SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
SAUCIS
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 4, 701-717, 2025
DOI: 10.35377/saucis...1681525
SAKARYA UNIVERSITY
RESEARCH ARTICLE

Predictive Machine Learning Modeling of Performance Issues in Virtual Reality Devices Using Synthetic Dataset

Osman Güler[1*], Taha Etem [2]

[1,2] Cankiri Karatekin University, Faculty of Engineering, Computer Engineering, Cankiri, Türkiye, ror.org/011y7xt38

Corresponding author
Osman Güler, Cankiri Karatekin University, Faculty of Engineering, Computer Engineering, Cankiri\ Turkiye
osmanguler@karatekin.edu.tr

Article History:
Received: 22.04.2025
Revised: 12.09.2025
Accepted: 16.09.2025
Published Online: 13.10.2025

ABSTRACT
This study presents a novel approach to predicting and analyzing performance problems in standalone Virtual Reality (VR) devices through the development of a comprehensive synthetic dataset and machine learning methodology. The research created a synthetic dataset simulating the performance of ten different standalone VR devices, incorporating both technical specifications and real-time performance metrics. The dataset generation process considered realistic device behavior patterns, including temperature variations under different load conditions, performance degradation factors, and network-related issues. The methodology employed seven different machine learning models. The dataset comprised 14,400 samples, with data collected at 5-second intervals over 120-minute sessions. Results demonstrated exceptional performance from tree-based models, with Random Forest and Decision Tree achieving near-perfect accuracy (99.97%). Extreme Gradient Boosting (99.69%) and Neural Network (98.92%) also showed strong performance. The study found that Overheating and Packet Loss predictions were particularly accurate across most models, while High Latency classification proved more challenging for some algorithms due to class imbalance. The synthetic dataset and methodology offer a foundation for future research in VR system optimization and real-time performance monitoring. The study addresses a significant gap in the literature by integrating both hardware specifications and performance metrics into a comprehensive analysis framework.

Keywords: Virtual reality, Performance prediction, Machine learning, Synthetic dataset, Overheating

1. Introduction

Virtual Reality (VR) is a type of digital technology that has been widely used in every aspect of our lives in recent years, reshaping human-computer interaction, that is, the way people interact with digital environments [1] [2]. VR technology separates users from the physical environment and offers them immersive experiences in a virtual world created entirely in a digital environment, such as education, simulation, and entertainment. [3]. The immersive virtual reality experience provides a first-person perspective system through a head-mounted display (HMD) that allows users to experience virtual worlds realistically [4]. In 2021, with Mark Zuckerberg changing the name of his social platforms to Meta, people imagined an environment where they could interact in a 3-dimensional (3D) virtual world on online platforms with HMD and live without real-world obstacles [5]. Among the technical challenges for this imagined metaverse environment is the need for a new and independent HMD as user interface hardware to provide users with an immersive experience [6]. Today, these immersive and independent HMD hardware are called VR Glasses (VR-G) [7]. With these recent developments in VR technology, VR-G and interactive VR applications have begun to be developed at lower costs, and consumer access has become easier [8]. With these technological developments and changes, VR-G has gained popularity.

Although it has been widely used in recent years, the beginning of VR technology dates back to earlier times. In 1962, a device capable of providing multi-sensory feedback during film experiences called Sensorama was introduced by film producer Morton Heilig [9]. This invention is considered one of the earliest attempts in the field of VR. Sensorama used a head-mounted display to play static video and sensory stimuli. In 1968, Ivan Edward Sutherland, known as a pioneer in the field of computer graphics, created the first head-mounted display that created images according to the changing pose of the viewer, thus creating the first VR system, as shown in Figure 1 [10].

Cite as: O. Güler & T. Etem, "Predictive machine learning modeling of performance issues in virtual reality devices using synthetic dataset," *Sakarya University Journal of Computer and Information Sciences,* vol. 8, no. 4, pp. 701-717, 2025. doi: 10.35377/saucis...1681525

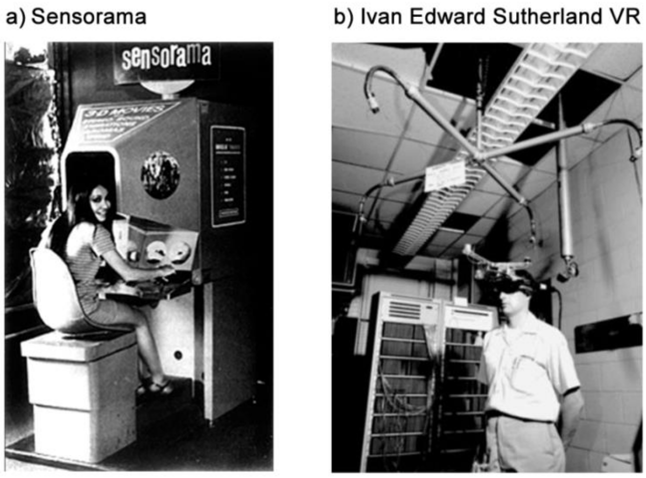

Figure 1. a) Sensorama b) Ivan Edward Sutherland's VR platform

VR-G is equipped with sensors to track the head, hand and even pupil movements of users with pose tracking to provide the illusion of movement in a virtual environment, has a screen for the image and headphones for the audio transmission [11], [12], [13]. VR technology consists of different devices characterized by different levels of immersion [14]. Today, VR-G can be grouped into three types. These are the Tethered VR-G, Standalone VR-G, and Mobile Phone VR-G. The standalone VR-G includes all the necessary computing hardware components, such as CPU, GPU, memory, screen, and battery [15]. Tethered VR glasses are devices that work by connecting to a computer or game console with a cable. [16]. Another type, Mobile phone VR-G, is a VR-G that operates independently, similar to a standalone VR-G, but does not contain any internal hardware. A compatible smartphone is placed in this VR-G chamber to experience VR [17].

Figure 2. a) Standalone VR-G b) Tethered VR-G c) Mobile phone VR-G

The use of 3D environments and 3D devices for an immersive VR experience increases the user experience, enjoyment and satisfaction level [18]. To provide an immersive experience, VR-G must first meet optical performance requirements such as field of view (FoV), eye box, angular resolution, dynamic range and correct depth, making people feel like they are in the 3D environment [19]. For optimal optical performance, a realistic 3D virtual environment design is necessary to give the user a sense of reality, which requires creating depth perception, i.e., a stereo view, within the person. For this, a sequence of images created from two different perspectives is transferred to the VR-G screen, and a different image is transmitted to each eye [20], [21]. In this way, users perceive the distance between objects, i.e., depth, and an immersive experience occurs.

A well-designed virtual environment and stereo view are crucial for an immersive experience with VR-G, as well as the quality of these experiences, which depends on the VR-G features, performance, and stable operation used [22]. Problems such as lag, interruption, and overheating that may occur during use can negatively impact performance and user experience, potentially causing health issues like nausea and dizziness. The performance of VR devices depends on many factors. These include the technical specifications of the device (resolution, refresh rate, RAM, CPU, GPU), usage time, ambient temperature, and network connection quality [23]. Processing high-resolution content and high refresh rates can strain a device's hardware resources, leading to overheating and subsequent performance issues. Additionally, VR-G can overheat and potentially cause interruptions to long-term use. Delays in tracking user movements with sensors can result in unpredictable user movements and significant playback interruptions [24]. Therefore, continuously monitoring the performance of VR-Gs and predicting potential problems in advance are critical to improving the user experience. Studies on performance analysis and interruption prediction generally focus on hardware and software optimization. Various cooling methods have been developed to reduce overheating problems [25], [26]. Similarly, solution methods have been developed for latency problems that occur during the processing of high-resolution content [27]. However, these studies usually focus on a specific device or scenario and do not provide a universal methodology. This deficiency reveals the need for a comprehensive dataset and methodology to analyze the performance of VR devices and predict interruption situations.

