the huge amount of data from the physical world and captured by sensors [12] is omitted. Moreover, sensor data may be more suitable for the eSports domain since models trained on in-game data only quickly become obsolete when a new patch is released. Information about the player’s physiological conditions, e.g. heart rate, muscle activity, and movements can supplement logs obtained from in-game data to provide additional information for predictive models and potentially improve their performance. In addition, this can be used for selecting proper players for particular opponents and making a proper substitution during the game. Multimodal systems utilizing this information have already been explored for audio, photo, and video stimuli [7].

In this article, we report on predicting the eSports player performance in Counter Strike:Global Offensive (CS:GO) using the data collected from different sensors and recurrent neural network for data analysis. While there is a number of relevant research papers dealing with the prediction of a player skill in general [9, 70], there is still lack of research on the estimation of current player performance at a particular moment of time relying on various sensors and the data collected from professional players. This immediate prediction can provide the instantaneous feedback and can serve as a useful tool for the eSports team analysts and managers to monitor the current conditions of players. Another practical application is the real-time performance monitoring tool for eSports enthusiasts who want to progress towards the professional level and sign a contract with a professional team. Since playing a game is a high mental load and stress, we propose to use a multimodal system to record the players’ physiological activity (heart rate, muscle activity, eye movement,

[1] The Power of AI in eSports, https://medium.com/the-esports-analyst-club-by-itero-gaming/the-power-of-ai-in-esports-80edd6a4270e.

skin resistance, mouse movement), gaming chair movement, and environmental conditions (temperature, humidity, and CO_2 level). This data may help explain variations in gaming performance during the game and to identify which factors affect the performance the most.

Contribution of this work is threefold. (i) Experimental testbed and heterogeneous data collection from various sensors. The dataset is collected in collaboration with a professional eSports team. (ii) Investigation of the optimal neural network architecture for predicting a player performance and interpreting the obtained results. The model is characterized by the ability to predict performance of a new player not included in the training set. (iii) In terms of data analysis, a special emphasis is made on the current performance status of the player instead of considering the overall player skill. In this regard, the sensor data are used to evaluate the in-game performance in a First-Person Shooter (FPS) game discipline.

This paper is organized as follows: in Section 2 we introduce the reader to the relevant research in the area. Afterwards, the methods used in this research are presented in Section 3. Experimental results are demonstrated in Section 4. Finally, we provide concluding remarks in Section 5.

2 Related work

Wearable sensors and body sensor networks have been widely applied for assessing a human behaviour and activity recognition in many areas [64]. However, this approach has not been extensively used for assessing eSports players: typically, in-game data or data collected from computer keyboard and mouse have been analyzed so far. Due to this limitation the performance evaluation methods are limited as well.

This section is therefore divided into two parts: first, we overview relevant research in terms of data collection and activity recognition and, second, we discuss recent research on performance evaluation methods.

2.1 Activity recognition using sensors

Indeed, there is a lack of prior research utilizing sensors data to predict eSports player behavior. It happens due to young research domain in eSports. Recent research on using sensor data in eSports is limited to predicting the overall player skill or finding simple dependencies in the data. Also, the limited number of sensors is used (maximum three are reported in [32]). We utilize the findings from other studies to shape a useful set of sensors for extensive research.

Another challenge we addressed in this work, is the data collection from the eSports professional players using sensing technologies.

In [13] authors investigate the correlations between psychophysiological arousal (heart rate, electrodermal activity) and self-reported player experience in a first-person shooter game. Similar research about the relation between the player stress and game experience has been investigated in the Multiplayer Online Battlefield Arena (MOBA) genre [8]. The connection between the gaze and player skill is investigated in [65]. Mouse and keyboard data is a natural source of information about the players. Its relation with player performance in first-person shooters is covered in [9] for Red Eclipse and [32] for Counter-Strike: Global Offensive (CS:GO). Player performance can also be predicted by activity on a chair during the game [56] or in reaction to key game events [57], and these features were informative in our study as well.

In [58], the authors investigate how to predict whether the player wins the next encounter in League of Legends (LoL) using the sensor data. Their best model (Transformer network) achieves ROC AUC score (the area under the ROC curve (AUC), where ROC stands for the receiver operating characteristic) of 0.706 in predicting whether the player will win the encounter occurring 10 seconds later, and they built the system predicting the ‘player burnout’ with 73.5% precision and 88.3% recall. While this study also investigates how to use sensor data for eSports analytics, their research is focused on League of Legends discipline (MOBA genre) instead of CS:GO discipline (FPS genre) considered in this work. Also, the frequency of events in League of Legends and CS:GO varies, thus it affects the target construction and data processing (the smoothing, the number of outliers, the intensity/influence of noise).

Application of sensors for activity recognition tasks in eSports and sport is summarized in Table 1.

Apart from eSports, there is extensive research work carried out on data collection and activity recognition in other applications including sports, medicine, and daily activity monitoring. In sports, wearable sensing systems are used for detection and classification of training exercises for goalkeepers [23], assessing header skills in soccer [60]. Also, wearable systems were designed to classify tricks in skateboarding [22], classify popular swimming styles using sensors [69], and other activities in sports [67]. In terms of daily activity monitoring and medical applications, they have been studied for nearly three decades with the use of wearable sensors. Many medical studies deal with the investigation of human gait, for example, for patients with the Parkinson’s disease [6].

Table 1 Data collection using sensors in eSports and sport

Game discipline	Sensors	Target	Reference
FPS (Prey, Doom 3, Bioshock)	Heart rate monitor (HRM), electrodermal activity (EDA)	Player experience	[13]
MOBA (LoL)	Photoplethysmogram (PGP), galvanic skin response (GSR), inertial measurement unit (IMU), mouse, keyboard	Player stress	[8]
FPS (CS:GO)	Eye-tracker	Player skill	[65]
FPS (Red Eclips)	Mouse, keyboard	Player performance	[9]
FPS (CS:GO)	Mouse, keyboard	Player performance	[32]
FPS (CS:GO)	IMU	Player performance	[56, 57]
MOBA (LoL)	Electromyography (EMG),	Win prediction GSR, IMU	[58]
Soccer	IMU	Movement classification	[23]
Soccer	IMU	Head impact monitoring	[60]
Skateboarding	IMU	Tricks classification	[22]
Swimming	IMU	Swimming stroke phase segmentation	[69]

2.2 Performance evaluation methods

Professional players compete in the video gaming context in front of large audience and imagine the opportunity to win a huge prize pool. It requires their top form and ability to perform in stressful situation. Therefore, the players need their optimal mental, cognitive, and physical abilities. Training allows for evaluation of the players performance. However, eSports players do not have access to a comprehensive health management system. Findings from multidisciplinary study in domains like movement and cognitive science, as well as game research, could lead to the development of holistic body ‘n’ brain training methodologies for eSports athletes. Exergaming, which combines physical and cognitive training in a fun gaming environment, is a promising and unique training method. Exergames could serve as an alternative training option to comprehensively enhance gaming performance and overall health in eSports athletes, taking into account game design and training concepts as well as eSports particular requirements [46]. Another training method for eSports players is reported in [51]. The proposed method employs a network analysis of eye movement coordinates in conjunction with the crucial decisions to generate new information that neither method can produce on its own.

Most of the current research in eSports analytics relies exclusively on the in-game data collection and further analysis. It has been shown that information about kills, deaths, and other game events can help predict a game outcome in MOBA discipline for Dota 2 [27], League of Legends [41], and Rocket League [59]. Another opportunity to predict the game outcome in the MOBA related disciplines is based on the features extracted from players’ match history, as well as in-game statistics [74]. Players match history can also be used to create a rating system for predicting the matches outcome in the FPS genre [44].

As noticed earlier, the in-game data in eSports is widely used for analytic studies in the area. Drachen et al. consider clustering a player behavior to learn the optimal team compositions for League of Legends discipline to develop a set of descriptive play style groupings [14]. Research by Gao et al. [20] targets the identification of the heroes that players are controlling and the role they take. The authors have used classical machine learning algorithms trained on game data to predict a hero ID and one of three roles and achieve the accuracy ranging from 73% to 89% which depends on features and targets used. Eggert et al. [16] has continued the work by Gao et al. [20] and applied the supervised machine learning to classify the behavior of DOTA players in terms of hero roles and playstyles. Martens et al. [45] have proposed to predict a winning team analyzing the toxicity of in-game chat. In [66] authors used pre-match features to predict the outcome and analyze blowout matches (when one team outscores another by a large margin). The research reported in [10] describes the cluster evaluation, description, and interpretation for player profiles in Minecraft. The authors state that automated clustering methods based on game interaction features help identify the real player communities in Minecraft.

In many domains the skills and performance can be assessed and/or predicted based on sensors data [52]. The connection between physiological data and mental stress as well as the performance of drivers was demonstrated in [25]. The authors used electrocardiogram, electromyogram, skin conductance, and respiration data from 24 drivers. In sport, the data from the Inertial Measurement Unit (IMU) can be helpful for estimating volleyball athlete skill [68]. Tennis player performance can be assessed from the IMU data on the hand and chest [2], or on the waist, leg, and hand [1]. Similar techniques have been investigated for skill estimation in soccer [23], climbing [38], golf [76], gym exercising [37], and alpine skiing [47].

