electronics

MDPI

Article

Gaze-Based Interaction Intention Recognition in Virtual Reality

Xiao-Lin Chen [1,2] and Wen-Jun Hou [1,3,*]

[1] Beijing Key Laboratory of Network Systems and Network Culture, Beijing University of Posts and Telecommunications, Beijing 100876, China; cxl95@163.com
[2] School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3] School of Digital Media and Design Arts, Beijing University of Posts and Telecommunications, Beijing 100876, China
* Correspondence: hou1505@163.com

Abstract: With the increasing need for eye tracking in head-mounted virtual reality displays, the gaze-based modality has the potential to predict user intention and unlock intuitive new interaction schemes. In the present work, we explore whether gaze-based data and hand-eye coordination data can predict a user's interaction intention with the digital world, which could be used to develop predictive interfaces. We validate it on the eye-tracking data collected from 10 participants in item selection and teleporting tasks in virtual reality. We demonstrate successful prediction of the onset of item selection and teleporting with an 0.943 F_1-Score using a Gradient Boosting Decision Tree, which is the best among the four classifiers compared, while the model size of the Support Vector Machine is the smallest. It is also proven that hand-eye-coordination-related features can improve interaction intention recognition in virtual reality environments.

Keywords: intention prediction; virtual reality; gaze-based interaction

Citation: Chen, X.-L.; Hou, W.-J. Gaze-Based Interaction Intention Recognition in Virtual Reality. *Electronics* **2022**, *11*, 1647. https://doi.org/10.3390/electronics11101647

Academic Editors: Jorge C. S. Cardoso, André Perrotta, Paula Alexandra Silva and Pedro Martins

Received: 15 April 2022
Accepted: 18 May 2022
Published: 21 May 2022

1. Introduction

The Metaverse has recently attracted a great deal of attention in industry and academia, especially after Facebook changed its name to Meta. If the Metaverse is realized in the future, extended reality technology, including virtual reality technology, will be one of its essential supporting technologies. Biocca and Delaney [1] define virtual reality (VR) as "the sum of the hardware and software systems that seek to perfect an all-inclusive, sensory illusion of being present in another environment". The core characteristics of VR are immersion, interaction and imagination [2]. Immersion and interaction mean higher requirements for human–computer interaction in VR systems. Interaction should be more natural and intuitive. The first step is to identify and understand the interaction that the user wants to perform so that the system can provide appropriate help in time. Interaction intent recognition enables the system to provide shortcuts to the user by predicting the intended interaction, facilitating the interaction, and reducing the operational load of the user. For example, if the system knows what object the user would like to interact with within the virtual environment, it can connect a certain input command to the inferred interaction target and allow the user to complete the entire interaction without manual pointing, which can greatly reduce the physical and cognitive load of the user. Especially under the concept of the Metaverse, 24/7-wearable AR and VR devices for production and work are facing the problem that prolonged usage can exacerbate fatigue, so adaptive interaction interface that can accurately predict the interaction intention of the user has the potential to reinvent human–computer interaction under extended reality.

Research on the application of eye tracking in VR and human–computer interaction began early [3], but has not been widely used due to the cost and accuracy of the eye-tracking equipment. In 2017, we witnessed the acquisitions of companies that can provide eye-tracking technology by well-known companies in VR and augmented reality, highlighting the importance of eye tracking in this field. In 2019, companies such as FOVE Inc.,

Microsoft and HTC had already provided systems with built-in eye-tracking for professional and consumer markets. The applications of eye movements in VR fall into four main categories [4]: diagnostic (eye-movement behavior analysis), active (as a human–computer interface), passive (gaze-contingent rendering), and expressive (synthesizing eye movements of virtual avatars). This research mainly focuses on active applications; that is, eye movement as a human–computer interface.

