IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 17 July 2023, accepted 31 July 2023, date of publication 3 August 2023, date of current version 10 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3301775

RESEARCH ARTICLE

GameDepot: A Visual Analytics System for Mobile Game Performance Testing

DONGHOON JANG[1] AND JAEMIN JO[2]

[1]Department of Digital Media Communication Engineering, Sungkyunkwan University, Suwon 16419, South Korea
[2]Department of Artificial Intelligence and Department of Computer Science and Engineering, Sungkyunkwan University, Suwon 16419, South Korea

Corresponding author: Jaemin Jo (jmjo@skku.edu)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT), Artificial Intelligence Graduate School Program (Sungkyunkwan University) under Grant 2019-0-00421.

ABSTRACT In this paper, we present GameDepot, a visual analytics system designed to enable interactive analysis of performance testing logs for mobile games. Due to the emergence of computational-heavy mobile games, the performance testing of mobile games has become a crucial practice to strike a balance between user experience, battery life, and thermal management. However, policymakers in device manufacturers face challenges in understanding the complex performance testing logs generated for different game and device combinations. To address this issue, we conduct a design study with ten domain experts and develop GameDepot, a visual analytics system with six tightly coordinated views that provide an overview of thousands of testing sessions and facilitate the identification of abnormal events, such as excessive throttling. We implement a data modeling pipeline based on Long Short Term Memory (LSTM) to support advanced data manipulation operations that are needed in practice, such as measuring the similarity between devices and or predicting the performance of an unseen combination of games and devices. Our evaluation demonstrates that GameDepot not only supports the seven essential tasks we identified during the design study but also facilitates the identification of abnormal sessions.

INDEX TERMS Information visualization, visual analytics, mobile games, log visualization, machine learning.

I. INTRODUCTION

Optimizing the performance of smartphones becomes especially crucial to run high-performance 3D games, which require substantial resource computations. These games consume a significant portion of smartphone resources such as CPU, GPU, and memory, leading to a considerable current generation and rapid temperature increase in the device. When managed poorly, the temperature of a smartphone brings substantial negative impacts on user experience, such as frame drops, sometimes even bringing severe damage to the user, such as low-temperature burns.

To strike a delicate balance between performance and stability without compromising the overall gaming experience, smartphone manufacturers run a special team, a Game Performance Analysis Team (GPAT). Their goal is to make decisions regarding the adjustment of resource usage policy informed by the testing logs collected by human testers. In this work, we collaborated with the GPAT of one of the most prominent smartphone manufacturers. From the collaboration, we found out that the GPAT currently resorts to an ad-hoc and ineffective combination of existing tools to analyze the log data, such as GameBench [1], challenged by the vast amount of data collected. Specifically, we identified the following three challenges in the current tasks:

- **Pre-test Planning:** Human testing is an expensive operation. Given the limited number of tests allowed every day, the team should carefully plan tests, seeking a balanced schedule over various games and devices. To this end, the comparison and selection of games and devices should incorporate multiple perspectives, such as similarities between them and diverse ranking criteria. Unfortunately, existing tools only provide a simple overview of tested combinations with a few statistics, rendering them less effective in pre-test planning.
- **Post-test Analysis:** The existing tools do not support the holistic view of the test history, encompassing thousands of session logs. Although a few tools (e.g.,

The associate editor coordinating the review of this manuscript and approving it for publication was Ikramullah Lali.

"GameBench") display multiple time series of performance indicators overlapped on a single screen, merely overlapping time series does not allow for post-test analysis, such as scalable exploration of sessions.
- **Post-test Tuning:** Given the wide range of games and devices, it is essential to conduct specialized analyses tailored to each game and device in order to establish effective performance policies. As it stands, present tools only support reviewing a single test session and lack the capabilities to rank performance metrics or investigate similarities among games and devices. Owing to the sheer volume and intricate nature of session data, there is a pressing demand for dedicated tools that can address these challenges and facilitate performance analysis across individual games and devices.

As our answer to these challenges, we propose a visual analytics system called "GameDepot" that enables the interactive analysis of large volumes of testing session data from various perspectives. GameDepot not only provides an aggregated view of multiple sessions but also offers the user the ability to compare sessions from a new perspective through rich visualizations, explore similar sessions or outliers using machine learning algorithms, and predict the performance of new game and device combinations.

In this paper, we present our research in a structured manner: In Section II, we introduce research related to the development of GameDepot. In Section III, we define the domain problem that our study addresses. In Section IV, we present the GameDepot visualization system as a solution to the defined problem. Lastly, in Section V, we validate the process of examining large-scale sessions using this tool with experts.

In summary, our contributions are as follows:

- From the collaboration with ten domain experts, we identify and abstract seven important analytic tasks in mobile game performance testing.
- We design and improve a visual analytics system GameDepot for analyzing mobile game performance logging data.
- Finally, we evaluate our design by conducting a case study with two domain experts. We found out that the users could successfully perform the seven essential tasks and make informed decisions regarding a resource usage policy.

II. RELATED WORK

A. SMARTPHONE LOGGING COLLECTION SYSTEM

In recent years, there has been an extensive study of systems for collecting log data on user behavior and app performance on smartphones. In particular, research in this field consistently harnesses location information, tracks user behavior patterns such as app-switching, or monitors device state information (for example, battery level or screen resolution) as a topic for different studies in areas such as education, healthcare, disease management, and transportation. First, location-based log data can be used to investigate user behaviors [2], [3]. Such location-based information can also be applied to drivers, enabling research on how dangerous using a smartphone while driving can be [4]. Second, smartphone interaction is tracked and utilized. By analyzing smartphone logs, individual smartphone usage patterns and task-switching frequency can be measured and used in research [5], [6]. In particular, this information is widely used in healthcare [7], [8], [9], [10]. Not only for health, but research is also diversely utilized for disease prevention such as smartphone addiction [11], [12]. Yan et al. [13] investigate the validation and performance assessment of an Android smartphone-based GNSS/INS coupled navigation system, demonstrating the potential of using smartphone log data for navigation purposes. In another study, Singh et al. [14] provide a state-of-the-art review of smartphone battery state-of-charge (SoC) estimation and battery lifetime prediction techniques, highlighting the importance of log data in battery management. Sahoo et al. [15] examine the validity and efficiency of a smartphone-based electronic data collection tool for operative data in rotator cuff repair, showcasing the potential of log data in medical applications.

Furthermore, extensive research is being conducted on the use of log data for the overall stabilization of smartphone systems, encompassing areas such as network management, security, among others [16], [17], [18], [19]. Beyond these examples, smartphone log data has extensive potential for utilization across a wide range of fields. However, most of these smartphone logging systems do not provide visualization. In this paper, we present a method for visually analyzing CPU, GPU, and Memory usage, along with performance metrics such as Frames Per Second (FPS) and power consumption data, which can be collected during smartphone game execution. By providing users with effective visualizations, we aim to assist them in their analysis.

B. LOG DATA ANALYSIS SYSTEM

Recently, anomaly detection techniques using log data have been developed to distinguish normal and abnormal system behavior. In particular, the research by Zhang et al. proposes an algorithm that robustly detects anomalies even in unstable log data [20]. Applying such technology to log data analysis systems can become a crucial tool for quickly identifying and addressing system issues. CloudDet proposes an interactive visual analysis system for detecting anomalous performances in cloud computing systems [21]. The system utilizes techniques such as anomaly detection, visualization, and static log analysis to provide insights into system performance issues. This approach can be useful for improving system reliability and efficiency in cloud computing environments. Similarly, the LabelVizier also supports visualization for detecting anomalies in labels, making it relevant to our study [22]. The MiningVis paper proposes a visual analytics approach to study the Bitcoin mining economy, considering not only the economic value of mining activities but also aspects such as power consumption [23]. The

Anchorage presents a research study focused on transforming time-series data embedded in logs into anchor events for the visual analysis of satisfaction in customer service. This novel methodology leverages anchor events to measure and improve satisfaction levels in the customer service domain, offering a new approach to better understand and address customer needs and preferences [24]. And the research by Polk et al. presents a visual analytics approach for generating valuable insights from data in tennis matches. This research suggests that utilizing visual analysis tools can significantly contribute to improving player performance, strategic planning, and predicting match outcomes, ultimately aiding in the decision-making process within the sport [25]. However, while these papers play important roles in their respective domains, they do not provide support for analyzing game log data specifically.