Another popular domain for skill assessment based on the sensor data is surgery. In [31] authors use IMU data to create a skill quantification framework for surgeons. Ershad et al. [17] have shown the connection between the surgeon skill and behavior information collected from IMU. Ahmidi et al. [3] developed a system using motion and eye-tracking data for surgical task and skill prediction. The authors have used hidden Markov models [15] to represent surgeon state which is similar to the method proposed in this work in Section 3. The connection between the surgeon actions and pressure sensors data has been investigated in [73].

Physiological data have also been used for predicting skill level and skill acquisition in working activities, such as mold polishing [36] and clay kneading [72], as well as dancing [71].

In this research, we use a number of wearable and local unobtrusive sensors used for data collection during the game session and further data analysis. It was carried out with respect to the player's needs.

3 Methods

In this section, we describe the sensors used in this research, data collection procedure, data pre-processing, and data analysis helping predict the player's performance. In Fig. 1 we present an overview of prediction system.

3.1 Sensors

In our work, we use three groups of sensors: *physiological* sensors, sensors integrated into a *game chair*, and *environmental* sensors. The sensor network architecture is shown in Fig. 2. The list of sensors used, their locations, and sampling rates are presented in Table 2. Further issues associated with sampling rates are discussed in Section 3.5.1.

Physiological data recorded:

- **Electromyography** (EMG) data as an indicator of muscle activity. EMG data are related to physiological tension affecting player's current state [40].
- **Heart rate** data received by a heart rate monitor on a chest. A timeline of heart rate data is connected to heart rate variability (HRV), and HRV is connected to mental stress and arousal [61], which might affect the rationality of player decisions. Research in

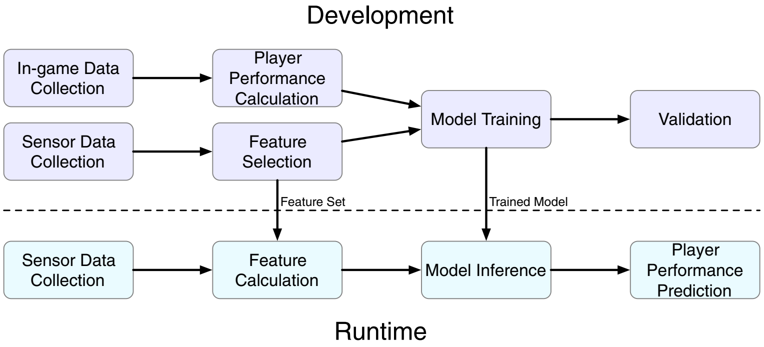

Fig. 1 An overview of prediction system during the development and runtime phases

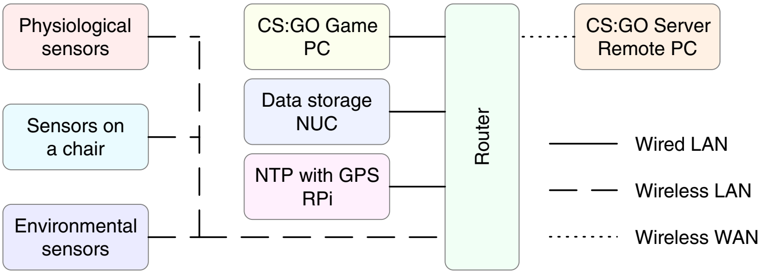

Fig. 2 Sensor network architecture

[13] indicates significant correlation between heart rate and self-reported gameplay experience.

- **Electrodermal activity** (GSR) or skin resistance data as a measure of person arousal [19]. This value is also connected with the stress level. Study [13] demonstrates the connection between GSR data and player experience.
- **Eye tracker** data of player gaze position on a monitor in pixel values. The player must check the minimap and other indicators on the screen to have relevant information about the game and, thus, make effective decisions [65].
- **Mouse movements** captured by a custom python script as a measure of the intensity of a player input. This data is an indirect indicator of the hand movement activity as well as the player skill [9]. Research in [32] shows that information about number of clicks and their duration in first-person shooter games improves the player skill prediction algorithm.

A 3-axial accelerometer and a 3-axial gyroscope present the sensors integrated into a game chair. We illustrate axes orientation for the chair in Fig. 3. Recorded data includes:

- **Linear acceleration** of a chair. It captures the player movements to the game table, parallel to it, chair height changes, and small oscillations possible in stress conditions. Behavior on a chair is connected with the player skill [56, 57].

Table 2 Sensors location and sampling rates

Sensor type	Sensor	Location	Sampling rate
Physiological Sensors	Electromyography sensor [53]	Elbow	70 Hz
	GSR sensor [54]	Fingers	70 Hz
	Heart rate monitor [21]	Chest	3 Hz
	Eye tracker	Under the PC monitor	90 Hz
	Mouse	Hand	250 Hz
Sensors on a Chair	3-axis accelerometer and 3-axis gyroscope [30]	The bottom of the chair	100 Hz
Environmental Sensors	CO_2 sensor [43]	Environment, ≈1m from the player	0.2 Hz
	Temperature sensor [28]		5 Hz
	Humidity sensor [29]		5 Hz

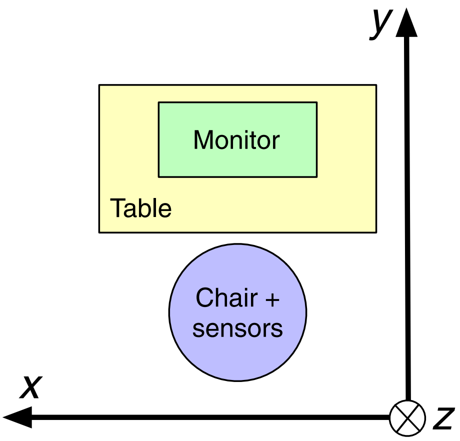

Fig. 3 Axes orientation for the chair

- **Angular velocity** of a chair. This data provides the information about the person's wiggling and spinnings on a chair.

Environmental data recorded:

- CO_2 **level**. High CO_2 level results in the reduction of cognitive abilities [4], thus directly affect the gaming process.
- **Relative humidity**. High level of relative humidity results in the reduction of neurobehavioral performance [77]
- **Environmental temperature**. Too warm conditions may affect the human performance [39].

3.2 Sensor network and synchronization

Apart from a number of heterogeneous sensors, the sensor network has a dedicated storage server (based on Intel NUC PC), gamer PC (a high-speed Intel I7 PC with the DDR4 (Double Data Rate) memory, and an advanced GPU card). For synchronization reasons, we have an Network Time Protocol (NTP) server with Global Positioning System and Precise Positioning Service (GPS and PPS) support. A high-speed wireless router connects all the devices in the network. PC with strict requirements to ping value (gamer PC, NTP server) has a wired connection to the router (Local Area Network, or LAN). The sensors have a wireless connection to the network as they are placed near the eSports athletes (Wireless LAN, or WLAN). The router has a low latency connection to the Internet (WAN). Proper synchronization of the sensors and gamer PC is essential for further data collection and analysis.

3.2.1 NTP server

At present, there are many options for building time-synchronous systems for industrial applications, we decided to realize the synchronization on our own local NTP server. Such server can be located close enough and characterized by the minimum delay in transmitting the packets over the network.

A single-board computer Raspberry PI 3B was selected as a server, and a GPS (MTK MT3333) signal was used as a source of reasonably accurate time. Raspberry PI was located

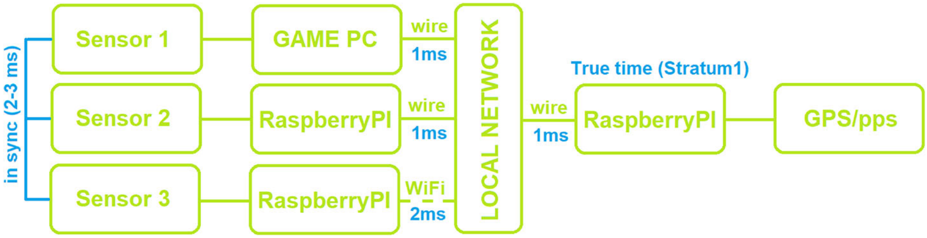

Fig. 4 Synchronization scheme

near the window for better satellite signal reception and connected to the local area network via a wired interface. The presence of a dedicated PPS signal acquired by a separate IO pin (GPIO) Raspberry PI made it possible to ensure time accuracy in the range of $10^{-5} - 10^{-6}$ s (time accuracy of $1 - 10$ us).

3.2.2 Sensors

The sensors in our network are deployed on Raspberry PI (RPI). The broadcast network “sync” command was sent to the sensors prior to measurements. After the command reception, a custom made script synchronized the local time to the local NTP server (Stratum 1) time on each RPI. Feedback status with the current time difference was also reported from every RPI to the local data storage PC. In this case, all RPI were synchronized before the measurement procedure starts. Time drift of local RPI time was measured: it is in the range 10–20 ms per hour. In this case, the sync command was repeated every 10 min. This allows us to have synchronized sensors all the time (Fig. 4).