In this study, to address the deficiencies in the literature, a synthetic dataset simulating the performance of VR-Gs was created, and a comprehensive methodology was presented to predict the interruption status and type of performance problem using this dataset. In this context, the technical specifications of the devices, such as resolution, refresh rate, RAM, CPU, and GPU, as well as performance metrics, including FPS, latency, temperature, and packet loss, were taken into account. In addition, factors such as overheating, low FPS, high latency, and packet loss, which cause the interruption status, were analyzed to determine the type of performance problem and the interruption status. The results obtained can be considered as an important step in the performance analysis and interruption prediction of VR-Gs. In future studies, machine learning models can be developed using this dataset, and comparisons can be made with real-world data. In Figure 3, the general structure of the proposed system is given, and the main contribution of the paper is as follows:

- Created a comprehensive synthetic dataset that simulates VR device performance by incorporating both technical specifications (e.g., resolution, refresh rate, hardware components) and performance metrics (e.g., FPS, latency, temperature, packet loss).
- Established a methodology that systematically analyzes multiple factors contributing to VR system outages, such as overheating, low FPS, high latency, and packet loss, thereby enabling accurate prediction of interruption status and performance problem types.
- Provided a controlled, reproducible experimental framework that serves as a benchmark for the development and testing of future machine learning models aimed at outage prediction in VR systems.
- Addressed a notable gap in the literature by integrating both hardware performance metrics and outage-causing factors, thus offering a more holistic approach to VR performance analysis.
- Laid the groundwork for further research by highlighting the potential for subsequent validation with real-world data, which will help in refining predictive models and extending the methodology's applicability across diverse VR configurations.

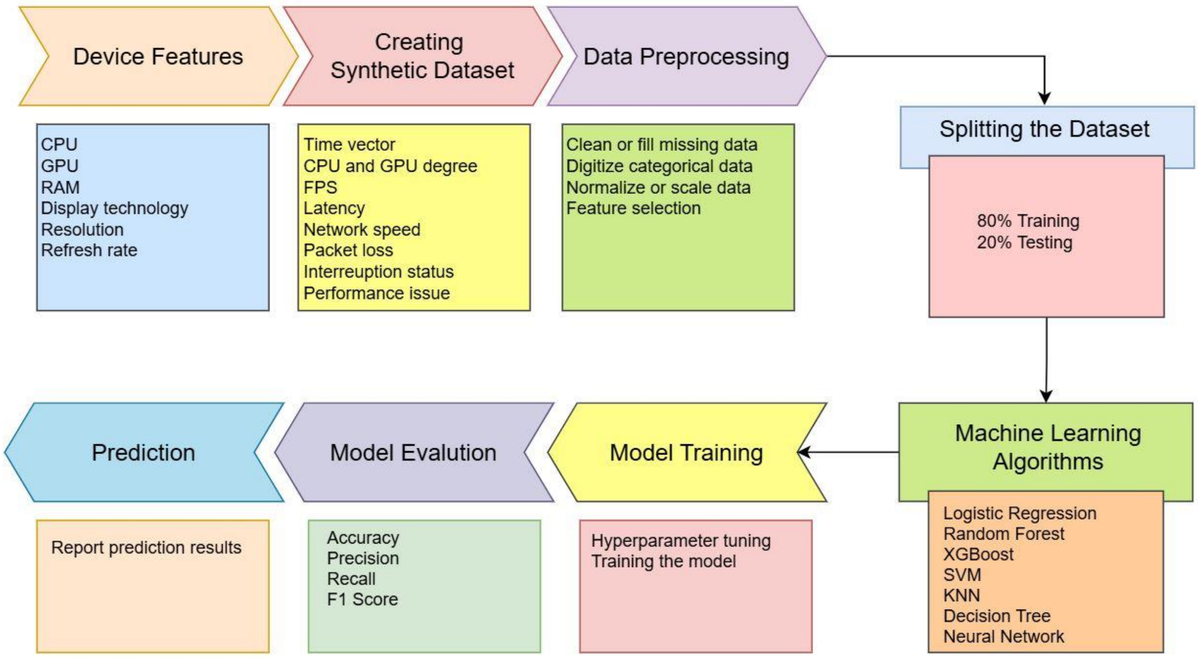

Figure 3. General Structure of the Proposed VR Performance Prediction System

The flowchart in Figure 3 highlights the selection of device features, including CPU, GPU, RAM, display technology, resolution, and refresh rate, which serve as the foundational inputs for assessing VR device performance. These parameters are chosen to capture both hardware capabilities and software-level constraints, thereby providing a comprehensive view of potential performance bottlenecks. Once these features are identified, a synthetic dataset is generated to reflect a diverse range of operational scenarios. This dataset simulates variations in CPU and GPU usage, FPS, latency, network speed, packet loss, and other critical factors that can precipitate performance degradation or interruptions in VR systems.

After constructing this dataset, the process moves to data preprocessing, where any missing or corrupted values are handled through cleaning and imputation methods. This step also includes the digitization of relevant parameters, scaling of numeric features, and a feature selection procedure aimed at retaining only the most salient predictors. Such preprocessing not only ensures data quality but also enhances the robustness of subsequent model training efforts. Once the data has been refined, it is split into training and testing sets to enable a fair assessment of model generalizability.

With the dataset partitioned, multiple machine learning algorithms are employed to classify the type of performance issue. Machine Learning algorithms are systematically trained using hyperparameter tuning to optimize their predictive accuracy. The trained models are then evaluated on the testing set using standard performance metrics, providing a comprehensive measure of each algorithm's capability to detect performance anomalies. Finally, the workflow concludes in the prediction stage, where the best-performing model is deployed to predict performance problems. By integrating hardware features, synthetic data generation, meticulous preprocessing, and a rigorous machine learning pipeline, the flowchart outlines a robust methodology for enhancing the reliability and efficiency of VR systems.

2. Related Works

A study evaluated the performance of three video see-through (VST) head-mounted displays (HMDs), including two prototypes (PD1/PD2) and one commercial device (CD1). Through their mixed-methods research design, they measured both quantitative data points, such as visual acuity, alongside depth perception and latency, while 20 participants provided qualitative feedback. Three tasks comprised reading, object handling, and performing virtual navigation. The commercial device (CD1) maintained superior resolution and accurate color output compared to the prototypes, yet it generated higher visual disturbances for users as they moved. The virtual prototypes gained preference because they provided comfort alongside a lower incidence of cybersickness, although their display quality was substandard. The research established that successful user needs management requires optimized hardware and software technology adjustments for different functional requirements [28].

Another study examined the effect that both constant and sinusoidal display lag had on cybersickness in a group of 50 people using HMDs. A research investigation evaluated the variance in virtual and physical head pose across space and time to assess correlations between the measured values and levels of sickness. The data showed that increased cybersickness existed under all lag time conditions, yet failed to demonstrate differences between these lag types compared to the baseline data. Factors that affect stability before starting VR exposure contribute to the development of cybersickness symptoms, while head movements within VR sessions have a direct effect on sickness severity. VR design principles should focus on display lag reduction because these findings demonstrate that user vulnerability to adverse effects depends on design features and personal characteristics [29].

A paper introduced a foldable haptic device designed to simulate shape and texture feedback in VR environments. Linear resonant actuators, in combination with a 1-bar mechanism, enabled the portable delivery of realistic tactile sensations using the device. User tests demonstrated that the haptic device successfully enhanced users' ability to perceive textures and shapes; however, it still needs to improve its functionality for advanced haptic control. The official release of the device, along with open-source access, supported the promotion of haptic technology development and its wider adoption. The study concluded that foldable haptic devices hold promise for improving VR immersion without adding significant bulk to the hardware [30].

A study compared the performance of traditional 2D steady-state visual evoked potential (SSVEP) stimuli with 3D stereoscopic stimuli in a VR headset (PICO Neo3 Pro). Participants performed 4-object selection tasks using luminance modulation (3D paradigm) and opacity modulation (3D-Blink paradigm). Results showed that the 3D paradigm achieved 85.67% accuracy, slightly lower than the 86.17% accuracy of the 2D paradigm, but with reduced signal-to-noise ratios. The 3D-Blink paradigm, while reducing visual fatigue, had lower accuracy (75%). The study concluded that stereoscopic stimuli improve user comfort but require algorithmic adjustments to enhance performance in brain-computer interface (BCI) applications [31].

Another study systematically evaluated six VR anatomy applications, including 3D Organon VR Anatomy and VEDAVI VR Human Anatomy, in terms of functionality, accuracy, and usability. Feedback was collected from 120 medical students and 6 professors. Results indicated that 3D Organon and Human Anatomy VR excelled in system classification and interactivity but lacked disease visualization. Apps like Sharecare YOU Anatomy integrated disease states, but had limited magnification tools. Key gaps identified included inconsistent anatomical terminology and insufficient tutorials for novice users. The study emphasized the need for standardized terminology and better onboarding processes to maximize the educational utility of VR anatomy applications [32].