Indeed, there are systems for analyzing game log data. For example, Shoukry and Göbel [26], [27] present a mobile multimodal platform for serious games analytics. This platform combines user performance data with physiological and environmental sensor data to provide insights into the cognitive and emotional states of game players. With potential applications in various fields such as healthcare and education, the platform aims to develop games that promote positive outcomes. There are other studies that have applied insights gained from game log data to improve the gaming experience. In a related study, Smerdov et al. [28] proposed an AI-enabled approach for predicting video game player performance using data from heterogeneous sensors, which could help developers and researchers better understand and improve the gaming experience for players. In the study by Lee et al. [29], a game data mining competition was held focusing on churn prediction and survival analysis using commercial game log data. This research highlights the importance and potential of utilizing game log data to predict player behavior and improve player retention in the gaming industry. In the work by Harpstead et al. [30], the authors explore the creation of engagement profiles of players using game log data. Their research demonstrates the potential of game log data to better understand player motivations and preferences, which can ultimately lead to a more engaging and personalized gaming experience. However, these studies primarily focus on providing information for game developers and operators to enhance game performance, optimization, and user experience. In contrast, our paper presents, for the first time, a method for analyzing log data collected from devices during game execution, with a focus on manufacturers' perspectives.

III. DOMAIN ANALYSIS

In this section, we introduce the background of smartphone thermal management, the main problem our target users have at hand and want to solve by analyzing testing logs, followed by abstraction results of their data and tasks.

A. SMARTPHONE THERMAL MANAGEMENT

Smartphone thermal management, i.e., maintaining the heat emitted from smartphones below a certain level, is critical for ensuring a safe and reliable user experience. There are a few reasons that such a problem is more challenging for smartphones compared with other devices such as desktops. First, due to their compact form factors, smartphones typically lack dedicated cooling fans and instead rely on heat-dissipation pipes that offer limited cooling compared with desktop fans. Second, because smartphones are handheld devices, their components must operate at lower temperatures than desktops. For example, a temperature of $90°C$ may be acceptable for a desktop GPU, but it is entirely unacceptable for any component in a smartphone. Finally, smartphones are increasingly adopting thin, metal materials with high thermal conductivity together with more powerful Application Processors (APs), each of which makes thermal management even more challenging.

The techniques for smartphone thermal management can be broadly classified as either hardware-based or software-based. Hardware-based techniques include optimizing the design of smartphone circuits and heat-dissipation pipes. However, these methods offer less flexibility as the hardware components are chosen at the time of manufacturing and cannot be changed according to smartphone usages.

On the other hand, software-based techniques provide greater flexibility and enable application-specific optimization. One simple yet widely-used strategy of software-based thermal management is called **throttling**. When activated, throttling deliberately decreases AP's (CPU or GPU) performance, which in turn decreases the power consumption and eventually heat emission. Additionally, such software-based throttling technologies are gaining more attention due to their ability to change policies in real-time based on feedback from in-game monitoring. For example, when the device is performing other important tasks such as network communication or app installation, throttling parameters can be applied more strongly. In contrast, the parameters can be applied more loosely during game initiation for a quicker launch, or even throughout the playtime for high-performance games.

It is crucial to apply throttling at the right moment and with the appropriate degree, considering its potential negative impact on user experience. For instance, when throttling is triggered due to high heat levels exceeding a certain threshold, it may result in sudden frame drops or delays, significantly degrading the user experience, especially in responsiveness-critical applications like mobile games. Hence, a context-aware approach to throttling is essential, striking a balance between user experience and thermal management; for example, in the example mentioned above, the degree of throttling should be limited to minimize the impact on user experience.

In modern smartphones, throttling behavior is defined by approximately 30 parameters that are specified by the operating system and can be customized and updated by the manufacturer. These parameters encompass various aspects, including the degree of performance reduction (i.e., the level to which AP's performance is throttled), the context in which throttling is triggered (i.e., which applications can trigger throttling), and the timing of throttling (i.e., how long throttling persists). For instance, a particular configuration may temporarily pause medium-level throttling for 10 seconds after a mobile game is launched, aiming to minimize delays at the loading screen and enhance user experience. Such customization of throttling parameters enables manufacturers to fine-tune throttling behavior in specific usage scenarios.

Firstly, previous studies have often focused on thermal management by measuring device temperature. However, device temperature is significantly influenced by external factors, such as ambient air temperature and the user's hand temperature. Therefore, analyzing power consumption provides a more comprehensive understanding of the device's overall performance and energy efficiency, as reported in previous research [31], [32]. This approach enables better resource optimization for the device, primarily focusing on enhancing the device's battery life and identifying potential areas where improvements can be made.

B. TARGET USERS

The smartphone manufacturer we have collaborated with had a special team, Game Performance Analysis Team (GPAT), that is responsible for game performance and gameplay experience. The team was especially interested in optimizing for mobile games, one of the most popular and computation-intensive application types. Their goal is to ensure device stability and user experience by carefully adjusting the throttling parameters according to the game the user is running.

We identified two important roles in the team: **testers** and **analysts**. The task of **testers** is to reproduce user-reported performance issues on their devices. The tests could be run either automatically (e.g., scripting) or manually, but we found most tests were done manually as most issues were about a certain edge case. The devices used for manual testing were installed with special logging software, so performance logs, such as FPS, CPU usage, power consumption, etc. are collected during testing. When an issue is confirmed significant and reproducible by testers, the performance logs collected from the testing session are sent to analysts for further analysis.

In our design study, we set **Analysts** as the main user group. The goal of analysts in this context includes the following:

- **Setting up and monitoring game and device testing plans:** Analysts are responsible for determining the games and devices to be tested, providing a comprehensive list to the testing team to ensure adequate coverage.
- **Analyzing collected test results:** Analysts examine the session data obtained from tests to identify patterns, trends, and anomalies. This information is then used to gain insights into the performance of games and devices. For example, they may analyze resource usage time-series for game test data on specific devices where performance issues have occurred.
- **Determining and implementing performance optimization policies:** Based on their analysis, Analysts decide on policies to optimize device resource usage control parameters for games and devices, and implement and apply them accordingly. The parameters mentioned, such as CPU level and GPU level, have positive values that decrease the limitation of Application Processors (AP) like CPU and GPU, while negative values increase the AP limitation. Another parameter includes the duration value, which sets the duration for opening the AP limitation to its maximum during game launch for a few seconds.

C. DATA ABSTRACTION

We define a single run of a specific application on a specific device as a **session**. Then, the log data collected by testers so far can be seen as an array of sessions where each session is described by the test configuration (e.g., game title and device name) along with time-series of performance indicators. The session data are stored in a well-known Android performance benchmarking database, GameBench, which can be accessed remotely. Without any filtering applied, we could access over 150K sessions for the last five years in the entire database.

Table 1 shows the abstraction result of a session.

First, **Game** and **Device** are nominal attributes that represent the game and device tested, respectively. They contain low-level codes, such as Android package names (Game), e.g., ''com.epicgames.fortnite'', or smartphone model names (Device), e.g., ''SM-S901''. Although our users could work with those low-level codes, for better readability, we use game titles and marketing names in the description of our system, e.g., ''Fortnite'' and ''Galaxy S22''. We also collected the metadata of games and devices, such as their release date, the genre of a game (e.g., ''Action''), and the class of a device (e.g., ''Entry'', ''Mid-Range'', and ''Flagship''). Hundreds of games and devices have been tested, respectively, and we asked domain experts to choose the games and devices of their current interest. We identified the 27 most popular games and 32 devices, most of which have been on the market in the past five years, and collected sessions from them.