3.2.3 Gamer PC synchronization

Performing the time synchronization on a gamer PC was another significant issue. The accuracy of the clock within 1 ms requires to meet a number of conditions.[2] In our experiment (taking into account the local time server Stratum 1 based on RPI), all the requirements were met with the exception of the ping value (it was <1 ms, instead of the required value <0.1 ms). However, it allows us to achieve the necessary synchronization accuracy.

In the case of proper registry settings[3] after some time, the drift is compensated by the internal Windows algorithms, and the clocks become synchronous with the time server (within 2–3 ms accuracy).

Upon synchronizing the hardware in the network, we start the data collection procedure.

3.3 Game selection

CS:GO is in the short-list of top eSports disciplines with the average numbers exceeding 500k and 1000k peak concurrent players.[4] Major tournament has up to 1 mln. USD prize

[2] https://docs.microsoft.com/en-us/windows-server/networking/windows-time-service/support-boundary

[3] https://docs.microsoft.com/en-us/windows-server/networking/windows-time-service/configuring-systems-for-high-accuracy

[4] https://steamcharts.com/app/730

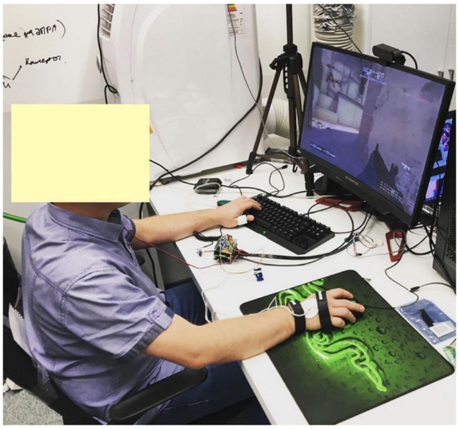

Fig. 5 Experimental testbed

pool.[5] CS:GO is a FPS game. Personal skills, team management and coordination is a key for the ability to win in this discipline. Game Application Programming Interface (API) by itself and sensor data recorded during the game is a rich dataset for analysis, players characterization and prediction analytics by various methods. It is important to use non-invasive type of sensors during data collection (HRM/eye-tracking/mouse/keyboard/chair-IMU), so pose or movement would not restrict the player. In case of GSR and body-IMU sensors location should be selected carefully (GSR on left hand fingers, IMU on right hand elbow) in this case data will not be treated by noise or false body movements (players usually use left hand for controlling keyboard and right hand for controlling mouse).

3.4 Data collection

CS:GO is one of the most popular computer games in the world. It is a team game in 5x5 format (5 players per team). Different international tournaments have various game scenarios, one of the most popular is best of 3 matches (every match consists of 30 rounds, the winning team should complete the 30th round with +2 score, in case of less distance in score an additional round is played).

We invited 21 participants to play the FPS CS:GO for 30–35 min. We note here that six pro-players took part in this experiment. Professional player is a player who has a contract with a team participating in tournaments and receives a salary for his game. All the participants were informed about the project and the experimental details. Every participant signed a written consent form, which allowed recording physiological and in-game data. Then players were equipped with the sensors for data collection. We did not receive any complaints about the uncomfortable gaming experience because of the sensors. The experimental testbed is shown in Fig. 5. 2 participants out of 21 were left-handed. For those participants we switched the locations of GSR and EMG sensors.

Players needed to play *Deathmatch* mode of CS:GO with other random players of normal skill on an open CS:GO server. Participants played at different times during several consecutive days and didn't interact with each other. The example of the game screen interface is shown in Fig. 6. Deathmatch mode is a special game mode where people play against all others (1 to many), and it's focused on personal skills for shooting and movement. In this

[5] https://www.esports.net/news/csgo-major-winners/

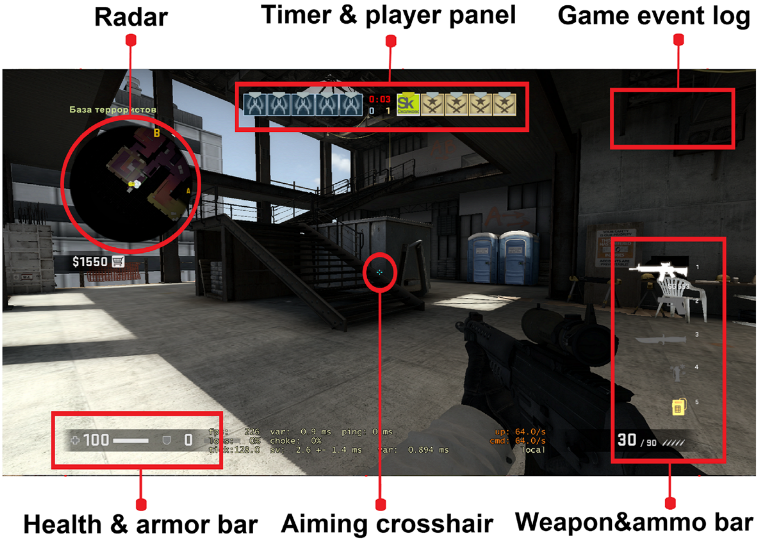

Fig. 6 CS:GO screen interface. Most of the time the players watch at the aiming crosshair

mode the goal of each player is to achieve as many kills of other players as possible and to minimize the number of their own deaths. When the player is killed it immediately respawns in a random location in the game. All participants were instructed about the controls and the goal of the game. This mode is often used by eSports players in their training routine. After the game had been finished, we saved the replays for the future game events extraction.

Collected data samples for the 5-min intervals for two players are shown in Fig. 7. For the reader convenience, we color the intervals in the 1-second vicinity of kill and death events.

It is clear that for the data from some physiological indicators, e.g. skin resistance or heart rate, there are global and local trends, which might align with changes in the player's efficiency in the game. Another point is that the players usually do not move a lot on a gaming chair. However, they can change their posture from time to time, and this event is captured by the IMU on the chair and also might be connected with the game events and player performance.

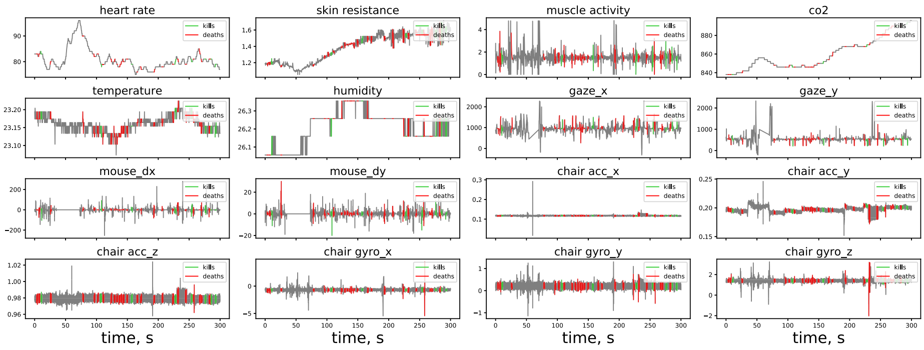

Fig. 7 5-minutes data samples for one player. Green and red coloring correspond to *kill* and *death* events, respectively

3.5 Data pre-processing

To get rid of the noise and occasional outliers, we have clipped all the data by 0.5 and 99.5 percentiles and smoothed the data by 100 ms moving window. We have also reparametrized gaze, mouse, and muscle activity signals. Mouse signal has been converted from x and y increments to Euclidian distance passed to reflect the mouse speed; gaze data has been transformed from x and y coordinates to Euclidian distance passed; muscle activity signal has been changed to L1-distance to the reference level for the player in order to represent the intensity of muscle tension. For 3.7% data missed we have used linear interpolation to fill in the unknown values since it provides a stable and accurate approximation.

3.5.1 Sensor data resampling

In order to predict the player performance at each moment of time, it is convenient to resample the data from all the sensors to the common sampling rate. This helps apply the proposed data analysis for discrete time series predictions, such as hidden Markov models [15] or recurrent neural networks [63].

However, data from different sensors has different underlying nature and should be resampled accordingly. While it is reasonable to average the data within a time step interval for heart rate, skin resistance, muscle activity, environmental data, chair acceleration and rotation, averaging is not applicable for the gaze movement and mouse movement data. The reason behind it is that we are interested in the total distance passed within the time step instead of the average distance passed per measurement. The total distance does not depend on a number of samples, but only on their sum.

Resampling introduces an important hyperparameter *time step*. Throughout the manuscript we refer to it is as $\mathrm{d}t \in \mathbb{R}$. Big time step values, e.g. 5 min, are not meaningful for our problem since we need to extract the relevant information about the player. On the other hand, too small timestamp, e.g 0.1 s, may lead to an excessive number of observations and noisier data. Indeed, the resampling time step should not be smaller than the time between the measurements for the majority of sensors.

After converting the sampling rates to the common value, we obtained a 15-dimensional feature vector for each moment of time. Further in the paper we will refer to this feature vector as $\boldsymbol{x}(t) \in \mathbb{R}^n$. Its components are described in Table 3.