A study investigated the impact of thermal management in VR headsets on user comfort and device performance. Researchers tested three high-end HMDs under prolonged use with 27 participants, measuring skin temperature, subjective comfort ratings, and rendering performance. Results showed that devices with active cooling systems maintained stable frame rates but caused localized facial discomfort due to airflow noise and uneven heat distribution. Conversely, passively cooled headsets exhibited thermal throttling but were rated as more comfortable. The study concluded that balancing thermal design with user comfort requires hybrid cooling solutions and adaptive rendering algorithms [33].

Another paper proposed a neural rendering framework to reduce latency in wireless VR devices. The method used deep learning to predict and pre-render frames based on head movement patterns, tested on 30 participants using a Meta Quest Pro. Compared to traditional rendering, the neural approach reduced perceived latency by 32% and improved motion-to-photon times. However, participants reported occasional visual artifacts during rapid head turns. The study emphasized the potential of AI-driven rendering to enhance wireless VR performance, though artifact mitigation remains a challenge for widespread adoption [34].

Another study explored the efficiency of foveated rendering on standalone VR devices. Using eye-tracking data from 47 participants, researchers compared the detection of different expressions across five scenarios: anger, closed eyes, happiness, neutral, and surprise. The findings advocate eye-tracking integration in next-gen standalone VR devices to optimize robustness to makeup and hair colors. The work showed that eyes can encode a large amount of information about human expressions [35].

A study explored VR as a platform for navigation and selection tasks to investigate how different levels of immersion affect spatial memory and selection performance. The results revealed that while higher immersion negatively impacted selection performance and did not enhance spatial learning, most participants still felt that immersion helped them learn object locations and preferred the immersive setup with a head-mounted display. These findings suggest that developers should not automatically assume that immersive VR will improve the performance of everyday software applications [36].

A comprehensive paper examines the recent progress in display technologies, with a specific focus on AR and VR. The study begins by analyzing advancements in AR and VR near-eye displays (NEDs), which have undergone significant evolution in terms of field of view (FOV), eye box size, and form factor. For VR displays, notable innovations include the development of lenslet arrays and polarization folding optics, enabling the creation of devices with sub-centimeter thickness while maintaining a wide FOV. The paper also covers significant progress in OLED technology, particularly in the development of blue OLEDs and the implementation of thermally activated delayed fluorescence (TADF) and hyper fluorescence (HF) mechanisms. The study concludes by highlighting how these technologies are approaching or surpassing the performance of traditional liquid crystal displays (LCDs), with each offering unique advantages for specific applications [37].

While prior research has substantially advanced the evaluation of various aspects of VR systems, the present study offers a novel perspective by employing a synthetic dataset to simulate the performance and interruption phenomena of VR devices. Firstly, there is limited research about the latency, FPS, and thermal behavior of VR devices. One of the key advantages of this approach is its comprehensive integration of technical specifications (resolution, refresh rate, RAM, CPU, GPU) alongside performance metrics (FPS, latency, temperature, and packet loss). This dual consideration facilitates a more granular investigation of the multifaceted factors that precipitate outages, such as overheating, low FPS, high latency, and network issues. In contrast to many studies that rely on empirical data from limited hardware configurations or isolated performance indicators, the synthetic dataset enables controlled experimentation and reproducibility. It also provides a robust platform for the future development of machine learning models for predicting VR performance. However, this strength is accompanied by a notable limitation: the synthetic nature of the dataset may not fully capture the variability and unpredictability inherent in real-world VR environments. Consequently, while the proposed methodology establishes an important benchmark and addresses key gaps in the literature, its real-world applicability will benefit from subsequent validation with actual device data to further enhance robustness and generalizability.

Table 1. Device Specifications of the Dataset

Device Name	Resolution	Refresh Rate	RAM	CPU	GPU	Display
Oculus Quest 2	1832x1920	120 Hz	6	Octa-core Kryo 585 (1 x 2.84 GHz, 3 x 2.42 GHz, 4 x 1.8 GHz)	Adreno 650	LCD
Meta Quest 3	2064x2208	120 Hz	8	Octa-core Kryo (1 x 3.19 GHz, 4 x 2.8 GHz, 3 x 2.0 GHz)	Adreno 740	LCD
Meta Quest Pro	1800x1920	90 Hz	12	Octa-core Kryo 585 (1 x 2.84 GHz, 3 x 2.42 GHz, 4 x 1.8 GHz)	Adreno 650	QLED
Meta Quest 3S	1832x1920	120 Hz	8	Octa-core Kryo (1 x 3.19 GHz, 4 x 2.8 GHz, 3 x 2.0 GHz)	Adreno 740	LCD
HTC Vive Focus Vision	2448x2448	90 Hz	12	Octa-core Kryo 585 (1 x 2.84 GHz, 3 x 2.42 GHz, 4 x 1.8 GHz)	Adreno 650	LCD
HTC Vive XR Elite	1920x1920	90 Hz	12	Octa-core Kryo 585 (1 x 2.84 GHz, 3 x 2.42 GHz, 4 x 1.8 GHz)	Adreno 650	LCD
Pico 4 Ultra	2160x2160	90 Hz	12	Octa-core Kryo (1 x 3.19 GHz, 4 x 2.8 GHz, 3 x 2.0 GHz)	Adreno 740	LCD
Pico 4	2160x2160	90 Hz	8	Octa-core Kryo 585 (1 x 2.84 GHz, 3 x 2.42 GHz, 4 x 1.8 GHz)	Adreno 650	LCD
Lenovo ThinkReality VRX	2280x2280	90 Hz	12	Octa-core Kryo 585 (1 x 2.84 GHz, 3 x 2.42 GHz, 4 x 1.8 GHz)	Adreno 650	LCD
Lenovo Legion VR700	1832x1920	90 Hz	8	Octa-core Kryo 585 (1 x 2.84 GHz, 3 x 2.42 GHz, 4 x 1.8 GHz)	Adreno 650	LCD

3. Dataset Creation

In this study, a synthetic dataset was created on the MATLAB platform to simulate VR-G performance and predict interruption status. The dataset generates realistic performance metrics (FPS, latency, temperature) by considering the technical specifications of the devices (e.g., resolution, refresh rate, RAM, CPU, GPU) and usage time. The performance metrics are used to predict the interruption status and the Type of Performance Problem. In the scope of the study, 10 different standalone VR-G features using the Android operating system were given as input to the simulation environment. The following VR-G features were used as input parameters while creating the dataset:

- Resolution: The screen resolution of the device (width x height).
- Refresh Rate: The refresh rate of the screen per second (Hz).
- RAM (Memory): The memory capacity of the device (GB).
- CPU Type and Speed: The processor type and clock speed (GHz).
- GPU Type: The graphics processing unit (GPU) type.
- Display Technology: Display technology such as LCD or QLED.

Using the VR-G features given in Table 1, the following performance metrics were produced in the simulation environment. Considering that VR-Gs operate for an average of 120 minutes, 1440 samples were generated for each VR-G, one sample every 5 seconds.

CPU Temperature (°C): The temperature value of the processor.

GPU Temperature (°C): The temperature value of the graphics processor.

Processor Utilization (%): The utilization rate of the CPU.

Memory Usage (MB): The amount of RAM used.

Screen Refresh Rate (FPS): Frames per second.

Latency (ms): The response time between the user and the game.

Network Speed (Mbps): The speed of the network connection.

Packet Loss Rate (%): The rate of packet loss on the network connection.

The formulas used to generate CPU/GPU temperature, FPS, and latency in the synthetic dataset were determined by combining multiple complementary sources to ensure both theoretical validity and practical realism. Previous research on VR performance and thermal behavior provided empirical ranges for parameters such as heating and cooling rates, frame rate thresholds, and latency fluctuations. Furthermore, manufacturer specification sheets for devices such as the Oculus Quest, Meta Quest, and HTC Vive were carefully reviewed to define the operational limits of the equations, including safe temperature ranges, refresh rate limits, and expected latency conditions. By integrating findings from the literature on hardware specifications and user feedback, the formulas were designed to mimic realistic device behavior while maintaining mathematical simplicity for repeatability and accuracy. For example, the heat rate coefficient in the temperature model reflects both the faster thermal buildup of older processors and the moderate heating observed in newer architectures. Similarly, the FPS formula balances resolution, refresh rate, and system memory, while the latency equation accounts for the impact of CPU timing and memory usage. This multi-source approach ensures that the synthetic dataset approximates the operational dynamics of real-world VR devices during simulation. According to the performance metrics produced, the Interruption status and Performance problem type of the VR glasses are estimated as the Dependent Variable. While Interruption consists of two classes, it is considered that:

- **1:** There is an interruption.
- **0:** No downtime.