The **Date** and **Duration** attributes of a session represent when and how long testing was performed. The average duration of sessions was approximately 45 minutes.

In addition to the attributes describing the testing configuration, a session also has time-series of five performance indicators that were recorded regularly during the testing phase as below:

- **FPS**: Frames per second (FPS) is the number of frames drawn on the screen every second. The maximum FPS value of a game is predetermined by the device or game, such as 120Hz or 60Hz. However, the actual FPS output of the game may be measured at a lower value,

TABLE 1. Data Abstraction.

Attribute	Attribute type	Description
ID	Ordinary	Unique session identifier
Game	Nominal	Game application name
Package name	Nominal	Unique identifier for an application
Game genre	Nominal	Category defining gameplay and style
Device	Nominal	Smartphone name
Device class	Nominal	Device type classification based on specifications
Date	Date	Date of game testing
Duration	Quantitative (Integer)	Tested time (ms)
FPS	Series of Integers	Frame per seconds (frames per second)
CPU	Series of Integers	CPU usage time series data (%)
GPU	Numeric array	GPU usage time series data (%)
Memory	Numeric array	Memory usage time series data (MB)
Power	Numeric array	Power consumption time series data (mW)
Tag	Nominal	Tester's comment, e.g., Test environment, OS version

depending on the system's condition. Additionally, the average value in the sessions was approximately 56.5Hz.

- **CPU** and **GPU**: These two indicators represent the utilization of APs. CPU and GPU indicate the percentage of a CPU's and GPU's processing power being utilized, respectively. In our study, the average CPU usage of the sessions was approximately 14%, while the average GPU usage was significantly higher at around 88%.
- **Memory**: Memory represents the amount of memory (RAM) occupied by applications or processes in MB. For memory usage, the minimum value was 465MB and the maximum value was 1690MB. However, with an average value of 1660MB, the majority of sessions displayed a distribution close to the maximum value.
- **Power**: Watt used per second. Power consumption refers to the amount of electrical energy consumed by a smartphone when running a game, and it is a crucial factor in determining the device's energy efficiency and battery life during gameplay.

The **Tag** represents comments that testers intend to communicate to analysts. For example, to indicate that the new parameter, Performance mode, has been activated, they may create a tag such as ''Perf On''.

D. TASK ABSTRACTION

We collaborated with the Game Performance Analysis Team (GPAT) of a smartphone manufacturer for 18 months to develop GameDepot, a visual analytics system for the smartphone thermal management problem. We initially started with ten analysts to gather and understand their domain requirements and have closely worked with two of them to iteratively refine the system requirements and design.

As GPAT conducts nearly 100 test sessions daily, analyzing and monitoring the results is a core challenge. To clearly describe this task, we employed the visualization design framework by Brehmer and Munzner [33], [34]. Each task is described in the form of [Action → Target]. As a result, we abstract their requirements into seven tasks as follows:

- **T1: Identifying Similar Games and Devices**. GPAT sets up testing plans considering the coverage of tests. Specifically, they want to avoid testing similar combinations as it can be resource-wasting. One simple approach to measuring the similarity between combinations is to check if they have games or devices of the same categories, e.g., of the same genre (i.e., shooting) or of the same class (i.e., mid-range). However, a more accurate method is needed, which takes the actual performance trait into account. [Compare → Game/Device]
- **T2: Overviewing Test Results**. Tests are conducted, and game session data is collected to address various performance issues that occur during gameplay. To manage monitoring results systematically, the process of reviewing collected data must be supported. [Present, Summarize → Test results]
- **T3: Identifying Relationships between Features**. analysts analyze the relationships between features for game performance optimization. For instance, they may discover the correlation between GPU usage and FPS or investigate which feature has the most significant impact on power consumption. These relationships may manifest as clusters specific to different games or devices. Alternatively, the game genre or the device's grade can also be considered. By leveraging the results of such analyses, analysts can identify characteristics of games or devices and detect potential bottlenecks, such as limitations in device specifications. Ultimately, such insights lead to improvements in overall performance. [Summarize, Compare → Features]
- **T4: Identifying Similar Sessions**. Analysts can address issues by observing similar sessions. However, finding relevant historical data when problems arise in a session can be time-consuming or difficult to locate. Identifying and reviewing the outcomes of similar sessions can provide hints for addressing the current issue. [Search → Intra outlier]
- **T5: Predicting the Average Performance Indicators of a Combination**. GPAT is also interested in recognizing a new combination of a game and device that has never been tested but is considered promising. As manual testing is an expensive and time-consuming

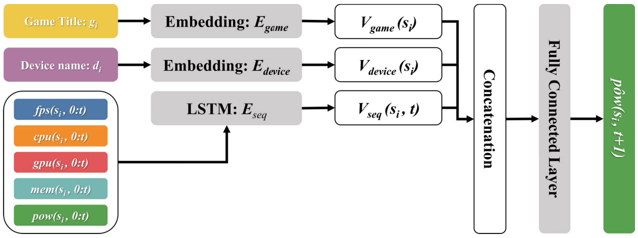

FIGURE 1. The LSTM-based architecture of GameDepot performs continuous predictions using autoregressive forecasting.

operation, it is unviable to test every combination of games and devices. Therefore, before testing, analysts carefully plan the target combinations to be tested and inform testers of the combinations. By predicting the average performance indicators of the combinations prior to testing, it is possible to anticipate potential savings in testing resources. [Discover → New combination of games and devices]

- **T6: Predicting Power Consumption**. Due to system constraints, power consumption values are not recorded in the raw data when the device is not charging; therefore, it is necessary to provide these values through prediction methods. Power consumption is influenced by the usage of system resources such as CPU, GPU, and Memory, and is also related to display output (FPS). [Discover → Missing power consumption]
- **T7: Exploring Anomalies of a Session**. To support analysts, it is essential to access detailed information and characteristics of sessions. Anomalies can be detected by examining the time series data contained within sessions. For example, a sudden spike in CPU usage and a sharp drop in FPS at a specific time point may indicate throttling occurring on the device. Moreover, such anomalies can also be identified through dimensionality reduction techniques, such as Principal Component Analysis (PCA). [Explore → Inter outlier]

E. DATA CLEANING

We discovered that the raw session data contained erroneous values in attributes such as missing values or wrong values that are no longer valid. In cases where a session has at least one missing or wrong value, we have excluded the entire session. We also excluded sessions that are shorter than 20 minutes, which could be due to interrupted tests, and longer than one hour, which might not be finished properly.

We also discarded sessions with missing FPS, CPU, GPU, and Memory values. There were sessions where power consumption is not recorded due to performance issues (T6). For these sessions, we predicted the power consumption using a neural network, which will be discussed in the next section. In the raw data, the five-time series are synchronized for training purposes. For this study, the time interval was standardized to 5 seconds in order to make the visualization appear smoother. By synchronizing the time series, it allows for the data to be transformed and modeled as sequences, enabling the use of algorithms and techniques designed for handling sequential data. The data normalization was applied before the training process. We preprocessed a total of 6,232 datasets and used 2,327 session data for visualization. For model training, we used 1,183 sessions containing power consumption time-series.

These sessions used in our research were the results of testing conducted by testers affiliated with that company. We did not utilize sessions from general users.

F. DATA PROCESSING

Our task abstraction result shows that GameDepot needs to support a range of analysis tasks for the given data. Specifically, we identified the following four data-driven tasks that the system should address. First, the system needs to predict the power consumption of a session over time, given the series of other performance indicators, e.g., CPU, and a game-device combination (T6). Second, the similarity between two games or two devices must be computed in a data-driven way to support T4. Third, the system also needs to measure the distance between the time series of two sessions, allowing the user to identify outlying sessions (T7). Finally, the system should be able to predict the average performance indicators given a new combination of a game and a device (T5).