Table 3 Feature vector components

Feature group	Feature name
Physiological Activity	Heart Rate
	Muscle Activity
	Skin Resistance
	Gaze Movement
	Mouse Movement
	Mouse Scroll
Chair Movement	Linear Accelerations along x, y, z axes
	Angular velocity along x, y, z axes
Environment	CO_2 level
	Temperature
	Humidity

3.6 Player performance evaluation

There is no generic player effectiveness metric for the majority of eSports disciplines. The most popular evaluation metric for FPS and MOBA games is Kill Death Ratio (KDR) [55]. It equals the number of kills divided by the number of deaths for the time interval. If KDR $>$ 1, it means that the player performs skilfully, or, at least, better than some players on a game server do. Otherwise, the player most likely underperforms compared to other players.

KDR takes values from 0 to $+\infty$ which is not a clear range for prediction. When the player performs skilfully and has many kills and few deaths, KDR is fluctuating drastically because of division by a small number. In opposite, if there are many deaths and few kills, KDR is around 0 and changes slowly. This drastic inconsistency in the target creates difficulties for training machine learning algorithms. One possible solution could be to apply logarithm to KDR, but this does not solve the issue with the scale, because logarithm takes the values from $-\infty$ to $+\infty$. Another possible solution is to check how many potential targets the player hits, but, to the best of our knowledge, there is no technical opportunity to track potential in-game targets.

We propose a more numerically stable target value, which equals the proportion of kills for the player. More precisely,

$$p_\tau(t) = \frac{k_\tau(t)}{k_\tau(t) + d_\tau(t)}, \tag{1}$$

$$k_\tau(t) = K(t+\tau) - K(t), \tag{2}$$

$$d_\tau(t) = D(t+\tau) - D(t), \tag{3}$$

where $p_\tau(t) \in \mathbb{R}$ is the proportion or performance and equals the proportion of kills for a player at the moment $t \in \mathbb{R}$ considering the kills and death in the next $\tau \in \mathbb{R}$ seconds. In other words, it is the ratio of kills in the next τ seconds. $K(t) \in \mathbb{R}$ and $D(t) \in \mathbb{R}$ are the total number of kills and deaths at the moment t; therefore $k_\tau(t) \in \mathbb{R}$ and $d_\tau(t) \in \mathbb{R}$ equals the number of kills and deaths within the interval $[t, t+\tau]$. $p_\tau(t)$ varies from 0 to 1 and has higher values for skilful players. Bounding by 0 and 1 helps efficiently train the machine learning algorithms that are sensitive to the target scale.

The important hyperparameter introduced above is τ. Essentially, it is a window size for which the information about the future player performance is aggregated. Small values of the hyperparameter like 1 s lead to noisy target values, while large values like 10 min neglect the subtle yet important changes in the player performance. τ is commonly referred to as a *forecasting horizon*.

$p_\tau(t)$ is a well-defined metric for evaluating the player effectiveness, and it is possible to predict this value directly, thus considering the problem as a regression problem. However, it is unclear how to interpret the quality of regression results in an understandable and interpretable way. Formulating the problem in classification terms helps measuring the quality of prediction by more comprehensive classification metrics like accuracy, ROC AUC, and others. These metrics are much easier to compare with results obtained on other data or by other models.

The natural way to claim if a person plays skilfully or a person underperforms at the moment is to compare the current performance with his or her average performance in the past. It is important to consider the past events only to avoid overfitting. Formally:

$$y_\tau(t) = [p_\tau(t) > \overline{p_\tau(t)}], \tag{4}$$

$$\overline{p_\tau(t)} = \frac{\sum_{t'<t} p_\tau(t')}{\sum_{t'<t} 1}, \tag{5}$$

where $y_\tau(t) \in \{0, 1\}$ and equals 1 in case of good game performance and 0 otherwise; $\overline{p_\tau(t)}$ is an average player performance in the past.

Figure 8 demonstrates how the kills ratio $p_\tau(t)$ and corresponding binary target $y_\tau(t)$ changes over time for three forecasting horizons.

The substantial advantage of using $y_\tau(t)$ instead of $p_\tau(t)$ is target unification between the players. 0 and 1 values of $y_\tau(t)$ have the same meaning for all players and imply underperformance and good current performance, respectively. That is not the case for raw values of $p_\tau(t)$, because the same values of $p_\tau(t)$ may be good for one player, but not satisfactory for another one. For example, 0.5 kills ratio may be an achievement for a newbie player, but a failure for a professional player.

Another justification of using $y_\tau(t)$ as a target is robustness to the skill of other players on a server. Player's score $p_\tau(t)$ may be too low in absolute value because of the strong opponents, but target $y_\tau(t)$ is robust because it evaluates the performance within one game. The motivation for this target is to unify the target variable for all players and to provide in advance an immediate feedback for a player or for a manager that something is going wrong.

Predicting future player performance $y_\tau(t)$ is essential for coaches and progressing players as it provides the quick feedback on players' actions. That might help identify the inevitable failures in player performance, e.g. burnout, fatigue, etc., in advance and take measures to help the player to recover or even to change the player during the eSports competition. This target is also helpful for learning purposes: although a person plays skilfully or a person underperforms, it helps find the moments when the player performs a bit better or worse than average.

Despite we formulate the performance metric in terms of kills/deaths, the metric is directly applicable to the majority of First-Player Shooters, as well as other games including kills/deaths. The performance metrics for these and other games can be calculated in other terms (such as gold, scores, progress, etc.), while the data processing and algorithms may be the same.

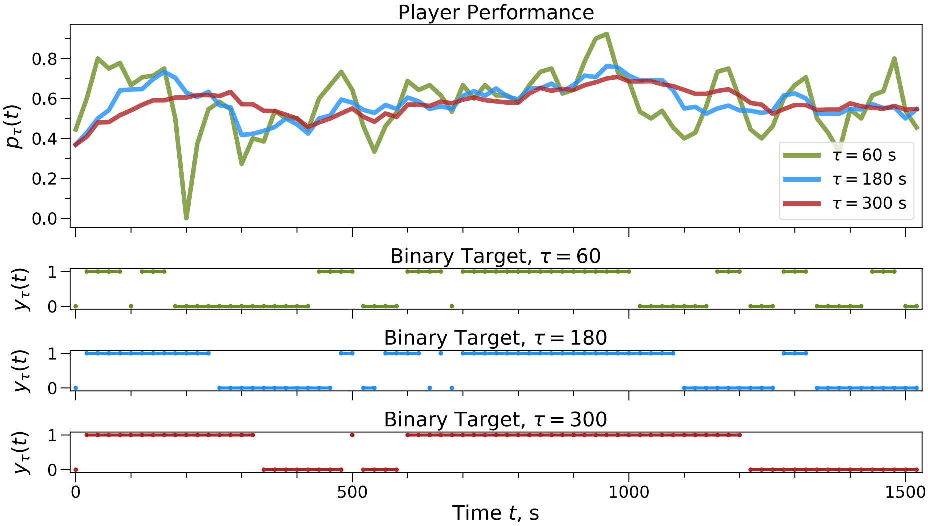

Fig. 8 Player performance as the kills proportion $p_\tau(t)$ and its binarization for three forecasting horizons τ for a player in a dataset

3.7 Predictive models

We trained four models for predicting a player performance using the data from sensors: baseline model, logistic regression, recurrent neural network, and recurrent neural network with attention. In this section, we describe these methods in detail. The output for all models is the probability that the person will play better in some fixed period in the future. All the models are evaluated by ROC AUC score discussed in Section 3.8.2.

3.7.1 Baseline

Before training a complex model, it is crucial to set up a simple baseline to compare it with. A common practice in time series analysis is to establish a baseline model using a current target value as a future prediction. For our problem, the baseline uses average player performance in the last τ seconds as a prediction. In other words, baseline prediction is $p_\tau(t)$. This prediction is correct because $p_\tau(t)$ takes values from 0 to 1, and then they treated as probabilities by the algorithm used to calculate the ROC AUC metric.

3.7.2 Logistic regression

Logistic regression [35] is a simple and robust linear classification algorithm. It takes a feature vector $\boldsymbol{x}(t) \in \mathbb{R}^n$ as an input at the moment of time t and provides the probability $y(t)$ equals 1 as an output:

$$P(y_\tau(t) = 1|\boldsymbol{x}(t)) = \frac{1}{1 + \exp(-\langle \mathbf{w}, \boldsymbol{x}(t)\rangle + b)}, \tag{6}$$

where $\mathbf{w} \in \mathbb{R}^n$ is the learnable weights vector, $b \in \mathbb{R}$ is the bias term. In our study, the dimensionality is $n = 15$ since we used 15 values from sensors. Logistic regression can capture only linear dependencies in the data because the feature vector $\boldsymbol{x}(t)$ involved in dot product with vector $\mathbf{w}$ only.

3.7.3 Recurrent neural network

A neural network can be considered as a nonlinear generalization of logistic regression. In this subsection, we first describe the essential components used in the network and then describe the entire architecture.