Performance Issue Type consists of 5 classes as follows:

- **Overheating:** When the CPU temperature exceeds 40°C or the GPU temperature exceeds 45°C.
- **Low FPS:** When the FPS value drops below 60.
- **High Latency:** When the latency exceeds 100 ms.
- **Packet Loss:** When the packet loss rate exceeds 0.2%.
- **Normal:** If there are no interruptions.

The following formulas and methods were used while creating the data set:

CPU & GPU Temperature:

Initial Temperatures are shown in Equations 1 and 2.

$$CPU_{Temp(1)} = 30 + 1 * rand(x) \tag{1}$$

$$GPU_{Temp(1)} = 35 + 5 * rand(x) \tag{2}$$

Here, rand() generates a random number between 0 and 1.

Temperature Change is shown in Equations 3 and 4.

$$CPU_{Temp(t)} = CPU_{Temp(t-1)} + (heating_{rate} * load_{factor(t)}) + (noise_{level} * randn(x)) \tag{3}$$

$$GPU_{Temp(t)} = GPU_{Temp(t-1)} + \left(heating_{rate} * load_{factor(t)}\right) + (noise_{level} * randn(x)) \tag{4}$$

Here:

heating_rate: Temperature rise rate (0.05) in case of high load.

load_factor(t): The load factor at the time step (between 0 and 1).

noise_level: Random fluctuation level (0.1).

Tendency to Cool is given in Equations 5 and 6.

$$CPU_{Temp(t)} = CPU_{Temp(t-1)} + \left(cooling_{rate} * \left(1 - load_{factor(t)}\right)\right) + (noise_{level} * randn(x)) \tag{5}$$

$$GPU_{Temp(t)} = GPU_{Temp(t-1)} + \left(cooling_{rate} * \left(1 - load_{factor(t)}\right)\right) + (noise_{level} * randn(x)) \tag{6}$$

Here, cooling_rate indicates the rate of temperature decrease in the case of low load (-0.02).

The FPS (Frames Per Second) formula is shown in Equation 7.

$$FPS(t) = Refresh_{Rate} + 30 * randn(x) - \left(\left(\frac{Resolution}{10^6}\right) * 5\right) - (RAM < 8) * 10 - \left(CPU_{Temp(t)} > 35\right) * 5 \tag{7}$$

Here, the parameters indicate:

Refresh_Rate: The refresh rate (Hz) of the device.

Resolution: The resolution (width x height) of the device.

RAM: The memory capacity (GB) of the device.

CPU_Temp(t): The CPU temperature at the time step.

Latency calculations are shown in Equation 8.

$$Latency(t) = 50 + (20 * randn(x)) + \left(\left(\frac{Resolution}{10^6}\right) * 5\right) + ((RAM < 8) * 10) + \left(\left(CPU_{Temp(t)} > 35\right) * 5\right) \tag{8}$$

The interruption is calculated as shown in Equation 9.

$$Interruption = \begin{cases} 1, & CPU\_Temp(t) > 40,\ GPU\_Temp(t) > 45,\ FPS(t) < 60, \\ & Latency(t) > 100,\ Packet\_Loss(t) > 0.2 \\ 0, & else \end{cases} \tag{9}$$

The type of performance issue is determined by the factors that caused the interruption. Each factor represents a different performance issue. The type of bottleneck is calculated by the following formulas. :The overheating calculation is given in Equation 10. Equations 11, 12, 13, and 14 define the performance issue.

$$Performance_{Issue(t)} = \text{Overheating } if\ CPU_{Temp(t)} > 40, GPU\_Temp(t) > 45 \tag{10}$$

$$Performance_{Issue}(t) = \text{"}Low\ FPS\text{"}\ if\ FPS(t) < 60 \tag{11}$$

$$Performance_{Issue}(t) = \text{"High Latency"} \ if \ \ Latency(t) > 100 \quad (12)$$

$$Performance_{Issue(t)} = \text{"Packet Loss"} \ \ if \ \ Packet\_Loss(t) > 0.2 \quad (13)$$

$$Performance_{Issue(t)} = \text{"Normal"} \ if \ Interruption(t) = 0 \quad (14)$$

When the CPU temperature exceeds 40°C, the performance issue is determined as "CPU Overheating". When the GPU temperature exceeds 45°C, the performance issue is determined as "GPU Overheating". When the FPS value drops below 60, the performance issue is determined as "Low FPS". When the latency exceeds 100 ms, the bottleneck is designated as "High Latency". When the packet loss rate exceeds 0.2%, the bottleneck is designated as "Packet Loss". If there are no interruptions, the performance issue is set to "Normal". The algorithm for creating the VR Device Performance Dataset includes the following steps:

1. Create the Time vector (from 1 to 1440).
2. Set initial CPU and GPU temperatures.
3. Generate load factor randomly.
4. Always for step:
    4.1. Calculate CPU and GPU temperatures (based on load factors).
    4.2. Calculate FPS and latency.
    4.3. Determine the interruption status.
    4.4. Determine the type of performance issue.
5. Add all data to a table.

The sample data set structure of the created dataset is presented in Table 2, and the visualization of the dataset's features is shown in Figure 4.

Table 2. Virtual Reality Google Devices Dataset Features

Device	**t**	**CPU Temp**	**GPU Temp**	**CPU Usage**	**RAM Usage**	**FPS**	**Latency**	**Network Speed**	**Packet Loss**	**Inter**	**Performance Issue**
Meta Quest 3	1	32.1	36.3	75.3	7000	110	50	55.6	0.12	0	Normal
Meta Quest 3	2	32.5	36.8	76.1	7050	108	52	54.8	0.13	0	Normal
Oculus Quest 2	1	41.5	46.8	85.7	4500	90	65	52.3	0.25	1	Overheating

The dataset is designed to have similar characteristics to real-world data. CPU and GPU temperature values gradually increase or decrease according to the load factor. The load factor determines whether the device is under load. In high-load conditions, the temperature increases rapidly, while in low-load conditions, it decreases gradually. Older processors and GPUs heat up faster and reach higher temperatures. Newer-generation devices heat up more slowly, but there may be sudden temperature increases under high loads. Factors such as resolution, RAM, and temperature have a significant impact on FPS and latency. FPS drop is calculated based on temperature and CPU usage. Latency fluctuates according to CPU and RAM capacity. Interruption status and Performance Issue Type are determined based on performance metrics.

This synthetic dataset can be used to analyze the performance of VR devices and predict the interruption status. The dataset provides independent variables to predict the interruption status and type of performance issue, taking into account the technical specifications and performance metrics of the devices.

4. Materials and Methods

This study employed seven machine learning models to predict VR device performance issues using a synthetic dataset. Below, the methodologies for each algorithm are detailed, including implementation frameworks, hyperparameters, and optimization strategies.

4.1 Dataset Description

The synthetic dataset simulated performance metrics across 10 standalone VR devices (Table 1), generating 1440 samples for each device with 5-second intervals over 120-minute sessions. Features included hardware specifications (resolution, CPU/GPU type) and dynamic performance metrics (CPU/GPU temperature, packet loss). The target variable comprised five classes: Latency, FPS, Normal, Overheating, and Packet Loss. Data was partitioned into 80% training and 20% testing sets.