Notation: Let S, G, D denote the set of tested sessions, games, and devices. For a session $s_i \in S$, let $game(s_i) \in G$ and $device(s_i) \in D$ denote the game and device tested in s_i. As a result of testing, we have the five sequences of performance indicators. Let $fps(s_i, t) \in \mathbb{R}$ ($t \in [1, len(s_i)]$) be the value of the FPS indicator at time t recorded in s_i. We will use similar notations for cpu, gpu, mem, and pow.

To support these operations on a single machine learning model, we propose an LSTM-based architecture as shown in Figure 1.

GPAT had internally developed a performance prediction solution using LSTM, and through interviews, it was decided to utilize LSTM for session prediction in GameDepot as well. LSTM is an artificial neural network architecture designed to solve the long-term dependency problem and is used for learning and predicting the temporal dependencies in input data.

Given that game sessions also exhibit time-series characteristics, the use of LSTM is particularly suitable and effective for our analysis. To predict $pow(s_i, t+1)$, the power consumption at $t+1$ in s_i, we first build an LSTM encoder, $\mathcal{E}_{seq}$ that encodes the five indicator sequences of s_i from the beginning to time t into a H_{seq}-dimensional vector generated by the hidden and cell state vectors of the last cell (Equation 1).

$$V_{seq}(s_i, t) = \mathcal{E}_{seq}([f_1; f_2; \ldots; f_{t-1}; f_t]) \in \mathbb{R}^{H_{seq}} \tag{1}$$

where f_t is a feature vector at time t given as below:

$$f_t = (fps(s_i, t), cpu(s_i, t), gpu(s_i, t), mem(s_i, t), pow(s_i, t)) \tag{2}$$

The power consumption at $t+1$ can be affected by the game and device. To reflect such an effect into the prediction process, we use two encoders to convert the categorical attributes to H_{game} and H_{dev}-dimensional vectors, respectively, given as below:

$$V_{game}(s_i) = \mathcal{E}_{game}(game(s_i)) \in \mathbb{R}^{H_{game}} \quad (3)$$

$$V_{dev}(s_i) = \mathcal{E}_{dev}(device(s_i))) \in \mathbb{R}^{H_{dev}} \quad (4)$$

Finally, the three representations are concatenated and fed into a fully connected layer to predict $pow(s_i, t+1)$:

$$\hat{pow}(s_i, t+1) = FC([V_{seq}(s_i, t); V_{game}(s_i); V_{dev}(s_i)]) \quad (5)$$

The three encoders, $\mathcal{E}_{seq}$, $\mathcal{E}_{game}$, and $\mathcal{E}_{dev}$ are also used to model the similarity between entities. For example, the distance between two games, g_i and g_j, is modeled as the L2 distance between their embeddings:

$$Dist(g_i, g_j) = \left\| \mathcal{E}_{game}(g_i) - \mathcal{E}_{game}(g_j) \right\| \quad (6)$$

Similarly, we also computed the distance between two devices, d_i and d_j:

$$Dist(d_i, d_j) = \left\| \mathcal{E}_{dev}(d_i) - \mathcal{E}_{dev}(d_j) \right\| \quad (7)$$

To measure the distance between the time series of two different sessions, we consider two their distance in the feature and embedding space. The distance between two sessions in the embedding space is given as below:

$$Dist_{emb}(s_i, s_j) = \left\| V_{seq}(s_i, len(s_i)) - V_{seq}(s_j, len(s_j)) \right\| \quad (8)$$

The distance between two sessions in the feature space was measured using Dynamic Time Warping (DTW).

The final distance between two sessions is computed by weighting the two distances.

$$Dist(s_i, s_j) = w \cdot Dist_{emb}(s_i, s_j) + (1-w) \cdot Dist_{DTW}(s_i, s_j) \quad (9)$$

To achieve a balanced representation of the results from two functions, we set $w = 0.5$ in our implementation (Equation 9). This value can be dynamically adjusted according to user preferences.

We used 1,183 out of 2,327 sessions to train the encoders, whose power consumption is recorded. For $\mathcal{E}_{seq}$, we used a single-layer LSTM model with 5 input dimensions and 16 hidden dimensions. The model employs an embedding dimension of 8 and a batch size of 64 for training. The input consists of five time-series data points, and considering the number of games and devices used in our research, we set the embedding length and hidden dimension accordingly. The models are trained using a learning rate of 0.05 for 1000 epochs, which yielded the highest accuracy in our experiments. To improve training efficiency, we employed early stopping when the validation loss ceased to decrease.

We verified the accuracy of session data prediction by obtaining the Mean Absolute Error (MAE) of 100.6 through 5-fold cross-validation, using 1,183 sessions.

We experiment with models having both single and multiple output dimensions, specifically 1 and 5.

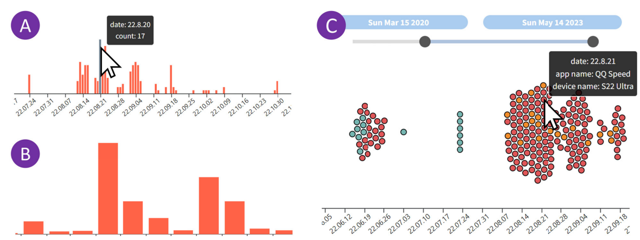

FIGURE 2. The design of the Timeline View is as follows: (a) The default is a daily line chart. (b) It can be toggled to a monthly view. (c) The bee swarm chart incorporates color encoding for genres.

IV. THE GameDepot DESIGN

Figure 3 shows an overview of the GameDepot interface. It consists of six views, which we will discuss in the order that the user is expected to use the views according to our task abstraction result.

A. TIMELINE VIEW

As the first step, the user can choose the sessions to be analyzed by specifying a desired time period on Timeline View (Figure 3A).

1) VISUAL ENCODING

In the Timeline View, the user can check the number of sessions tested over time and choose the timespan of interest. We designed the interface to allow the user to toggle between the line chart and bee swarm chart (Figure 2), enabling them to either focus on the number of sessions over time or see individual sessions as circles whose color encodes game genres, depending on their interests [35].

2) DESIGN ALTERNATIVES

Initially, a date picker was provided for choosing a range of time, but after design iterations, it was decided to provide a summary of session counts over time as visualization, enabling the user to understand the temporal trends.

3) INTERACTION

By default, the number of sessions per day is displayed as a line chart. The user can switch the daily line chart to a monthly line chart or change it to a bee swarm chart that depicts an individual session as a circle. Finally, the user can choose a time range using the slider, and the other five views are updated once a range is chosen.

B. GAME/DEVICE VIEWS

Next to the Timeline View, Game View and Device View provide an overview of games and devices tested during the chosen period. These views not only show the mean performance indicators of the sessions of each game or device but also allows the user to judge the similarity between games or devices based on their performance footprints.