A drawback in gaze-based interfaces is the Midas touch problem, i.e., unintentionally activated commands while the user is looking at an interactive element [5]. Fixation or dwell time is an indicator of an intention of the user to select an object through eye gaze alone [6–9]. However, this time threshold can negatively impact the user experience. For example, when the required dwell time is too short, it puts pressure on the user to look away and avoid unwanted selection. On the contrary, it may result in a longer wait time if it is too long [10]. If the interaction intention of the user can be recognized through natural eye-movement behavior rather than intentional, the mental and operational load of the user can be greatly reduced. Another common way to avoid the Midas touch problem is using a physical trigger as a confirmation mechanism, such as a hand controller or keyboard [6,8,11–13]. In such a case, it also makes sense to recognize the interaction intent to simplify physical buttons' operation or give more information as visual feedback based on the recognition result.

The eye has been said to be a mirror to the soul or window into the brain. This may be the first reason eye movements have attracted researchers' interest. There are many studies related to eye movements in the field of attention [14–18]. Eye movements can indicate areas of interest (active or passive attraction) and quantify the changes in human attention. Therefore, they are widely used in visual attention modeling. Eye movements can also reflect human perception [19], cognitive state [20,21], decision-making processes [22,23], and working memory [15]. Eye movements have also been used in studies of human activity classification [24–27], especially in human–computer interaction [24,27–32].

These studies have demonstrated that human eye-movement behavior can be significantly different across activities. All of the above studies focus on understanding human behavior and thinking through eye movements, which is a prerequisite and basis for the application of eye movements in intention recognition. Gaze behavior reflects cognitive processes and can give hints of our thinking and intentions.

An intention is an idea or plan of what you will do. A great deal of existing gaze-based intention recognition research aims to recognize the intention of daily human behavior [25,26,33–35] or higher-level intention involving game strategy [36]. The interaction intention in this study is the way that the user wants to interact with the computer system, i.e., to identify the interaction intention of the user before he/she performs the actual interaction. However, the interaction intent we want to identify here is low-level intent; more specifically, the intent to perform an interaction without involving complex contextual relationships and specific interaction environments. Similar to the task-independent interaction intent prediction studied by Brendan et al. [37], the application context of our study is in VR.

Our approach tracks the eye movements of the user in controller-based interaction in VR and fuses the eye movements and hand-eye coordination information collected via gaze and controller to predict the current intention of the user. Briefly, our research is conducted as follows. Initially, we collect controller and gaze data in two controller-based interaction tasks in VR (selection and teleporting) and build a multimodality database. We then extract gaze-based features from this database and train intention recognition models using supervised machine-learning techniques. Finally, we use a separate dataset to verify the accuracy of our models. The main contributions of this paper are as follows:

- We introduce a new dataset of human interaction intentions behind human gaze and hand behaviors. It contains gaze-and controller-related data of selection and teleporting in VR from multiple participants.

- We propose a gaze-controller-based feature-set representation based on human vision and behavioral studies to predict user intention through the gaze. These features are neither subject nor interface specific.
- We train four classifiers with supervised machine-learning and evaluate them in several aspects, including F_1-Score and model size. In addition, we perform feature selection to assess the relevance and redundancy of feature representations. The experimental results show that for behaviors from different people, the Gradient Boosting Decision Tree (GBDT) approach achieves F_1-Score of 0.924 for binary classification and 0.953 for three-class classification. Such results offer the possibility of a more natural implementation of the interaction interface paradigm, i.e., more intelligent delivery of low-cost interaction patterns by providing the right interventions at the right time.

Section 2 gives an outline of state-of-the-art gaze-based intention recognition studies. Our approach consists of three major parts: data collection, feature extraction, and intention recognition. They are detailed in Section 3. Section 4 compares and analyzes different classifiers' classification performance and feature importance. Section 5 includes a discussion of our work and a summary of future directions. Section 6 concludes our work.