Recurrent Neural Network Background

A Recurrent Neural Network (RNN) is a network that maintains an internal state inherent to some sequence of events. It is proven to be efficient for discrete time series prediction [63]. One of the simplest examples of RNN is a neural network with one hidden layer.

Denote the sequence of input features and targets as $\Big\{\boldsymbol{x}\big(0\big), \boldsymbol{x}\big(\mathrm{d}t\big), \ldots, \boldsymbol{x}\big((N-1)\mathrm{d}t\big)\Big\}$ and $\Big\{y\big(0\big), y\big(\mathrm{d}t\big), \ldots, y\big((N-1)\mathrm{d}t\big)\Big\}$ respectively, where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the n-dimensional feature vector for the moment t, $y(t) \in \mathbb{R}$ is the corresponding target, $\mathrm{d}t$ is the time step used for discretization, $N \in \mathbb{N}$ is the total number of steps.

At each moment t, the recurrent network has an m-dimensional hidden state vector $\mathbf{h}(t) \in \mathbb{R}^m$ which is calculated using the current input $\boldsymbol{x}(t)$ and the previous hidden state $\mathbf{h}(t - \mathrm{d}t)$:

$$\mathbf{h}(t) = \tanh\Big(\mathbf{W}\boldsymbol{x}(t) + \mathbf{U}\mathbf{h}(t - \mathrm{d}t) + \mathbf{b}\Big), \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{m\times n}$, $\mathbf{U} \in \mathbb{R}^{m\times m}$, $\mathbf{b} \in \mathbb{R}^m$ are the learnable matrices and the bias vector.

The intuition behind using the RNN architecture might be in considering the hidden state of the network as a current state of the player represented by the data from sensors. The state is a vector with many components and some of them can present how skilfully the person will play. The final prediction $\hat{y}_\tau(t)$ at the moment t is calculated by the feed-forward network consisting of 1 or more linear layers:

$$\hat{y}_\tau(t) = f(\mathbf{h}(t)), \tag{8}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ is the function corresponding to the feed-forward network. We used a sigmoid function as a final activation for the network to ensure $\hat{y}_\tau(t) \in [0, 1]$, so $\hat{y}_\tau(t)$ has a meaning of probability.

Gated Recurrent Unit

More advanced modifications of the recurrent layer include the Gated Recurrent Unit (GRU) [11] and Long Short-Term Memory (LSTM) [26]. Both of them utilize the gating mechanism to better control the flow of information. LSTM architecture incorporates an input, output, and forget gates and a memory cell, while a simpler GRU architecture uses the update and reset gates only. We found GRU performs better in our task, so we formally define it as follows:

$$\mathbf{h}(t) = (1 - \mathbf{z}(t)) \odot \mathbf{h}(t - \mathrm{d}t) + \mathbf{z}(t) \odot \tilde{\mathbf{h}}(t), \tag{9}$$

$$\tilde{\mathbf{h}}(t) = \tanh(\mathbf{W}_h\boldsymbol{x}(t) + \mathbf{U}_h(\boldsymbol{r}(t) \odot \mathbf{h}(t - \mathrm{d}t)) + \boldsymbol{B}_h), \tag{10}$$

$$\mathbf{z}(t) = \sigma(\mathbf{W}_z\boldsymbol{x}(t) + \mathbf{U}_z\mathbf{h}(t - \mathrm{d}t) + \boldsymbol{B}_z), \tag{11}$$

$$\boldsymbol{r}(t) = \sigma(\mathbf{W}_r\boldsymbol{x}(t) + \mathbf{U}_r\mathbf{h}(t - \mathrm{d}t) + \boldsymbol{B}_r), \tag{12}$$

where $\mathbf{z}(t)$ and $\boldsymbol{r}(t)$ are the update and reset gates, $\mathbf{W}_h$, $\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{U}_h$, $\mathbf{U}_z$, $\mathbf{U}_r$, $\boldsymbol{B}_h$, $\boldsymbol{B}_z$, $\boldsymbol{B}_r$ are the learnable parameters, $\odot$ is Hadamard product, σ is the sigmoid function.

Attention Mechanism

A popular technique for improving the network quality and interpretability is an attention mechanism. Temporal attention can help emphasize the relevant hidden states from the past. Input attention helps to select the essential input features. It is also possible to combine both of them [50]. Since the proposed GRU model uses only one previous hidden state for prediction, it makes no sense to use the temporal attention. However, the input attention can be used.

The attention layer provides the weights vector $\boldsymbol{\alpha}(t) \in \mathbb{R}^n$, which is applied to a vector $\boldsymbol{x}'(t) \in \mathbb{R}^n$ by the element-wise multiplication:

$$\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}'(t) \odot \boldsymbol{\alpha}(t). \tag{13}$$

This operation demonstrates important components in the vector $\boldsymbol{x}(t)$ while decreasing the contribution of its non-relevant components. Typically, $\boldsymbol{\alpha}(t)$ components are bounded by 0 and 1 and produced by another linear layer integrated into the network. In order to consider both the current and the previous data from sensors, the attention layer takes current measurements $\boldsymbol{x}(t)$ and hidden state $\mathbf{h}(t)$ produced by GRU:

$$\boldsymbol{\alpha}(t) = \sigma(\mathbf{W}_a[\mathbf{h}(t - \mathrm{d}t), \boldsymbol{x}(t)] + \boldsymbol{B}_a), \tag{14}$$

where the $\mathbf{W}_a \in \mathbb{R}^{(m+n)\times m}$, $\boldsymbol{B}_a \in \mathbb{R}^m$ are trainable parameters.

The intuition behind the input attention mechanism is a feature selection. Based on the current input and hidden state, it can help ignoring the uninformative features for each moment of time and keep relevant features unaltered. The ability to provide the time-dependent feature importance is a significant advantage compared to other methods, which can either provide feature importance for one particular moment or provide it only for the whole time series.

Recurrent Neural Network With Attention Architecture

The neural network architecture is shown in Fig. 9. First, the sensor data are processed by the dense layers for each feature group. Then the attention block is applied to amplify a signal from the features relevant at the moment. The resulting vector goes through the GRU cell to update the hidden state $\mathbf{h}(t)$. This hidden state is saved for further iterations and goes through a feed-forward network to form the eventual prediction $\hat{y_\tau}(t)$. The total inference time is about 5 ms on a CPU.

The network was trained by the truncated backpropagation through time [62] technique designed for RNNs and Adam optimizer [34] with learning rate warmup [42] technique to improve the convergence. For attention and feed-forward networks, we used 1 and 2 linear layers, respectively, with ReLU nonlinearity. This activation function can improve the convergence and numerical stability [24]. To validate whether the attention mechanism helps improve network performance, we also trained another network without the attention block.

GRU cell is a crucial part of the network architecture. It helps accumulate information about previous player states, so the network can use the retrospective context for prediction. This information is stored in the hidden layer of the GRU cell. According to the experiments, 8 neurons in the hidden layer works for our problem the best. Too few neurons caused low predictive power, while too many led to overfitting.

The motivation to use separate linear layers for three feature groups is to combine more complex features from the sensors data and to preserve the disentangled feature representation for the attention layer. This is an analog of grouped convolutions [75] in convolutional networks. For the same reasons, we applied the attention to each feature group separately, thus having a 3-dimensional attention vector at each moment of time.

3.8 Validation

3.8.1 Training and evaluation process

In order to correctly estimate the generalization capabilities of classical machine learning algorithms, we used repeated cross-validation [33]. In particular, we randomly split 21

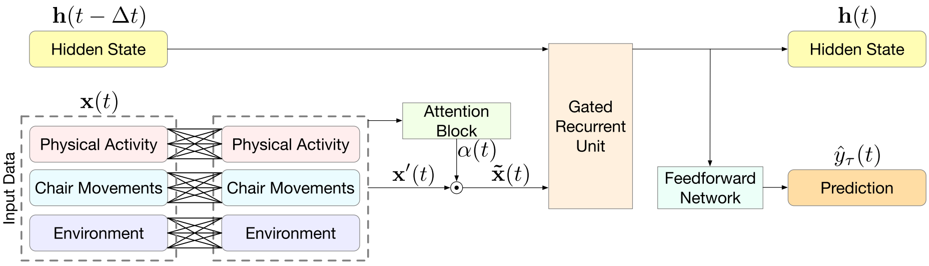

Fig. 9 Recurrent neural network architecture

players into the train and test groups of 16 and 5 players, respectively, and trained/evaluated algorithms on these data. We repeated random train/test splits and re-trained algorithms 100 times in total and averaged results across splits. This procedure helps lowering the variance in evaluation and get more reliable results.

For neural networks, we also used a validation set, so we randomly split players into train, validation, and test sets with 11, 5, and 5 players, respectively. The neural network has been trained until the error on the validation set is not improving for 5 epochs (early stopping procedure). One training epoch comprises of 20 batches. Each batch consists of input features and targets for all the time steps for a randomly selected player from the train set. To minimize the randomness in the evaluation results, each network has been trained 15 times with random weight initializations and train/val/test split. In both cases, the input features for train, test, and validation sets are normalized based on the mean values and standard deviations calculated on the train set.