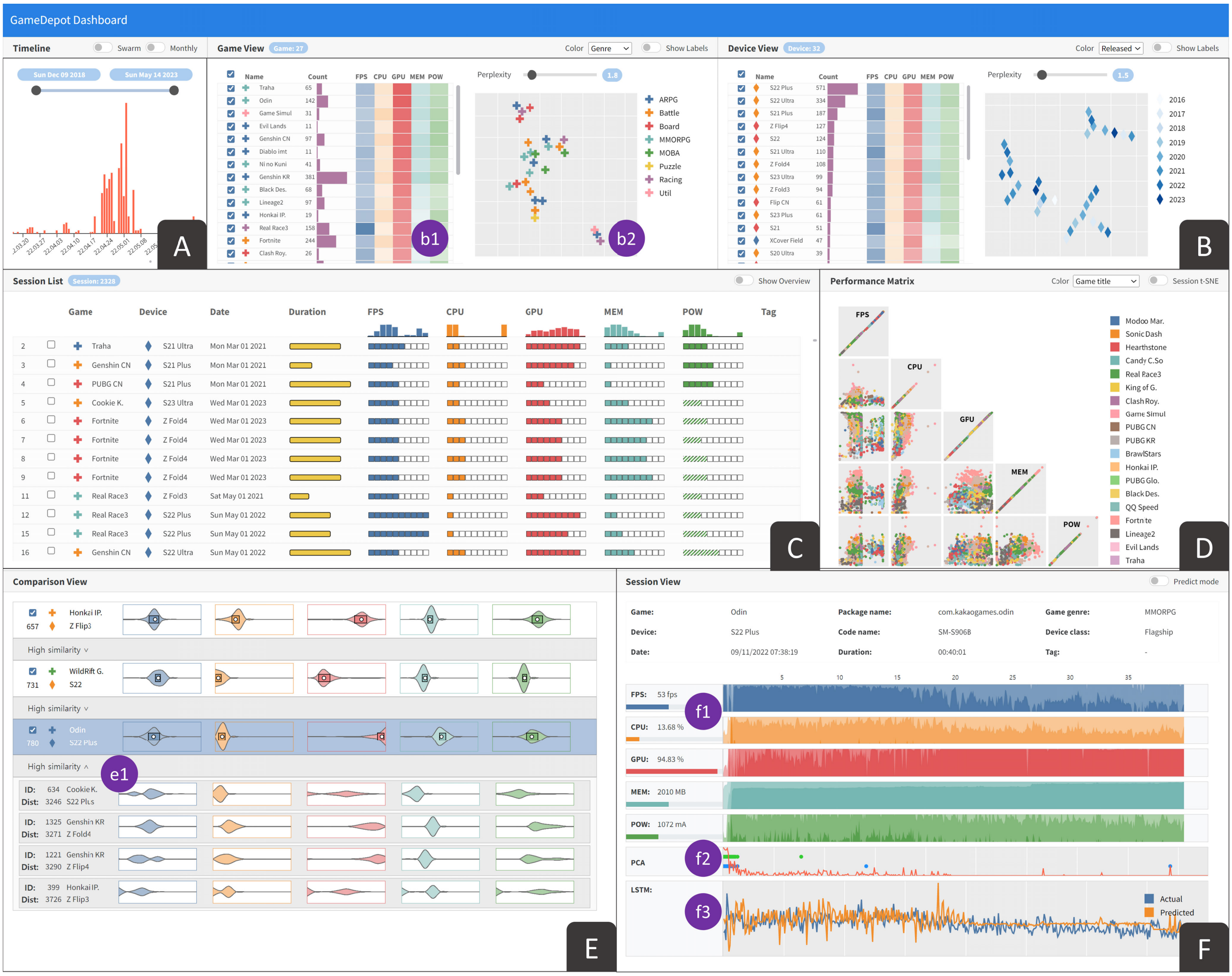

FIGURE 3. **The GameDepot interface has been designed to enable various perspectives to be explored by the user, ranging from an overview to detailed information, and from individual sessions to a comprehensive view of all sessions. The GameDepot contains six interactive modules: (A) Timeline View, (B) Game/Device View, (C) Session List, (D) Performance Matrix / Session Similarity View, (E) Comparison View and (F) Session View.**

1) VISUAL ENCODING

Each of the Game and Device Views consists of three horizontally juxtaposed visualization components: a histogram, a heatmap, and a scatterplot (Figure 3B). The histogram (Figure 3b1) shows the number of sessions where each game or device is tested, allowing the user to identify the games and devices of interest in the given period. Juxtaposed with the histogram, the heatmap encodes the mean values of the performance indicators, enabling the user to identify heavy games or slow devices (T1). These two components are specially designed to rank the games or devices tested according to a certain criterion, e.g., the game that used the most CPU.

The scatterplot (Figure 3b2) in each of the Game and Device Views provides a 2D depiction of games or devices. Specifically, we provide two t-distributed stochastic neighbor embedding (t-SNE) projections of their high-dimensional embeddings from the prediction model, $\mathcal{E}_{game}(g_i)$ and $\mathcal{E}_{dev}(d_i)$, respectively, allowing the user to evaluate the similarity between games and devices (T1). The color of a point in the t-SNE projection was used to overlay additional information on games and devices, such as genres, classes, or release dates.

2) DESIGN ALTERNATIVES

In the initial design iteration, we employed a 2D heatmap with games and devices used as the two axes and a cell representing a weighted mean of performance indicators for a game-device combination. While this approach could allow the user to identify the interaction effects between a game and a device, we found it often overwhelmed the user since every combination of games and devices are shown. As a result, we chose to provide Game and Device Views separately. To show the mean performance indicators for a game (or a device), we also considered parallel coordinates, but we found that they were not well-suited for our case due to visual clutters and decided to use a heatmap, which also naturally lends itself to attribute-based sorting.

3) INTERACTION

One important interaction that the user can perform on these two views is to select target games or devices by selecting the checkboxes in front of their names, narrowing down the scope of analysis. This allows for the display of only the selected games and devices in the Session List, enabling the user to focus on their desired information. Additionally, the user can sort the rows of the histogram and heatmap according to an attribute, such as game name, count, or mean indicator value. For the scatterplots, we allowed the user to change the perplexity hyperparameter of the t-SNE algorithm and the attribute to be mapped as the color of points.

C. SESSION LIST

Session List (Figure 3C) visualizes the sessions in the chosen period (Timeline View) with the chosen games and devices (Game and Device Views) as an interactive table. Inspired by the Line-Up interface [36], we provide two modes, an overview mode (Figure 4) for providing a comprehensive view of the sessions (T2) and a detail mode for supporting the inspection of an individual session.

1) VISUAL ENCODING

In the detail mode, each session is represented as a row in the interactive table. The game and device of a session are shown with symbols that are consistent with the Game and Device Views. We use six horizontal bars to represent the duration of the sessions as well as the mean value of the five performance indicators. We show ten scale markers on the bars for the indicators to support the user in reading their values without using a tooltip. For sessions whose power was not measured, we marked its bar with a stripe texture. Finally, we show the tag attribute in the rightmost cell that allows the user to see additional information about sessions.

In the overview mode, the row for each session is collapsed, eventually having a height of one pixel, as in the Line-Up interface. This condensed representation not only allows the user to see all sessions without scrolling but also is particularly useful when sorting the sessions to reveal trends and relationships of attributes.

2) DESIGN ALTERNATIVES

In the initial design, we employed a simple table without the two modes, which raised a scalability issue. We also considered using parallel coordinates, but it was challenging to effectively represent complex sessions, especially when there are numerous lines, as the visualization becomes limited in its ability to convey information.

3) INTERACTION

First, the user can choose sessions of interest by clicking a checkbox in their corresponding rows. The chosen sessions are shown in more detail in Comparison View, which will

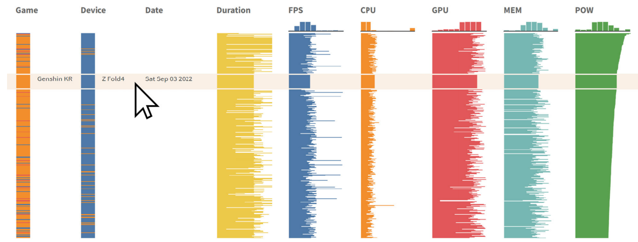

FIGURE 4. In the overview, each session is compressed and displayed as a single pixel, which does not show the game or device names, but allows for observing the overall distribution of the data. Furthermore, the sorting feature enables the user to examine the relationships between different features. However, the overview does not provide the functionality to select individual sessions.

be described next. Sessions can be sorted according to a particular attribute by clicking on an attribute name in the header. For the performance indicators, histograms are shown in the header, which allows the user to see the distribution and outliers for each indicator and create a filter on a range [37]. In the overview mode, the user can expand a condensed row to read off its values by hovering the cursor over it.