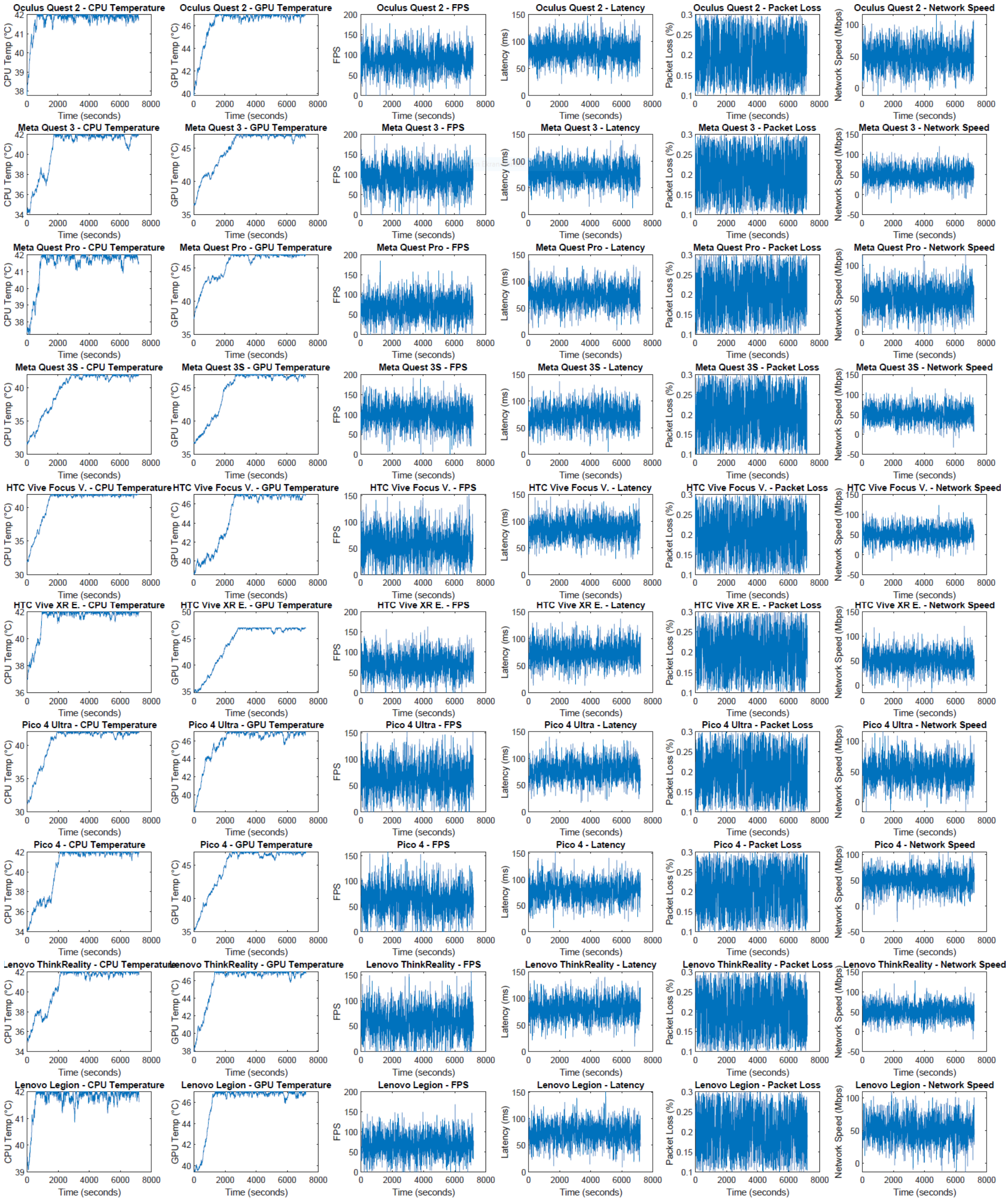

Figure 4. Standalone VR-G performance metric: CPU Temperature, GPU Temperature, FPS, Latency, Packet Loss, Network Speed, respectively.

4.2 Experimental Setup

All models were implemented in Python 3.9 using a TensorFlow environment. Hyperparameters were tuned via grid search with 5-fold cross-validation on the training set. Performance was evaluated using accuracy, precision, recall, and F1-score. Calculations of the metrics are shown in the following Equations. Class-specific metrics were macro-averaged to account for imbalance.

- Accuracy: $\frac{Correct\ Predictions}{Total\ Samples}$ (15)
- Precision: $\frac{TP}{TP+FP}$ (16)
- Recall: $\frac{TP}{TP+FN}$ (17)
- F1-score: $2 * \frac{Precision*Recall}{Precision+Recall}$ (18)

4.3 Model Configurations

4.3.1. Logistic Regression

Logistic Regression (LR) was applied as a baseline linear classifier. The multinomial variant with L2 regularization was used to handle the multi-class problem. The solver, limited-memory BFGS, optimized the cross-entropy loss, with a maximum of 1,000 iterations and a regularization strength (C) of 1.0. Features were standardized to a zero mean and unit variance to mitigate scale sensitivity [38].

4.3.2. Random Forest

A Random Forest (RF) ensemble of 100 decision trees was trained, with bootstrap sampling and Gini impurity for node splitting. To prevent overfitting, trees were limited to a maximum depth of 20, and a minimum of 5 samples was required to split a node. Feature randomness was introduced by considering $\sqrt{n}$ features at each split (n = total features) [39].

4.3.3. XGBoost

The Extreme Gradient Boosting (XGBoost) model utilized gradient-boosted trees with the following hyperparameters: learning rate (η) = 0.1, maximum tree depth = 6, and 200 estimators. The objective function softprob is optimized for multi-class classification, with L2 regularization (λ = 1) to penalize complex trees. Early stopping was applied if validation loss plateaued for 10 consecutive epochs [40].

4.3.4. Support Vector Machine (SVM)

A non-linear SVM with a radial basis function (RBF) kernel was implemented. The regularization parameter C was set to 10, and the kernel coefficient γ was scaled as γ=1n·Var(X)γ=n·Var(X)1. Due to computational constraints, the dataset was subsampled to 50% for hyperparameter tuning [41].

4.3.5. K-Nearest Neighbors (KNN)

KNN classified samples based on majority voting among the 5 nearest neighbors. Euclidean distance was used as the similarity metric, and weights were inversely proportional to distance. To address class imbalance, the k value is optimized during training [42].

4.3.6. Decision Tree

A single Decision Tree (DT) was trained using Gini impurity splitting, with an unlimited depth and a minimum of 2 samples per leaf. Pruning was avoided to maximize model flexibility, as the synthetic dataset lacked noise [43].

4.3.7. Neural Network

A feedforward Neural Network (NN) with three hidden layers (128, 64, and 32 units) was designed using TensorFlow. The architecture employed ReLU activation for hidden layers and softmax for the output layer. Training used the Adam optimizer (learning rate = 0.001) with categorical cross-entropy loss over 100 epochs. Batch normalization and dropout (rate = 0.3) were applied to reduce overfitting [44].

4.4.Training and Validation

All models were trained on an NVIDIA RTX 4070 GPU. For tree-based models (RF, XGBoost, DT), feature importance was analyzed to identify critical predictors (e.g., CPU temperature, resolution). The Neural Network's training process included a 10% validation split for early stopping. After Grid search optimization, the following parameters are obtained.

- **LR**: C ∈ {0.1, 1, 10}
- **RF/XGBoost**: max_depth ∈ {5, 10, 20}, n_estimators ∈ {50, 100, 200}
- **SVM**: C ∈ {1, 10, 100}, γ ∈ {0.1, 0.01}
- **KNN**: k ∈ {3, 5, 7}

5. Results and Discussion

The proposed methodology for predicting VR device performance issues was evaluated using seven machine learning models: Logistic Regression, Random Forest, XGBoost, SVM, KNN, Decision Tree, and Neural Network. The synthetic dataset (14,400 samples) included five classes: High Latency, Low FPS, Normal, Overheating, and Packet Loss. Key results are summarized in Table 3 as follows.

Table 3. Performance Metrics of the models, including Logistic Regression, Random Forest, XGBoost, SVM, KNN, Decision Tree, and Neural Network, are presented in the following order.

Model	**LOGISTIC REGRESSION**			
	Precision	**Recall**	**F1-Score**	**Support**
High Latency	0.84	0.86	0.85	174
Low FPS	0.94	0.95	0.95	423
Normal	1.00	1.00	1.00	94
Overheating	0.95	0.96	0.96	659
Packet Loss	0.99	0.98	0.98	1530
accuracy			0.96	2880
macro avg	0.94	0.95	0.95	2880
weighted avg	0.96	0.96	0.96	2880
MODEL	**RANDOM FOREST**			
	Precision	**Recall**	**F1-Score**	**Support**
High Latency	1.00	0.99	1.00	174
Low FPS	1.00	1.00	1.00	423
Normal	1.00	1.00	1.00	94
Overheating	1.00	1.00	1.00	659
Packet Loss	1.00	1.00	1.00	1530
accuracy			1.00	2880
macro avg	1.00	1.00	1.00	2880
weighted avg	1.00	1.00	1.00	2880
MODEL	**XGBOOST**			
	Precision	**Recall**	**F1-Score**	**Support**
High Latency	0.99	0.98	0.99	174
Low FPS	0.99	1.00	0.99	423
Normal	1.00	1.00	1.00	94
Overheating	0.99	1.00	0.99	659
Packet Loss	1.00	1.00	1.00	1530
accuracy			1.00	2880
macro avg	1.00	0.99	0.99	2880
weighted avg	1.00	1.00	1.00	2880
MODEL	**SVM**			
	Precision	**Recall**	**F1-Score**	**Support**
High Latency	0.90	0.89	0.89	174