3.8.2 Evaluation metric

Due to construction, the target is balanced: 50.1% belong to the positive class and 49.9% to the negative class. The common metric for classification evaluation is the area under the receiver operating characteristic curve (ROC AUC) [18]. It ranges from 0 to 1 with the 0.5 score for random guessing. Higher values are better.

For proper evaluation, we first calculated the ROC AUC scores for each individual participant in the train, validation, or test sets, and then averaged the results. That is the proper evaluation because it estimates at what extent the model can separate the high/low performance conditions for one individual participant and does not benefit from separating the participants between themselves in the case when the metric is calculated on predictions for all the participants.

4 Results

4.1 Performance of algorithms

We have trained the predictive models for several time steps dt (see Section 3.5.1) and forecasting horizons τ (see Section 3.6) combinations. According to our experiments, reasonable ranges are from 5 to 30 s for time step, and from 60 to 300 s for the forecasting horizon. The average results for the neural network with attention are shown in Table 4.

Table 4 Experimental results for the neural network with the attention for varying time steps and forecasting horizons. Results are reported in ROC AUC score

Forecasting horizon τ, s	Time step dt, s					
	5	10	15	20	24	30
60	0.602	0.658	0.632	0.617	0.645	**0.689**
120	0.609	0.621	**0.682**	0.631	0.680	0.657
180	0.611	0.622	0.652	**0.711**	0.682	0.642
240	0.667	0.661	0.654	**0.726**	0.724	0.703
300	0.619	0.667	0.721	0.680	0.717	**0.724**

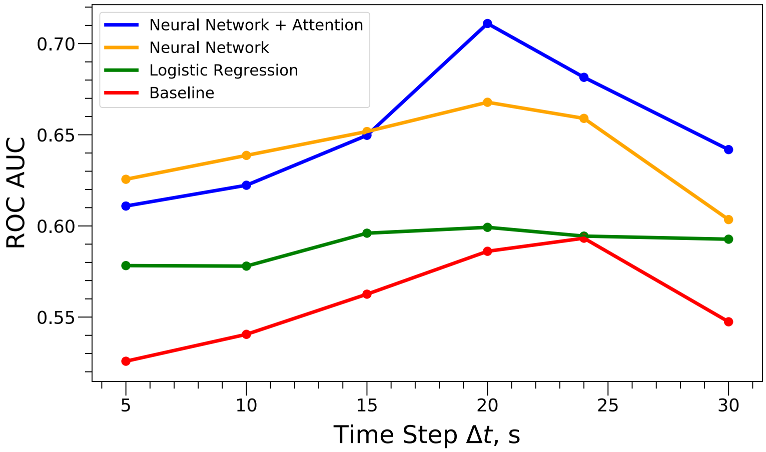

Fig. 10 Algorithms performance w.r.t. discretization time step dt with fixed forecasting horizon $\tau = 180$ s

According to Table 4, the best time step value is about 20 s, and the best forecasting horizon for the model is from 3 to 5 min. In other words, the optimal way to predict the player behavior is to aggregate the sensors data every 20 s and make a prediction for the next 3–5 min.

We have compared the performance of the algorithms described in Section 3.7 with respect to the time step values with fixed forecasting horizon τ equal 180 s. The results are shown in Fig. 10. There is a clear peak near 20–25 s for all the methods. The neural network consistently outperforms the logistic regression and the baseline model. The use of the attention block helps increase the model score.

Figure 11 demonstrates the relation between the forecasting horizon and algorithms performance with the time step dt equal to 20 s. The neural network outperforms other methods and achieves the maximum performance for forecasting horizons in the range from 3 to 5 min. The attention block helps to improve the model.

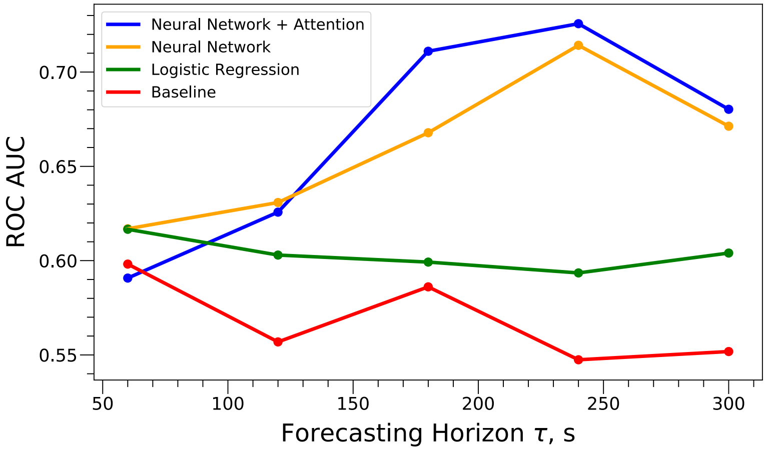

Fig. 11 Algorithms performance w.r.t. forecasting horizon τ with fixed time step d$t = 20$ s

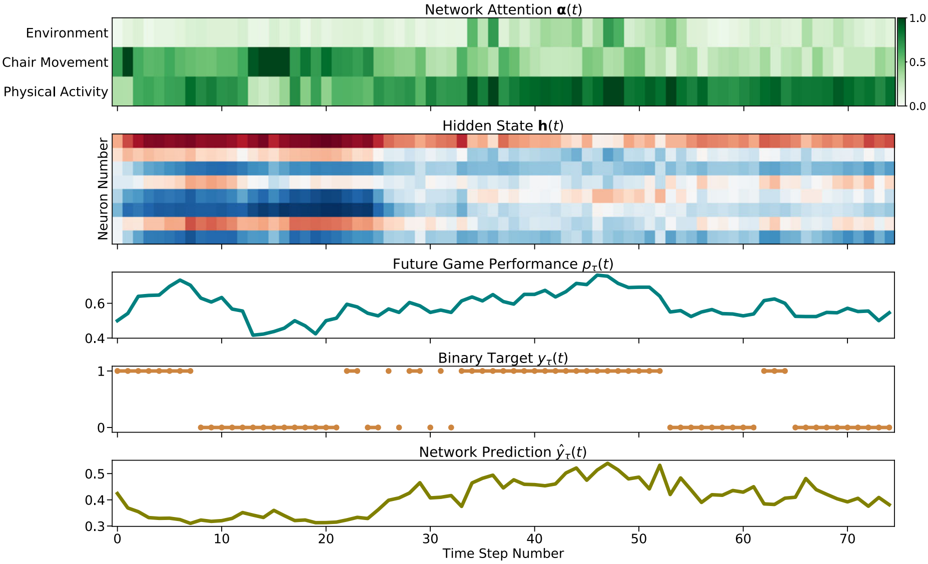

Fig. 12 Visualization of attention weights $\boldsymbol{\alpha}(t)$, hidden state $\mathbf{h}(t)$, game performance $p_\tau(t)$, binary target $y_\tau(t)$ and network prediction $\hat{y}_\tau(t)$ w.r.t. time step number for a player in the test set

4.2 Feature importance

In order to interpret the neural network predictions, we calculated the feature importances and visualized predictions of a pretrained network and its internal state for the discretization step equal to 20 s and the forecasting horizon equal to 3 min. Figure 12 shows how attention, network hidden state, target, and network prediction change over time. Clearly, the importance of different features varies over time, and periodically the data from some sensors is non-relevant. The network-hidden state, which can be treated as a player state, also varies during the game.

To calculate the feature importance, we trained 100 instances of the neural network with random weight initializations and the train/val/test splits, and the averaged attentions on the test set for the best epoch of each neural network. Afterwards, we averaged the results across all networks. The results are shown in Table 5.

Information about physiological activity such as heart rate, muscle activity, hand movement, and so on is the most relevant for the network. It is worth noting that all the feature groups have considerable importance, thus contribute to the overall prediction. Since the features in each feature group are mixed, we can not estimate the feature importance for every raw feature.

Table 5 Mean neural network attention for feature groups

Feature group	Mean attention
Physiological Activity	0.228
Chair Movement	0.207
Environment	0.197

4.3 Discussion

Experimental results have shown practical feasibility of predicting the eSports player's behavior using only data from sensors. The system updates the prediction several times per minute, so this interactivity is enough for potential users like eSports managers or professional players to understand that something goes wrong or to get a quick feedback. Feedback from the algorithm about the performance prediction and feature importance may suggest the users to change their gaming behavior. For example, an eSports manager may understand in time that poor results of the team are connected with the stuffy environmental conditions and adjust the air conditioning. Users may find information from the system useful to make the radical decisions, e.g., changing a player or gaming equipment (computer mouse, display, game chair, etc.). Negligible model inference time and data collection time on a PC proves the principal feasibility of deploying the model on the edge devices, some of which might be designed specifically for neural network operations. However, the model retraining still requires high computational capability.

An advantage of multi-sensor approach, as opposed to single-sensor approach in the majority of prior work, is adaptiveness. If a player finds some sensors disturbing or just wants to save money, then these data might be excluded from the prediction, although additional research on sensor importance is needed. For instance, EEG headset might be too expensive or incompatible with player's headphones, or GSR sensor on a hand can be disturbing for some people.