D. PERFORMANCE MATRIX/SESSION SIMILARITY VIEW

In Performance Matrix (Figure 3D), a scatterplot matrix (SPLOM) is used to display the relationship between the mean performance indicators (T3). Additionally, the user can switch the view to Session Similarity View (Figure 5B) two obtain an overview of the embeddings of sessions, i.e., $V_{seq}(s_i, len(s_i))$

1) VISUAL ENCODING

We used a scatterplot matrix, consisting of fifteen scatterplots, to show the relationship between each pair of performance indicators; in each scatterplot, a session is shown as a point whose x and y positions encode the means of two indicators, respectively. Switched to Session Similarity View, we use a scatterplot to visualize a t-SNE projection of sessions, using $Dist(s_i, s_j)$ as the distance measure between two sessions, s_i and s_j(Figure 5B).

2) DESIGN ALTERNATIVES

We chose to use a scatterplot matrix since it is one of the visualization methods used to show the relationship between multiple attributes [38] and it illustrates the correlation between the attributes (T3). One drawback of the scatterplot matrix could be that the complexity of the visualization increases as the number of attributes grows; however, in our case, such a limitation could be mitigated since the number of attributes was limited to five. In the design process, we considered juxtaposing the Performance Matrix and the Session Similarity View, but we allowed the user to see the views one by one since juxtaposing the two views spent a considerable amount of space, and the views were not needed to be visible at the same time according to user tasks.

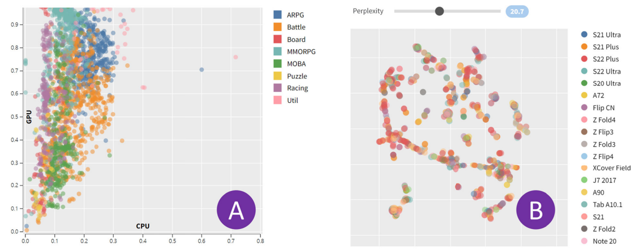

FIGURE 5. **The user can use the Performance Matrix to (a) examine the relationships between features and (b) check the *t*-SNE similarity.**

3) INTERACTION

Similar to the *t*-SNE projections in the Game and Device Views, the user can encode an attribute of sessions, such as game title, genre, and device class, to the color of points. Each scatterplot in the SPLOM is enlarged when the user can click on each (Figure 5A), showing the relationship in detail. When the view switched to Session Similarity View, the perplexity of *t*-SNE can be adjusted through a slider control on the fly.

E. COMPARISON VIEW

We provide Comparison View (Figure 3E) to facilitate the comparison between the sessions that the user selected in the Session List (T4). In contrast to the Session List, where the mean of each indicator of a session is displayed, the Comparison View shows the distribution of indicators recorded during a session. The user can select a session in the Comparison View by marking a checkbox to identify the sessions that have similar trends on the indicators to it; we compute the distance between sessions according to Equation X (Figure 3e1).

1) VISUAL ENCODING

We chose to visualize the distribution of indicators as a combination of a violin plot [39], [40] and a box plot. The violin plot shows the overall distribution of indicators while the box plot visualizes certain statistics, e.g., the median and 25% and 75% quartiles.

2) DESIGN ALTERNATIVES

Initially, we only used a box plot, but we found out that the distribution is often skewed, which is hard to be shown as a box plot. Therefore, we determined to overlay a violin plot to complement the box plot.

3) INTERACTION

The user can click on a session to see the actual time series in the following Session View and search for similar sessions. Or, the user can discard a session from the Comparison view by unchecking the corresponding checkbox.

F. SESSION VIEW

Session View (Figure 3F) provides the most detail of a session selected in the Comparison View, allowing the user to access the raw data (T7), allowing the user to inspect the time series of the five indicators, identify possible anomalies in the series, and see the predicted time series. The predicted time series provides two functionalities: it predicts missing values (T6) and anticipates the outcomes of new combinations (T5).

FIGURE 6. **The Session View offers a predict mode. The user can explore the performance of all game and device combinations through predictions.**

1) VISUAL ENCODING

We chose to use horizon charts (Figure 3f1) to display the time series of the five features because they effectively compress the visualization space of the five features and enable the user to easily observe trends at a glance [41]. To maintain consistency, the color encoding of each feature is kept the same as in previous views.

Two charts were added below the Horizon charts in Session View. One is a PCA result chart for supporting anomaly detection (Figure 3f2), and the other is an LSTM prediction chart for generating missing values (Figure 3f3).

2) DESIGN ALTERNATIVES

Initially, we considered using general line charts or area charts to visualize the multivariate time-series. However, these charts consume a large amount of space, and adjusting the chart height results in lines appearing flat, making it difficult to discern data trends. We also contemplated displaying all five values within a single chart, but this would lead to a loss of visual clarity. Ultimately, we chose horizon charts, as they efficiently utilize space while effectively displaying peaks and flows.

3) INTERACTION

With the mouse hover event on Horizon charts, it amalgamates and presents all five feature values for swift comparison. The user can toggle the predict mode on/off using the switch at the top of the Session View. In predict mode (Figure 6), the user can select a game and device to check the predicted values.

V. CASE STUDY

To validate the utility and usability of GameDepot, we conducted a case study with two analysts from the GPAT.

FIGURE 7. To compare the two games, the rankings of GPU, memory, and other features are examined by sorting them in the Game view (a) and (b). A more detailed ranking can be obtained by analyzing the feature relationships of each session in (c) and (d). Similarly, a rough comparison can be made through the overview in (e). In the images, the game represented in blue demonstrates higher overall GPU usage compared to the game depicted in orange.

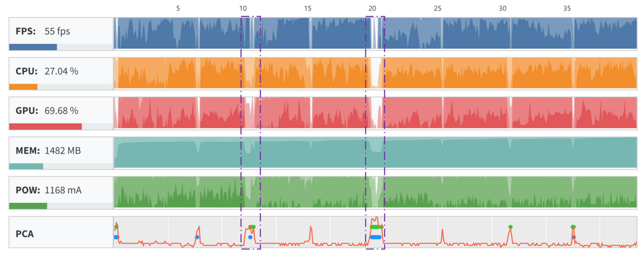

FIGURE 8. This is the time series for "Genshin Impact" session 337. Throttling events can be observed.

A. PARTICIPANTS AND PROCEDURE

We recruited three analysts from the GPAT who did not participate in the design study of GameDepot (**E1-E3**). The first participant (**E1**) was a former developer with 7 years of experience, and the second participant (**E2**) had also worked as a developer who developed the throttling feature for 10 years.

The third participant (**E3**) has worked as a developer in a few smartphone development fields and also has more than 5 years of development experience in GPAT. All participants were familiar with analyzing session data using the previous tool, GameBench.

Each analyst participated in the case study individually. We first demonstrated the features of our system for 20 minutes and then allowed them to freely use our system for at least 40 minutes. Since they had never used our system before, one author of this work assisted them when they had difficulties in using the system.

For the demonstration phase, a separate dataset consisting of 792 sessions was used, which was different from the dataset utilized in the free-exploration phase, to mitigate potential biases. No monetary incentives were provided for participation.

B. CASE STUDY 1: GAME AND DEVICE COMPARISON

E1 was interested in comparing the effects of two popular 3D games, "Genshin Impact" and "Odin" on the performance indicators. **E1** wanted to first view the sessions tested after 2022, so he used the slider of the Timeline View to choose the time period that he wanted. Then, **E1** proceeded to sort the games in the Game View to check the changes in the ranking of games according to the feature sorted (Figure 7A, Figure 7B). He expected that both games would have high indicator values due to their high requirements and particularly anticipated that "Genshin Impact", a visually impressive 3D graphics game, would have higher memory usage. Indeed, both games exhibited higher-than-average values for all metrics, with "Genshin Impact" in particular demonstrating the highest memory usage among the games (T1). To further compare "Genshin Impact" and "Odin", **E1** deselected all data by clicking on the "select all" checkbox and then individually checked the boxes for "Genshin Impact" and "Odin". The expert also selected 5 devices of interest in Device View, specifically the latest devices such as "Galaxy S22", for comparison. As a result, the Session list and Performance matrix were filtered to display only the sessions related to the two games.