Table 1 (Continued)				
Low FPS	0.96	0.95	0.96	423
Normal	1.00	1.00	1.00	94
Overheating	0.97	0.96	0.97	659
Packet Loss	0.98	0.98	0.98	1530
accuracy			0.97	2880
macro avg	0.96	0.96	0.96	2880
weighted avg	0.97	0.97	0.97	2880
MODEL	**KNN**			
	Precision	**Recall**	**F1-Score**	**Support**
High Latency	0.79	0.62	0.70	174
Low FPS	0.78	0.76	0.77	423
Normal	1.00	1.00	1.00	94
Overheating	0.80	0.85	0.82	659
Packet Loss	0.94	0.94	0.94	1530
accuracy			0.88	2880
macro avg	0.86	0.83	0.85	2880
weighted avg	0.88	0.88	0.88	2880
MODEL	**DECISION TREE**			
	Precision	**Recall**	**F1-Score**	**Support**
High Latency	1.00	1.00	1.00	174
Low FPS	1.00	1.00	1.00	423
Normal	1.00	0.99	0.99	94
Overheating	1.00	1.00	1.00	659
Packet Loss	1.00	1.00	1.00	1530
accuracy			1.00	2880
macro avg	1.00	1.00	1.00	2880
weighted avg	1.00	1.00	1.00	2880
MODEL	**NEURAL NETWORK**			
	Precision	**Recall**	**F1-Score**	**Support**
High Latency	0.93	0.99	0.96	174
Low FPS	1.00	0.97	0.98	423
Normal	1.00	1.00	1.00	94
Overheating	0.99	0.98	0.98	659
Packet Loss	0.99	1.00	1.00	1530
accuracy			0.99	2880
macro avg	0.98	0.99	0.98	2880
weighted avg	0.99	0.99	0.99	2880

Overall performance of the models deployed can be summarized as follows:

- Random Forest and Decision Tree achieved near-perfect accuracy (99.97%), demonstrating exceptional performance in classifying VR device issues.
- XGBoost (99.69%) and Neural Network (98.92%) followed closely, while Logistic Regression (96.42%) and SVM (96.94%) showed strong but comparatively lower performance.
- KNN lagged (87.67%), likely due to its sensitivity to feature scaling and non-linear decision boundaries in the dataset.
- Overheating and Packet Loss were predicted with near-flawless precision and recall (F1-scores $\geq$ 0.96 for all models except KNN). These classes dominated the dataset (659 and 1,530 samples, respectively), enabling robust pattern recognition.
- High Latency (174 samples) proved more challenging, particularly for KNN (F1-score: 0.70) and Logistic Regression (F1-score: 0.85), reflecting class imbalance effects.
- Normal (94 samples) was perfectly classified by most models, indicating clear separation between nominal and anomalous states.

While the synthetic dataset successfully mimics the behavior of VR devices, it cannot fully reflect real-world variability. For example, environmental noise, such as room temperature fluctuations and inconsistent Wi-Fi interference, user-induced variations, including different head/hand movement patterns and perspiration that affect thermal readings, and hardware aging parameters, including battery degradation and thermal paste efficiency degradation, were not modeled. In real-world data, such factors can lead to overlap between classes and degrade the near-perfect performance observed in the Random Forest and Decision Tree models.

Feature importance analysis focusing on latency classification revealed that RAM capacity, CPU utilization spikes, and network instability contribute most to misclassifications. In particular, latency is strongly affected by temporal changes in CPU timing and network jitter, which are less pronounced in the synthetic dataset, making it difficult for models to distinguish subtle latency patterns.

Tree-based models (Random Forest, Decision Tree, XGBoost) performed exceptionally well, likely due to their ability to handle non-linear relationships between hardware specifications and performance metrics. Neural Networks showed strong performance but required more computational resources, suggesting a trade-off between accuracy and efficiency. In Figure 4, the Confusion matrices of the deployed models are given.

The confusion matrices for the models highlight distinct performance patterns in classifying VR device performance issues. The Decision Tree achieved near-perfect classification, as evidenced by dominant diagonal values (1550 correct predictions for Packet Loss and 659 for Overheating) and minimal off-diagonal errors (only 4 misclassifications in total). This aligns with its 99.97% accuracy, underscoring its ability to capture clear decision boundaries in the synthetic dataset. In contrast, the KNN confusion matrix revealed significant misclassification, particularly for minority classes like Latency and FPS, which contributed to its lower overall accuracy (87.67%). The sparse off-diagonal entries in the KNN matrix reflect its sensitivity to class imbalance and non-linear feature relationships. In contrast, the Decision Tree's robustness to these challenges is evident in its tightly clustered predictions. These results reinforce the superiority of tree-based models for VR performance data, where hierarchical splitting effectively resolves multi-class interdependencies.

6. Conclusions

This study addressed the critical challenge of predicting performance issues in VR devices by developing a synthetic dataset and evaluating seven machine learning models. Random Forest algorithms, together with Decision Trees, proved effective in solving performance issues, as they demonstrate perfect accuracy in detecting overheating, latency, and packet loss problems. The models demonstrated their suitability for operational VR monitoring through their ability to explain hardware performance in non-linear processes within VR devices. Moving forward, the analysis of these findings should test them using real-life hardware data to confirm their applicability beyond synthetic datasets. The proposed method presents a comprehensive structure that improves VR device performance while simultaneously improving user satisfaction and reliability outcomes. The research focuses on solving fundamental VR performance obstacles to advance the technology, enabling better integration of virtual experiences.

The comparative analysis of seven different machine learning models revealed that tree-based algorithms, particularly Random Forest and Decision Tree, excel at classifying VR performance issues, achieving nearly perfect accuracy at 99.97%. The models demonstrate superior performance because they effectively detect complex, nonlinear patterns that link hardware specifications and performance metrics. The high levels of accuracy demonstrated by XGBoost (99.69%) and Neural Network (98.92%) support the methodological strength of the approach, while also highlighting alternative methods for VR performance prediction.

The creation of a synthetic dataset incorporating both technical specifications and performance metrics has addressed a crucial gap in the literature. The dataset provides valuable research material for optimizing VR systems by creating realistic simulations of device reactions that include performance breakdowns, network issues, and temperature fluctuations. The methodology successfully predicts diverse performance problems, indicating that it can be deployed for real-time monitoring and management of VR systems.