Regarding machine learning methods applied, we have found neural network with attention mechanism performing the best. It outperforms the baseline, which does not use the sensor data, as well as logistic regression and regular neural network, which use sensor data. That says that sensor data provide a signal about player performance, and the internal state of recurrent neural network combined with attention can catch this dependency better than alternatives.

We have found that 20 s time step and 3–5 min behavior forecasting horizon are the most natural parameters for CS:GO, but potential users can set up any other hyperparameters depending on their scenario. In the study [58] focused on League of Legends, authors find time step 1 s and forecasting horizon 10 s close to optimal for the game, although forecasting horizons up to 90 s are still reasonable. The difference in optimal forecasting horizons can be explained by higher frequency of kill/death/assist events in League of Legends, thus less time is needed to estimate player performance.

Importantly, the dataset includes data from professional eSports players, which is hard to obtain because of their availability and business. Significant advantage caused by the diversified training dataset is universality for the player skill level enabling a wide range of potential users. Larger and more diversified dataset could improve the performance of prediction.

In terms of research community, this research contributes towards a holistic data analysis system for eSports players [46].

4.4 Limitations

The limitations of the study is a small number of participants involved, the fuzziness of the definition of player's performance, and overrepresentation of data from males and young people in the dataset. Future work includes more diverse and extensive data collection with more subjects recorded, and the investigation of better metrics of players' performance.

These would allow researchers to utilize more complex machine learning methods and to develop more reliable and robust models.

5 Conclusions

In this article, we have reported on the AI-enabled system for predicting the performance of eSports players using only the data from heterogeneous sensors. The system consists of a number of sensors capable of recording players' physiological data, movements on the game chair, and environmental conditions. Upon data collection we have processed them into time series with meaningful sensors features and the target extracted from the game events. The Recurrent Neural Network (RNN) demonstrated the best performance comparing to baseline and logistic regression. The best model achieved ROC AUC score 0.73 in predicting whether a player will perform better or worse in the next 240 seconds in terms of in-game metrics based on KDR score. Other forecasting horizons from 60 to 300 seconds demonstrated similar ROC AUC scores. The important feature of the model is the ability to predict performance of a new player not included in the training set. Application of the attention mechanism for RNN has helped to interpret the network predictions as well as to extract the feature importance. Our work showed the connection between the player performance and the data from sensors as well as the possibility of making a real-time system for training and forecasting in eSports.

We have also investigated potential applications of the proposed AI system in the eSports domain. Given the growth of eSports activity due to the coronavirus pandemic in 2019–2021 and rapid development of consumer wearable devices within the last years, this work shows the prospectives of full-fledged research in the intersection of these two fields. Moreover, the model trained on the eSports domain can be transferred to other domains using domain adaptation methods to estimate user's performance similar to estimating in-game performance.

Considering the hundreds of millions of active gamers in the world and widespread of wearables, crowdsourcing data collection is a promising way to collect the data on a global scale. We also see potential improvements in our system with computer vision methods. Emotion recognition and pose estimation techniques applied to data collected from web camera can provide more information about the current state of a player.

Acknowledgments Authors would like to thank professional eSports team DreamEaters for fruitful discussions and data collection.

Funding The reported study was funded by Russian Foundation for Basic Research (RFBR) according to the research project No. 18-29-22077.

Data Availability The data that support the findings of this study are available from HeadKraken LLC but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of HeadKraken LLC.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Preprint of this manuscript is available at https://arxiv.org/pdf/2012.03491.pdf

References

1. Ahmadi A, Rowlands DD, James DA (2006) Investigating the translational and rotational motion of the swing using accelerometers for athlete skill assessment. In: SENSORS, 2006 IEEE. IEEE, pp 980–983
2. Ahmadi A, Rowlands D, James D (2009) Towards a wearable device for skill assessment and skill acquisition of a tennis player during the first serve. Sports Technol 2(3–4):129–136
3. Ahmidi N, Hager GD, Ishii L, Fichtinger G, Gallia GL, Ishii M (2010) Surgical task and skill classification from eye tracking and tool motion in minimally invasive surgery. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 295–302
4. Allen JG, MacNaughton P, Satish U, Santanam S, Vallarino J, Spengler JD (2016) Associations of cognitive function scores with carbon dioxide, ventilation, and volatile organic compound exposures in office workers: a controlled exposure study of green and conventional office environments. Environ Health Perspect 124(6):805–812
5. Anderson CG, Tsaasan AM, Reitman J, Lee JS, Wu M, Steel H, Turner T, Steinkuehler C (2018) Understanding esports as a stem career ready curriculum in the wild. In: 2018 10th International conference on virtual worlds and games for serious applications (VS-games), pp 1–6
6. Andò B, Marletta V, Baglio S, Crispino R, Mostile G, Dibilio V, Nicoletti A, Zappia M (2020) A measurement system to monitor postural behavior: Strategy assessment and classification rating. IEEE Trans Instrum Meas 69(10):8020–8031
7. Baig MZ, Kavakli M (2019) A survey on psycho-physiological analysis & measurement methods in multimodal systems. Multimodal Technol Interact 3(2):37
8. Blom PM, Bakkes S, Spronck P (2019) Towards multi-modal stress response modelling in competitive league of legends. In: 2019 IEEE conference on games (cog). IEEE, pp 1–4
9. Buckley D, Chen K, Knowles J (2013) Predicting skill from gameplay input to a first-person shooter. In: 2013 IEEE conference on computational intelligence in games (CIG). IEEE, pp 1–8
10. Cornforth DJ, Adam MTP (2015) Cluster evaluation, description, and interpretation for serious games. In: Serious games analytics. Springer, pp 135–155
11. Dey R, Salemt FM (2017) Gate-variants of gated recurrent unit (gru) neural networks. In: 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). IEEE, pp 1597–1600
12. Diaz-Romero DJ, Rincón AMR, Miguel-Cruz A, Yee N, Stroulia E (2021) Recognizing emotional states with wearables while playing a serious game. IEEE Trans Instrum Meas 70:1–12
13. Drachen A, Nacke LE, Yannakakis G, Pedersen AL (2010) Correlation between heart rate, electrodermal activity and player experience in first-person shooter games. In: Proceedings of the 5th ACM SIGGRAPH symposium on video games. ACM, pp 49–54
14. Drachen A, Sifa R, Bauckhage C, Thurau C (2012) Guns, swords and data: clustering of player behavior in computer games in the wild. In: 2012 IEEE conference on computational intelligence and games (CIG). IEEE, pp 163–170
15. Dymarski P (2011) Hidden Markov models: theory and applications. BoD–Books on Demand
16. Eggert C, Herrlich M, Smeddinck J, Malaka R (2015) Classification of player roles in the team-based multi-player game dota 2. In: International conference on entertainment computing. Springer, pp 112–125
17. Ershad M, Rege R, Fey AM (2018) Automatic surgical skill rating using stylistic behavior components. In: 2018 40th Annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE, pp 1829–1832
18. Fan J, Upadhye S, Worster A (2006) Understanding receiver operating characteristic (roc) curves. Can J Emerg Med 8(1):19–20
19. Fowles DC (1980) The three arousal model: implications of gray's two-factor learning theory for heart rate, electrodermal activity, and psychopathy. Psychophysiology 17(2):87–104
20. Gao L, Judd J, Wong D, Lowder J (2013) Classifying dota 2 hero characters based on play style and performance. Univ. of Utah Course on ML
21. Garmin Ltd (2012) Heart rate monitor. Technical report. http://static.garmin.com/pumac/HRM_Soft_Strap_ML_Web.pdf. Accessed 3 Aug 2022
22. Groh BH, Fleckenstein M, Kautz T, Eskofier BM (2017) Classification and visualization of skateboard tricks using wearable sensors. Pervasive Mob Comput 40:42–55. ISSN 1574-1192
23. Haladjian J, Schlabbers D, Taheri S, Tharr M, Bruegge B (2020) Sensor-based detection and classification of soccer goalkeeper training exercises. ACM Trans Internet Things 1(2):1–20
24. Hara K, Saito D, Shouno H (2015) Analysis of function of rectified linear unit used in deep learning. In: 2015 International joint conference on neural networks (IJCNN). IEEE, pp 1–8