E1 then examined the CPU-GPU relationship in the Performance matrix (Figure 7C), zooming in for a closer look (T3). "Odin" had higher GPU usage compared to "Genshin Impact" while conversely, but "Genshin Impact" had higher CPU usage. Additionally, by expanding the MEM-GPU relationship (Figure 7D), he observed that memory usage was higher for "Genshin Impact" in most sessions. Lastly, he confirmed through the chart that "Genshin Impact" had slightly higher power consumption. As a result, he determined that "Genshin Impact" was generally a heavier game compared to "Odin". However, **E1** concluded that it was meaningful to monitor both games since the system features utilized to their maximum potential differed between the two games. The expert generally showed positive responses to the comparison features of the system, as he was able to investigate aspects that he had been curious about.

We demonstrated to the expert that he could also observe trends through the overview of the session list (T2). In the Figure 7E, it can be seen that most of the "Odin" sessions, with the game displayed in blue, are distributed at the top when sessions are sorted by GPU usage. **E1** found the performance matrix to be more familiar and easily adaptable, while the overview feature was somewhat unfamiliar. However, they acknowledged that sorting each column through the overview allowed them to quickly observe trends.

C. CASE STUDY 2: EXPLORING ABNORMAL SESSIONS

We individually examined sessions where throttling occurred together with **E2** and **E3**. As in the previous case study, they filtered "Genshin Impact" sessions after 2022. In the Session List, they sorted sessions by power consumption and investigated the top-ranked sessions. The session with ID 337 was one of them, and **E2** clicked on this data, which we had preloaded with known issues. The session was saved in the Comparison View and called up in the Session View for review (T7). At this stage, the expert confessed unfamiliarity with the comparison view visualization, despite having heard a prior explanation. He mentioned having no experience with violin plots or box plots in existing commercial tools. We provided additional explanations about the visualization before continuing the demonstration. Looking at the horizon chart and PCA anomaly detection chart for session 337, throttling events were confirmed at around 11 minutes and 20 minutes (Figure 8). **E2** asked if such information could be provided in the earlier stage (session list) since throttling could be detected using the horizon chart, and the PCA chart could identify such sections. Through similarity recommendations, they verified the usefulness of sessions similar to that session, which also had occurrences of throttling or high CPU usage (T4). Lastly, the expert mentioned that predictions for sessions without power consumption data are essential for monitoring purposes (T6).

E3 also underwent the same procedure as the preceding analyst, stating that the ability to identify outliers through PCA in the Session View was beneficial.

D. USER FEEDBACK

The two experts from GPAT provided useful, abundant, and insightful feedback in two categories.

- **Limitations:** The experts who participated in the interviews agreed on the usefulness of this tool. They provided extensive feedback on how to expand the tool for practical use in the future. Particularly, they expressed interest in expanding the somewhat limited dataset used in this study due to its constraints. For example, they provided feedback that more precise predictions could be made by incorporating additional data features, such as temperature data, device cooling specifications, game relevance, chipset types, and OS information (T5). They suggested that by expanding the scope of the tool, it could establish itself as a more significant tool in the field. Moreover, **E2** provided feedback that plans should be considered to offer access to raw data, such as logs, in the future within the tool.
- **Visualization and Interaction:** The experts mentioned that they initially felt overwhelmed by the numerous views and charts, and vibrant color schemes when first encountering GameDepot. They also admitted to experiencing some difficulty in understanding the charts immediately, even after receiving explanations, due to their unfamiliarity with the types of charts used in the information visualization field. However, after the demonstration and trying out the tool step by step, they reported feeling that it was no different from the commercial tools they had been using. **E1** mentioned that while existing tools primarily provide numerical information, with GameDepot, one needs to hover over the charts or access the Session View to accurately read the numbers.

  Adding to this, **E3** expressed the opinion that it would be beneficial to have a feature to search and filter by typing the names of games and devices. As they became more accustomed to using the tool, this consideration should be taken into account in the next development cycle, potentially providing numerical values in the session list as well.

  While the experts agreed on the GameDepot's intuitive interface, **E2** raised concerns about its weak categorization capabilities, such as the lack of features for collecting selected sessions through drag actions on the plots or categorizing by date, season, or binary. **E1** also commented on the inability to select multiple sessions in the sorted session list overview from the scatterplot. Additionally, **E3** suggested the idea of implementing a feature in this tool that would allow for the creation of a baseline after sorting, such as sessions with a GPU usage of over 90%.

VI. CONCLUSION

In this paper, we propose a visual analytics system, GameDepot, designed for the analysis of testing sessions from games and devices, anomaly detection, comparison, and prediction of sessions. GameDepot has successfully provided valuable insights into the monitoring and analysis of smartphone games. It facilitates performance optimization of games and devices, improves user experience, and helps to understand relationships between features. Future research directions for GameDepot include integrating additional data sources and metrics, enhancing existing visualization and analysis tools to support a wider range of use cases, and incorporating advanced machine learning techniques and real-time data processing capabilities to improve prediction accuracy and responsiveness.

REFERENCES

[1] GameBench. Accessed: Aug. 6, 2023. [Online]. Available: https://www.gamebench.net/
[2] B. Hu, Q. Lin, Y. Zheng, Q. Yan, M. Troglia, and Q. Wang, "Characterizing location-based mobile tracking in mobile ad networks," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Jun. 2019, pp. 223–231.
[3] C. Harding, A. F. Imani, S. Srikukenthiran, E. J. Miller, and K. Nurul Habib, "Are we there yet? Assessing smartphone apps as full-fledged tools for activity-travel surveys," *Transportation*, vol. 48, no. 5, pp. 2433–2460, Oct. 2021.
[4] C. Ahlström, J. Wachtmeister, M. Nyman, A. Nordenström, and K. Kircher, "Using smartphone logging to gain insight about phone use in traffic," *Cognition, Technol. Work*, vol. 22, no. 1, pp. 181–191, Feb. 2020.
[5] Y. Jeong, H. Jung, and J. Lee, "Cyberslacking or smart work: Smartphone usage log-analysis focused on app-switching behavior in work and leisure conditions," *Int. J. Hum. Comput. Interact.*, vol. 36, no. 1, pp. 15–30, Jan. 2020.
[6] T. Deng, S. Kanthawala, J. Meng, W. Peng, A. Kononova, Q. Hao, Q. Zhang, and P. David, "Measuring smartphone usage and task switching with log tracking and self-reports," *Mobile Media Commun.*, vol. 7, no. 1, pp. 3–23, Jan. 2019.

[7] Y. Fukazawa, T. Ito, T. Okimura, Y. Yamashita, T. Maeda, and J. Ota, "Predicting anxiety state using smartphone-based passive sensing," *J. Biomed. Informat.*, vol. 93, May 2019, Art. no. 103151.

[8] E. Kanjo, E. M. G. Younis, and C. S. Ang, "Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection," *Inf. Fusion*, vol. 49, pp. 46–56, Sep. 2019.

[9] Y. Ranjan, Z. Rashid, C. Stewart, P. Conde, M. Begale, D. Verbeeck, S. Boettcher, R. Dobson, A. Folarin, and T. Hyve, "RADAR-base: Open source mobile health platform for collecting, monitoring, and analyzing data using sensors, wearables, and mobile devices," *JMIR Mhealth Uhealth*, vol. 7, no. 8, Aug. 2019, Art. no. e11734.

[10] D. A. Drew, L. H. Nguyen, C. J. Steves, C. Menni, M. Freydin, T. Varsavsky, C. H. Sudre, M. J. Cardoso, S. Ourselin, J. Wolf, T. D. Spector, and A. T. Chan, "Rapid implementation of mobile technology for real-time epidemiology of COVID-19," *Science*, vol. 368, no. 6497, pp. 1362–1367, 2020.