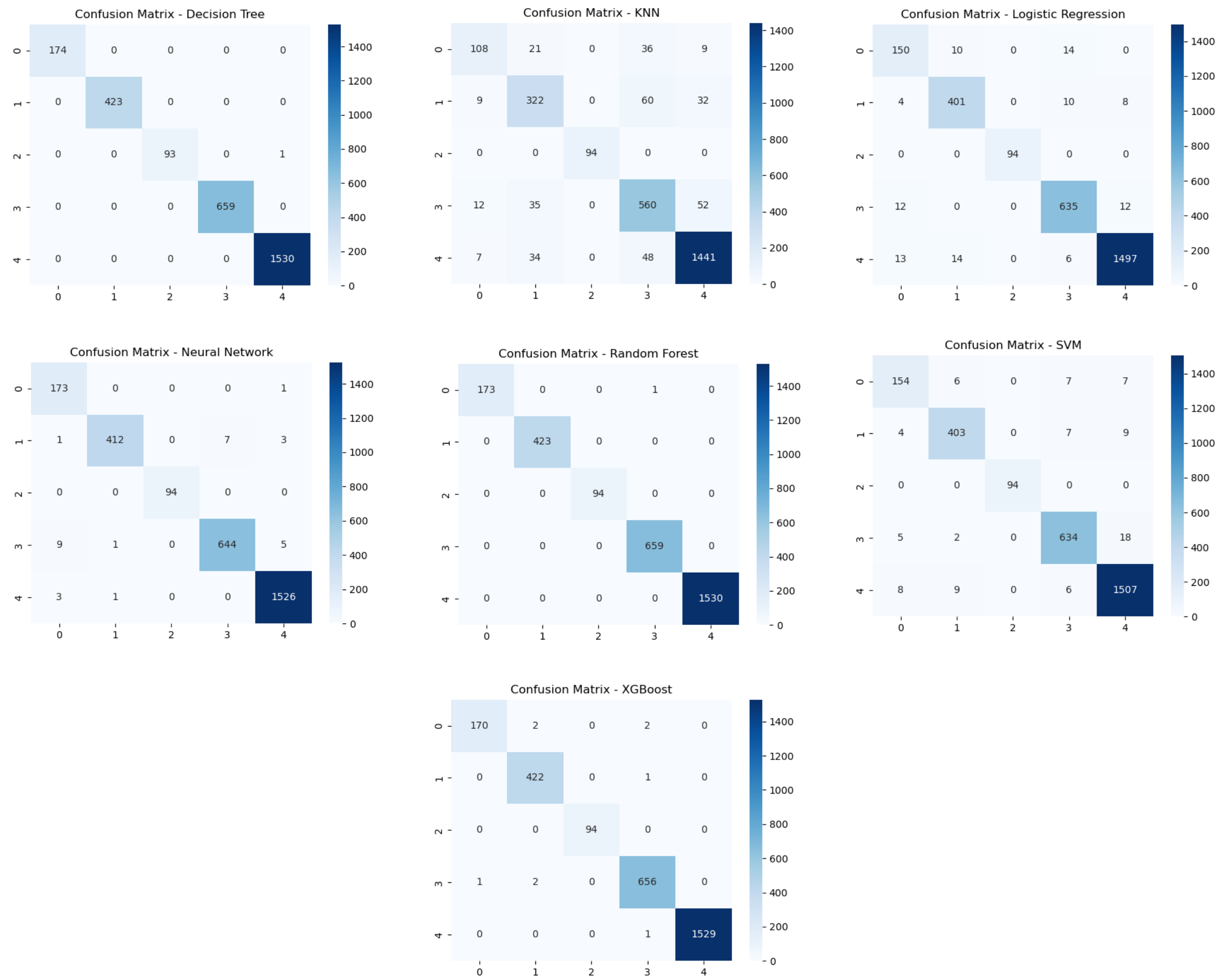

Figure 4. Confusion Matrices of Decision Tree, KNN, Logistic Regression, Neural Network, Random Forest, SVM, XGBoost, respectively.

However, it is important to acknowledge the limitations of this study. The synthetic data characteristics enable experimental control, but they fail to represent real-life VR usage conditions accurately. Some models achieve near-perfect accuracy, which implies that the synthetic data may fail to capture the complete variance and noise present in actual VR platforms.

In future work, we aim to validate our models using real-world datasets. Such data can be collected through controlled sessions using recording tools integrated into VR headsets. However, challenges include device diversity, different hardware generations, user variability, session duration, motion intensity, and environmental factors. Addressing these challenges is crucial for building robust and generalizable prediction systems.

Future studies could benefit from the multiple promising opportunities presented in this research. Future development of this methodology should involve integrating user patterns with environmental variables and multiple scenarios of interaction. To achieve the practical adoption of the models, it will be essential to validate them using data collected from various VR devices. Additionally, the development of real-time monitoring systems based on these predictive models could help prevent performance issues before they impact user experience. The advancement of VR technology creates an urgent need for effective prediction and prevention strategies regarding performance problems. The methodology could serve as a foundation for developing more robust and reliable VR systems, ultimately contributing to better user experiences in virtual environments.

References

[1] A. Gong, Y. Cheng, J. Su, and L. Zhang, "Research on hybrid synchronization methods in multi-user collaborative VR simulation medical surgery training system," *Concurr Comput*, vol. 36, no. 11, p. e8008, May 2024, doi: 10.1002/CPE.8008.

[2] Y. Ye, H. Liu, W. Kuang, and W. Chen, "Virtual reality modeling application based on multi-perspective and deep learning in the new media presentation and brand building of Dongguan City memory," *Concurr Comput*, vol. 36, no. 11, p. e8015, May 2024, doi: 10.1002/CPE.8015.

[3] H. Namrouti, C. Sik-Lányi, and T. Guzsvinecz, "Exploring measurement tools for color perception in virtual reality: A systematic review," *Displays*, vol. 87, p. 102937, Apr. 2025, doi: 10.1016/J.DISPLA.2024.102937.

[4] P. Campo-Prieto, J. M. Cancela-Carral, and G. Rodríguez-Fuentes, "Wearable Immersive Virtual Reality Device for Promoting Physical Activity in Parkinson's Disease Patients," *Sensors (Basel)*, vol. 22, no. 9, May 2022, doi: 10.3390/S22093302.

[5] O. Güler and S. Savaş, "All Aspects of Metaverse Studies, Technologies and Future," *Gazi Journal of Engineering Sciences*, vol. 8, no. 2, pp. 292–319, Sep. 2022, doi: 10.30855/GMBD.0705011.

[6] M. Jansen, J. Donkervliet, A. Trivedi, and A. Iosup, "Can My WiFi Handle the Metaverse? A Performance Evaluation Of Meta's Flagship Virtual Reality Hardware," *ICPE 2023 - Companion of the 2023 ACM/SPEC International Conference on Performance Engineering*, pp. 297–303, Apr. 2023, doi: 10.1145/3578245.3585022.

[7] T. Erol Akar and S. Ünver, "Effectiveness of Virtual Reality Glasses on Surgical Fear and Anxiety in Patients Before Open-heart Surgery: A Double-blind Randomized Controlled Trial," *J Perianesth Nurs*, 2024, doi: 10.1016/J.JOPAN.2024.08.011.

[8] B. Xie *et al.*, "A Review on Virtual Reality Skill Training Applications," *Front Virtual Real*, vol. 2, p. 645153, Apr. 2021, doi: 10.3389/FRVIR.2021.645153/PDF.

[9] W. H. Wan, A. Ho, Y. Lam, W. H. Wan, and A. H. Y. Lam, "The Effectiveness of Virtual Reality-Based Simulation in Health Professions Education Relating to Mental Illness: A Literature Review," *Health N Hav*, vol. 11, no. 6, pp. 646–660, Jun. 2019, doi: 10.4236/HEALTH.2019.116054.

[10] T. Li, "The Art Foundation of Virtual Reality Interactivity," in *Application of Intelligent Systems in Multi-modal Information Analytics*, Springer Science and Business Media Deutschland GmbH, 2022, pp. 952–958. doi: 10.1007/978-3-031-05484-6_127.

[11] C. Chung and S. H. Lee, "Continuous Prediction of Pointing Targets With Motion and Eye-Tracking in Virtual Reality," *IEEE Access*, vol. 12, pp. 5933–5946, 2024, doi: 10.1109/ACCESS.2024.3350788.

[12] L. Wang, X. Li, J. Wu, D. Zhou, I. Sio Kei, and V. Popescu, "AVICol: Adaptive Visual Instruction for Remote Collaboration Using Mixed Reality," *Int J Hum Comput Interact*, vol. 41, no. 2, pp. 1260–1279, 2025, doi: 10.1080/10447318.2024.2313920.

[13] J. Liebers, S. Brockel, U. Gruenefeld, and S. Schneegass, "Identifying Users by Their Hand Tracking Data in Augmented and Virtual Reality," *Int J Hum Comput Interact*, vol. 40, no. 2, pp. 409–424, 2024, doi: 10.1080/10447318.2022.2120845.

[14] L. Delbes, N. Mascret, C. Goulon, and G. Montagne, "Validation of an immersive virtual reality device accepted by seniors that preserves the adaptive behavior produced in the real world," *Front Bioeng Biotechnol*, vol. 10, Sep. 2022, doi: 10.3389/FBIOE.2022.917486.

[15] V. Angelov, E. Petkov, G. Shipkovenski, and T. Kalushkov, "Modern Virtual Reality Headsets," *HORA 2020 - 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, Jun. 2020, doi: 10.1109/HORA49412.2020.9152604.

[16] G. Goncalves, P. Monteiro, M. Melo, J. Vasconcelos-Raposo, and M. Bessa, "A Comparative Study between Wired and Wireless Virtual Reality Setups," *IEEE Access*, vol. 8, pp. 29249–29258, 2020, doi: 10.1109/ACCESS.2020.2970921.

[17] K. Aslan, Z. Özer, and M. K. Yöntem, "Effect of Virtual Reality on Pain, Anxiety, and Vital Signs in Endoscopy," *Pain Management Nursing*, Jan. 2025, doi: 10.1016/J.PMN.2024.11.009.

[18] A. A. Laghari, A. K. Jumani, K. Kumar, and M. A. Chhajro, "Systematic Analysis of Virtual Reality & Augmented Reality," *International Journal of Information Engineering and Electronic Business*, vol. 13, no. 1, pp. 36–43, Feb. 2021, doi: 10.5815/IJIEEB.2021.01.04.