25. Healey JA, Picard RW (2005) Detecting stress during real-world driving tasks using physiological sensors. IEee Trans Intell Transp Syst 6(2):156–166
26. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
27. Hodge V, Devlin S, Sephton N, Block F, Drachen A, Cowling P (2017) Win prediction in esports: mixed-rank match prediction in multi-player online battle arena games. arXiv:1711.06498
28. Inc Digilent (2016) Pmodtmp3 reference manual. Technical report. https://reference.digilentinc.com/_media/reference/pmod/pmodtmp3/pmodtmp3_rm.pdf
29. Inc Digilent (2017) Pmod hygro reference manual. Technical report. https://reference.digilentinc.com/_media/reference/pmod/pmodhygro/pmod-hygro-rm.pdf
30. InvenSense Inc (2016) Mpu-9250 product specification. Technical report. https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf
31. Khan A, Mellor S, Berlin E, Thompson R, McNaney R, Olivier P, Plötz T (2015) Beyond activity recognition: skill assessment from accelerometer data. In: Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing. ACM, pp 1155–1166
32. Khromov N, Korotin A, Lange A, Stepanov A, Burnaev E, Somov A (2019) Esports athletes and players: a comparative study. IEEE Pervasive Comput 18(3):31–39
33. Kim J-H (2009) Estimating classification error rate: repeated cross-validation, repeated hold-out and bootstrap. Comput Stat Data Anal 53(11):3735–3745
34. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
35. Kleinbaum DG, Dietz K, Gail M, Klein M, Klein M (2002) Logistic regression. Springer
36. Kodama K, Ioi K, Ohtsubo Y (2015) Development of a new skill acquisition tool and evaluation of mold-polishing skills. In: Proceedings of the 34th Chinese control conference and sice annual conference, pp 139–142
37. Kranz M, MöLler A, Hammerla N, Diewald S, PlöTz T, Olivier P, Roalter L (2013) The mobile fitness coach: towards individualized skill assessment using personalized mobile devices. Pervasive Mob Comput 9(2):203–215
38. Ladha C, Hammerla NY, Olivier P, Plötz T (2013) Climbax: skill assessment for climbing enthusiasts. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, pp 235–244
39. Lan L, Wargocki P, Wyon DP, Lian Z (2011) Effects of thermal discomfort in an office on perceived air quality, sbs symptoms, physiological responses, and human performance. Indoor Air 21(5):376–390
40. Lehrer PM, Batey DM, Woolfolk RL, Remde A, Garlick T (1988) The effect of repeated tense-release sequences on emg and self-report of muscle tension: an evaluation of Jacobsonian and post-Jacobsonian assumptions about progressive relaxation. Psychophysiology 25(5):562–569
41. Lin L League of legends match outcome prediction
42. Liu L, Jiang H, He P, Chen W, Liu X, Gao J, Han J (2019) On the variance of the adaptive learning rate and beyond. arXiv:1908.03265
43. Ltd Zhengzhou Winsen Electronics Technology Co (2016) Intelligent infrared co2 module. Technical report. https://www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf. Accessed 3 Aug 2022
44. Makarov I, Savostyanov D, Litvyakov B, Ignatov DI (2017) Predicting winning team and probabilistic ratings in “dota 2” and “counter-strike: Global offensive” video games. In: International conference on analysis of images, social networks and texts. Springer, pp 183–196
45. Märtens M, Shen S, Iosup A, Kuipers F (2015) Toxicity detection in multiplayer online games. In: 2015 International workshop on network and systems support for games (netgames). IEEE, pp 1–6
46. Martin-Niedecken AL, Schattin A (2020) Let the body ’n’ brain games begin Toward innovative training approaches in esports athletes. Front Psychol 11:article 138
47. Matsumura S, Ohta K, Yamamoto S-I, Koike Y, Kimura T (2021) Comfortable and convenient turning skill assessment for alpine skiers using imu and plantar pressure distribution sensors. Sensors 21(3):834
48. Nagorsky E, Wiemeyer J (2020) The structure of performance and training in esports, vol 15
49. Newzoo (2021) Global esports market report. https://newzoo.com/insights/trend-reports/newzoos-global-esports-live-streaming-market-report-2021-free-version/. Accessed 3 Aug 2022
50. Qin Y, Song D, Chen H, Cheng W, Jiang G, Cottrell G (2017) A dual-stage attention-based recurrent neural network for time series prediction. arXiv:1704.02971
51. Roose KM, Veinott ES (2020) Leveling up: using the tracer method to address training needs for esports players. In: Proceedings of the human factors and ergonomics society annual meeting, vol 64, pp 1780–1784
52. Saponara S (2017) Wearable biometric performance measurement system for combat sports. IEEE Trans Instrum Meas 66(10):2545–2555

53. Seeed Technology Inc (2015a) Grove–emg detector. Technical report. https://www.mouser.com/datasheet/2/744/Seeed_101020058-1217528.pdf. Accessed 3 Aug 2022
54. Seeed Technology Inc (2015b) Grove—gsr. Technical report. https://www.mouser.com/catalog/specsheets/Seeed_101020052.pdf. Accessed 3 Aug 2022
55. Shim KJ, Hsu K-W, Damania S, DeLong C, Srivastava J (2011) An exploratory study of player and team performance in multiplayer first-person-shooter games. In: 2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing. IEEE, pp 617–620
56. Smerdov A, Kiskun A, Shaniiazov R, Somov A, Burnaev E (2019a) Understanding cyber athletes behaviour through a smart chair: Cs:go and monolith team scenario. In: 2019 IEEE 5th world forum on internet of things (WF-iot), pp 973–978
57. Smerdov A, Burnaev E, Somov A (2019b) esports pro-players behavior during the game events: Statistical analysis of data obtained using the smart chair. In: 2019 IEEE Smartworld, ubiquitous intelligence computing, advanced trusted computing, scalable computing communications, cloud big data computing, internet of people and smart city innovation (smartworld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pp 1768–1775
58. Smerdov A, Somov A, Burnaev E, Zhou B, Lukowicz P (2021) Detecting video game player burnout with the use of sensor data and machine learning. IEEE Internet Things J 8(22):16680–16691
59. Smithies TD, Campbell MJ, Ramsbottom N, Toth AJ (2021) A random forest approach to identify metrics that best predict match outcome and player ranking in the esport rocket league
60. Stone T, Stone N, Roy N, Melton W, Jackson JB, Nelakuditi S (2019) On smart soccer ball as a head impact sensor. IEEE Trans Instrum Meas 68(8):2979–2987
61. Taelman J, Vandeput S, Spaepen A, Van Huffel S (2009) Influence of mental stress on heart rate and heart rate variability. In: 4th European conference of the international federation for medical and biological engineering. Springer, pp 1366–1369
62. Tang H, Glass J (2018) On training recurrent networks with truncated backpropagation through time in speech recognition. In: 2018 IEEE Spoken language technology workshop (SLT). IEEE, pp 48–55
63. Tsoi AC, Back A (1997) Discrete time recurrent neural network architectures: a unifying review. Neurocomputing 15(3–4):183–223
64. Tuncer T, Ertam F, Dogan S, Subasi A (2020) An automated daily sports activities and gender recognition method based on novel multikernel local diamond pattern using sensor signals. IEEE Trans Instrum Meas 69(12):9441–9448
65. Velichkovsky BB, Khromov N, Korotin A, Burnaev E, Somov A (2019) Visual fixations duration as an indicator of skill level in esports. In: Lamas D, Loizides F, Nacke L, Petrie H, Winckler M, Zaphiris P (eds) Human-computer interaction—INTERACT 2019. Springer International Publishing, Cham, pp 397–405. ISBN 978-3-030-29381-9
66. Viggiato M, Bezemer C-P (2020) Trouncing in dota 2: an investigation of blowout matches. In: Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment, vol 16, pp 294–300
67. Wang Z, Li J, Wang J, Zhao H, Qiu S, Yang N, Shi X (2018a) Inertial sensor-based analysis of equestrian sports between beginner and professional riders under different horse gaits. IEEE Trans Instrum Meas 67(11):2692–2704
68. Wang Y, Zhao Y, Chan RHM, Li WJ (2018b) Volleyball skill assessment using a single wearable micro inertial measurement unit at wrist. IEEE Access 6:13758–13765
69. Wang J, Wang Z, Gao F, Zhao H, Qiu S, Li J (2020) Swimming stroke phase segmentation based on wearable motion capture technique. IEEE Trans Instrum Meas 69(10):8526–8538
70. Xenopoulos P, Doraiswamy H, Silva CT (2020) Valuing player actions in counter-strike: Global offensive. In: 2020 IEEE international conference on big data (big data), pp 1283–1292
71. Xochicale M, Baber C, Oussalah M (2017) Analysis of the movement variability in dance activities using wearable sensors. In: Wearable robotics: challenges and trends. Springer, pp 149–154
72. Yamamoto T, Fujinami T (2008) Hierarchical organization of the coordinative structure of the skill of clay kneading. Hum Move Sci 27(5):812–822
73. Yamanaka H, Makiyama K, Osaka K, Nagasaka M, Ogata M, Yamada T, Kubota Y (2015) Measurement of the physical properties during laparoscopic surgery performed on pigs by using forceps with pressure sensors. Adv Urol 2015
74. Yang Y, Qin T, Lei Y-H (2016) Real-time esports match result prediction. arXiv:1701.03162
75. Yi S, Ju J, Yoon M-K, Choi J (2017) Grouped convolutional neural networks for multivariate time series. arXiv:1703.09938

76. Yuji O (2005) Mems sensor application for the motion analysis in sports science. Memory 32:128Mbit
77. Zhu M, Liu W, Wargocki P (2020) Changes in eeg signals during the cognitive activity at varying air temperature and relative humidity. J Expo Sci Environ Epidemiol 30(2):285–298

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.