[11] K. Yasudomi, T. Hamamura, M. Honjo, A. Yoneyama, and M. Uchida, "Usage prediction and effectiveness verification of app restriction function for smartphone addiction," in *Proc. IEEE Int. Conf. E-Health Netw., Appl. Services (HEALTHCOM)*, Mar. 2021, pp. 1–8.

[12] F. Giunchiglia, M. Zeni, E. Gobbi, E. Bignotti, and I. Bison, "Mobile social media usage and academic performance," *Comput. Hum. Behav.*, vol. 82, pp. 177–185, May 2018.

[13] W. Yan, Q. Zhang, Y. Zhang, A. Wang, and C. Zhao, "The validation and performance assessment of the Android smartphone based GNSS/INS coupled navigation system," in *Proc. China Satell. Navigat. Conf. (CSNC)*. Cham, Switzerland: Springer, 2021, pp. 310–319.

[14] M. Singh, J. Trivedi, P. Maan, and J. Goyal, "Smartphone battery state-of-charge (SoC) estimation and battery lifetime prediction: State-of-art review," in *Proc. 10th Int. Conf. Cloud Comput., Data Sci. Eng., Confluence*, Jan. 2020, pp. 94–101.

[15] S. Sahoo, J. Mohr, G. J. Strnad, J. Vega, M. Jones, M. S. Schickendantz, L. Farrow, K. P. Spindler, J. P. Iannotti, E. T. Ricchetti, and K. A. Derwin, "Validity and efficiency of a smartphone-based electronic data collection tool for operative data in rotator cuff repair," *J. Shoulder Elbow Surg.*, vol. 28, no. 7, pp. 1249–1256, Jul. 2019.

[16] V. K. Gurbani, D. Kushnir, V. Mendiratta, C. Phadke, E. Falk, and R. State, "Detecting and predicting outages in mobile networks with log data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[17] D. Kim, D. Shin, D. Shin, and Y.-H. Kim, "Attack detection application with attack tree for mobile system using log analysis," *Mobile Netw. Appl.*, vol. 24, no. 1, pp. 184–192, Feb. 2019.

[18] A. M. Saleh and O. Enaizan, "COG tool: An automated cognitive measurement of workload for mobile event logging," *Int. J. Interact. Mobile Technol. (iJIM)*, vol. 16, no. 20, pp. 143–161, Oct. 2022.

[19] K. Guner, M. S. Q. Z. Nine, M. F. Bulut, and T. Kosar, "FastHLA: Energy-efficient mobile data transfer optimization based on historical log analysis," in *Proc. 16th ACM Int. Symp. Mobility Manage. Wireless Access (MobiWac)*, 2018, pp. 59–66.

[20] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li, J. Chen, X. He, R. Yao, J.-G. Lou, M. Chintalapati, F. Shen, and D. Zhang, "Robust log-based anomaly detection on unstable log data," in *Proc. 27th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Aug. 2019, pp. 807–817.

[21] K. Xu, Y. Wang, L. Yang, Y. Wang, B. Qiao, S. Qin, Y. Xu, H. Zhang, and H. Qu, "CloudDet: Interactive visual analysis of anomalous performances in cloud computing systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 1107–1117, Jan. 2020.

[22] X. Zhang, X. Xuan, A. Dima, T. Sexton, and K.-L. Ma, "LabelVizier: Interactive validation and relabeling for technical text annotations," 2023, *arXiv:2303.17820*.

[23] N. Tovanich, N. Soulié, N. Heulot, and P. Isenberg, "MiningVis: Visual analytics of the Bitcoin mining economy," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 868–878, Jan. 2022.

[24] K. K. Wong, X. Wang, Y. Wang, J. He, R. Zhang, and H. Qu, "Anchorage: Visual analysis of satisfaction in customer service videos via anchor events," *IEEE Trans. Vis. Comput. Graph.*, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10045801

[25] T. Polk, D. Jäckle, J. Häußler, and J. Yang, "CourtTime: Generating actionable insights into tennis matches using visual analytics," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 397–406, Jan. 2020.

[26] L. Shoukry and S. Göbel, "Realizing a mobile multimodal platform for serious games analytics," *Int. J. Serious Games*, vol. 6, no. 4, pp. 19–48, Nov. 2019.

[27] L. Shoukry, "Mobile multimodal serious games analytics," Tech. Univ., Darmstadt, Germany, Tech. Rep., 2020, doi: 10.25534/tuprints-00013539.

[28] A. Smerdov, A. Somov, E. Burnaev, and A. Stepanov, "Ai-enabled prediction of video game player performance using the data from heterogeneous sensors," *Multimedia Tools Appl.*, vol. 82, no. 7, pp. 11021–11026, 2022.

[29] E. Lee, Y. Jang, D.-M. Yoon, J. Jeon, S.-i. Yang, S.-K. Lee, D.-W. Kim, P. P. Chen, A. Guitart, P. Bertens, Á. Periáñez, F. Hadiji, M. Müller, Y. Joo, J. Lee, I. Hwang, and K.-J. Kim, "Game data mining competition on churn prediction and survival analysis using commercial game log data," *IEEE Trans. Games*, vol. 11, no. 3, pp. 215–226, Sep. 2019.

[30] E. Harpstead, T. Zimmermann, N. Nagapan, J. J. Guajardo, R. Cooper, T. Solberg, and D. Greenawalt, "What drives people: Creating engagement profiles of players from game log data," in *Proc. Annu. Symp. Comput.-Human Interact. Play*, Oct. 2015, pp. 369–379.

[31] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIX Annu. Tech. Conf.*, Boston, MA, USA, Jun. 2010, p. 21.

[32] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app? Fine grained energy accounting on smartphones with eprof," in *Proc. 7th ACM Eur. Conf. Comput. Syst.*, Apr. 2012, pp. 29–42.

[33] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2376–2385, Dec. 2013.

[34] T. Munzner, *Visualization Analysis and Design*. Boca Raton, FL, USA: CRC Press, 2014.

[35] A. Eklund, "beeswarm: The bee swarm plot, an alternative to stripchart," R Package Version 0.2, Tech. Rep., 2016. Accessed: Aug. 06, 2023. [Online]. Available: https://github.com/cran/beeswarm

[36] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, "LineUp: Visual analysis of multi-attribute rankings," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2277–2286, Dec. 2013.

[37] K. Pearson, "Contributions to the mathematical theory of evolution," *Philos. Trans. Roy. Soc. London A, Math. Phys. Sci.*, vol. 185, pp. 71–110, Jan. 1894.

[38] E. R. Tufte, "The visual display of quantitative information," *J. Healthcare Quality (JHQ)*, vol. 7, no. 3, p. 15, 1985.

[39] K. Potter, H. Hagen, A. Kerren, and P. Dannenmann, "Methods for presenting statistical information: The box plot," in *Proc. VLUDS*, 2006, pp. 97–106.

[40] E. Molina, L. Viale, and P. Vázquez, "How should we design violin plots?" in *Proc. IEEE 4th Workshop Vis. Guidelines Res., Design, Educ. (VisGuides)*, Oct. 2022, pp. 1–7.

[41] L. Byron and M. Wattenberg, "Stacked graphs—Geometry & aesthetics," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1245–1252, Nov. 2008.

DONGHOON JANG received the B.S. degree in computer engineering from Sungkyunkwan University, South Korea, in 2012, where he is currently pursuing the M.S. degree in digital media communication (DMC) engineering.

Currently, he is with Samsung Electronics. His research interests include human–computer interaction, information visualization, mobile game performance, and mobile user experience.

JAEMIN JO received the B.S. and Ph.D. degrees in computer science and engineering from Seoul National University, Seoul, South Korea, in 2014 and 2020, respectively.

He is currently an Assistant Professor with the College of Computing and Informatics, Sungkyunkwan University, South Korea. His research interests include human–computer interaction and large-scale data visualization.