[19] J. Xiong, E. L. Hsiang, Z. He, T. Zhan, and S. T. Wu, "Augmented reality and virtual reality displays: emerging technologies and future perspectives," *Light Sci Appl*, vol. 10, no. 1, Dec. 2021, doi: 10.1038/S41377-021-00658-8.

[20] O. Güler and S. Savaş, "Stereoscopic 3D teaching material usability analysis for interactive boards," *Comput Animat*

Virtual Worlds, vol. 33, no. 2, p. e2041, Mar. 2022, doi: 10.1002/CAV.2041.

[21] M. Guo, H. Gao, S. Yang, K. Yue, Y. Liu, and Y. Wang, "Evaluation of stereoscopic visual fatigue in virtual reality with exploration of brain dynamics," *Displays*, vol. 87, p. 102898, Apr. 2025, doi: 10.1016/J.DISPLA.2024.102898.

[22] D. Cui and C. Mousas, "Effects of Tactile Interruption on Hand-Eye Coordination Task Performance," *Proceedings - 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, VRW 2024*, pp. 104–111, 2024, doi: 10.1109/VRW62533.2024.00024.

[23] J. Louca, K. Eder, J. Vrublevskis, and A. Tzemanaki, "Impact of Haptic Feedback in High Latency Teleoperation for Space Applications," *ACM Trans Hum Robot Interact*, vol. 13, no. 2, p. 21, Jun. 2024, doi: 10.1145/3651993/ASSET/6BA95049-F946-4101.

[24] Z. Hou, C. She, Y. Li, D. Niyato, M. Dohler, and B. Vucetic, "Intelligent Communications for Tactile Internet in 6G: Requirements, Technologies, and Challenges," *IEEE Communications Magazine*, vol. 59, no. 12, pp. 82–88, Dec. 2021, doi: 10.1109/MCOM.006.2100227.

[25] Z. Zou, D. Prasetyawan, H. H. Wu, K. Cheng, and T. Nakamoto, "Extension of Wearable Olfactory Display for Multisensory VR Experience," in *International Conference on Artificial Reality and Telexistence Eurographics Symposium on Virtual Environments*, 2024. doi: 10.2312/egve.20241374.

[26] K. Ragozin, X. Meng, R. Lalintha Peiris, K. Wolf, G. Chernyshov, and K. Kunze, "Thermoquest-awearable head mounted display to augment realities with thermal feedback," *ACM International Conference Proceeding Series*, pp. 62–66, May 2021, doi: 10.1145/3490632.3490649.

[27] A. Sarkar, J. Murray, M. Dasari, M. Zink, and K. Nahrstedt, "L3BOU: Low Latency, Low Bandwidth, Optimized Super-Resolution Backhaul for 360-Degree Video Streaming," *Proceedings - 23rd IEEE International Symposium on Multimedia, ISM 2021*, pp. 138–147, 2021, doi: 10.1109/ISM52913.2021.00031.

[28] J. de Souza and R. Tartz, "Visual perception and user satisfaction in video see-through head-mounted displays: a mixed-methods evaluation," *Front Virtual Real*, vol. 5, p. 1368721, Jun. 2024, doi: 10.3389/FRVIR.2024.1368721/BIBTEX.

[29] S. Palmisano, V. Morrison, R. S. Allison, R. G. Davies, and J. Kim, "Effects of Constant and Sinusoidal Display Lag on Sickness During Active Exposures to Virtual Reality," *Proceedings - 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, VRW 2024*, pp. 757–758, 2024, doi: 10.1109/VRW62533.2024.00175.

[30] Y. Sung, D. K. Kwak, T. Kim, W. Woo, and S. H. Yoon, "Deep-Texture: A Lightweight Wearable Ring for Shape and Texture Rendering in Virtual Reality," *Proceedings - 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, VRW 2024*, pp. 911–912, 2024, doi: 10.1109/VRW62533.2024.00252.

[31] H. Liu *et al.*, "A comparative study of stereo-dependent SSVEP targets and their impact on VR-BCI performance," *Front Neurosci*, vol. 18, p. 1367932, Apr. 2024, doi: 10.3389/FNINS.2024.1367932/BIBTEX.

[32] S. W. Baek *et al.*, "Systematic analysis of anatomy virtual reality (VR) apps for advanced education and further applications," *Scientific Reports 2024 14:1*, vol. 14, no. 1, pp. 1–34, Dec. 2024, doi: 10.1038/s41598-024-82945-z.

[33] Z. Wang, R. He, and K. Chen, "Thermal comfort and virtual reality headsets," *Appl Ergon*, vol. 85, p. 103066, May 2020, doi: 10.1016/J.APERGO.2020.103066.

[34] S. Y. Park and D. K. Koo, "The Impact of Virtual Reality Content Characteristics on Cybersickness and Head Movement Patterns," *Sensors (Basel)*, vol. 25, no. 1, p. 215, Jan. 2025, doi: 10.3390/S25010215.

[35] S. Hickson, V. Kwatra, N. Dufour, A. Sud, and I. Essa, "Eyemotion: Classifying facial expressions in VR using eye-tracking cameras," *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*, pp. 1626–1635, Mar. 2019, doi: 10.1109/WACV.2019.00178.

[36] K. Kargut, C. Gutwin, and A. Cockburn, "Efects of Device Environment and Information Layout on Spatial Memory and Performance in VR Selection Tasks," *Conference on Human Factors in Computing Systems - Proceedings*, May 2024, doi: 10.1145/3613904.3642486/.

[37] H. J. Jang, J. Y. Lee, G. W. Baek, J. Kwak, and J. H. Park, "Progress in the development of the display performance of AR, VR, QLED and OLED devices in recent years," *Journal of Information Display*, vol. 23, no. 1, pp. 1–17, 2022, doi: 10.1080/15980316.2022.2035835.

[38] Ö. Akbulut, A. Kaygısız, and İ. Yılmaz, "A Comparative Research on Data Analysis with Factorial ANOVA, Logistic Regression and CHAID Classification Tree Methods," *Black Sea Journal of Agriculture*, vol. 5, no. 3, pp. 314–322, Jul. 2022, doi: 10.47115/BSAGRICULTURE.1087820.

[39] S. Angadi and V. S. Reddy, "Multimodal sentiment analysis using reliefF feature selection and random forest classifier," *International Journal of Computers and Applications*, vol. 43, no. 9, pp. 931–939, Oct. 2021, doi:

10.1080/1206212X.2019.1658054.

[40] L. Jovanovic *et al.*, "Improving Phishing Website Detection Using a Hybrid Two-level Framework for Feature Selection and XGBoost Tuning," *Journal of Web Engineering*, vol. 22, no. 3, pp. 543–574–543–574, Jul. 2023, doi: 10.13052/JWE1540-9589.2237.

[41] S. Buyrukoğlu and A. Akbaş, "Machine Learning based Early Prediction of Type 2 Diabetes: A New Hybrid Feature Selection Approach using Correlation Matrix with Heatmap and SFS," *Balkan Journal of Electrical and Computer Engineering*, vol. 10, no. 2, pp. 110–117, Apr. 2022, doi: 10.17694/BAJECE.973129.

[42] S. Buyrukoğlu and S. Savaş, "Stacked-Based Ensemble Machine Learning Model for Positioning Footballer," *Arab J Sci Eng*, vol. 48, no. 2, pp. 1371–1383, Feb. 2023, doi: 10.1007/s13369-022-06857-8.

[43] M. Teke and F. Duran, "The design and implementation of road condition warning system for drivers," *Measurement and Control (United Kingdom)*, vol. 52, no. 7–8, pp. 985–994, Sep. 2019, doi:10.1177/0020294019858088.

[44] G. Buyrukoğlu, S. Buyrukoğlu, and Z. Topalcengiz, "Comparing Regression Models with Count Data to Artificial Neural Network and Ensemble Models for Prediction of Generic Escherichia coli Population in Agricultural Ponds Based on Weather Station Measurements," *Microb Risk Anal*, vol. 19, Dec. 2021, doi: 10.1016/J.MRAN.2021.100171.

Article Information Form

Authors Contributions

Authors are solely responsible for the design, execution, analysis, and writing of this study.

Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval

It is declared that during the preparation process of this study, scientific and ethical principles were adhered to, and all studies cited are listed in the bibliography.

Availability of data and material

All data and materials related to this study are available from the corresponding authors upon reasonable request.

Artificial Intelligence Statement

No artificial intelligence tools were used while writing this article.

Plagiarism Statement

This article has been scanned by iThenticate ™.