2. Related Work

The term intent has different definitions in different fields. To avoid ambiguity, the term interaction intention in this study needs to be clarified. In human–computer interaction, the intent is either explicit or implicit. An explicit intent is directly input into the system through the interaction interface. Implicit intent involves the internal activities of users. It requires the system to infer the intentions based on some hints such as natural facial expressions, behaviors, and eye movements. This is a key feature of intelligent interactive interfaces, i.e., understanding the current state of users and predicting the following action. The ultimate goal of our research is to enable computer systems, like humans, to understand and predict users' behavior and purpose for intuitive and safe interaction. VanHorenbeke and Peer [38] explore human behavior, planning, and goal (intent) recognition as a holistic problem. They argue that behaviors and goals are incremental in granularity (i.e., a series of behaviors constitute intentions) and in time (i.e., behavior recognition focuses more on actions that occur simultaneously, while intention recognition focuses on upcoming actions). On the other hand, planning is more complex, focusing more on the relationship between a series of behaviors or intentions and the specific meaning in the semantic context in the interaction. In our study, interaction intention recognition is the least fine-grained intention recognition. Let us consider the action of pressing a button. The expected interaction result behind the series of actions, including finding a specific location and pressing it, is the "interaction intent" in this study, i.e., selection or teleporting. We do not consider the deeper intent of winning a game or switching to a better visual perspective, i.e., the interaction intent is relatively weakly linked to the semantic context of the interaction.

Eye movements are a common source of information in intention or behavior recognition. Table 1 summarizes the research on using eye-related data to classify daily behaviors and intention classification in computer environments. According to the table, the most commonly used classification algorithms include Support Vector Machine (SVM), Logistic Regression (LR), and Random Forest (RF). Our study also chooses to perform a cross-sectional comparison of these classification algorithms. These studies are also aimed at different environments. The application environments of the above studies are mainly personal computers or tablets, and there are relatively few studies in VR. Our study is to recognize interaction intention of the user in controller-based interaction in VR based on eye-movement data.

Table 1. Task, activity, and intention classification studies using eye movement data.

Reference	Year	Platform	Scope	Classifier	Performance	Tasks/Activities/Intentions
[39]	2014	PC	Intention recognition	Nearest Neighborhood (NN) Support Vector Machine (SVM)	Average accuracy: 79.81 ± 4.93 Average accuracy: 85.26 ± 0.70	Navigational intent Informational intent
[40]	2014	PC	Intention recognition	Support Vector Machine (SVM)	Average accuracy: 90%	Navigational intent Informational intent
[41]	2017	PC	Intention recognition	Nearest Neighborhood (NN) Support Vector Machine (SVM)	Average accuracy: 85%	Unintentional intention Purposeful intention
[42]	2012	PC	Intention prediction	Support Vector Machine (SVM)	ROC-AUC: 0.807 Accuracy: 76%	Issue a command or not
[43]	2018	PC	Intention prediction	Support Vector Machine (SVM)	Accuracy: 77.2%	Monitoring Tracking Decision Burst Off loop
[36]	2013	PC	Cognitive states prediction	Support Vector Machine (SVM)	Best accuracy: 32 %	8-tiles puzzle game: Cognitions Evaluations Plans Intentions Current move
[28]	2004	PC	Activity recognition	-	-	Reading comprehension Mathematical reasoning Searching Object manipulation
[29]	2011	PC	Activity recognition	LHMM	Accuracy: 51.3% Accuracy: 89.1%	Evaluate website traffic task E-Learning quiz task
[30]	2013	PC	Activity recognition	Logistic Regression	Average accuracy: 53.18%	Retrieve values Filter Compute derived value Find extremum Sort
[24]	2018	PC	Activity recognition	Support Vector Machine (SVM) K-Nearest Neighbour (K-NN) Random Forest	F1 score: SVM 0.71 K-NN 0.61 Random Forest 0.73	Read Watch Browse Play Search Interpret Debug Write
[44]	2015	Reality	Intention prediction	Support Vector Machine (SVM)	Accuracy: 76%	Making sandwich
[33]	2009	Reality	Activity recognition	Support Vector Machine (SVM)	Average precision: 76.1% Average recall: 70.5%	Copy Read Write Video Browse Null
[34]	2011	Reality	Activity recognition	Support Vector Machine (SVM)	Average accuracy: 80.2% Average precision: 76.1% Average recall: 70.%	Reading or not reading Copy, read, write video, browse, null Visual memory (familiar/unfamiliar images)
[35]	2012	Reality	Activity recognition	Support Vector Machine (SVM)	Mean average precision 57%	Copy Read Write Video Browse Null
[25]	2019	Reality	Activity recognition	Random Forest	Average accuracy 67%	Common navigation tasks: Self-positioning and orientation. Local environment target search Map target search Route memorization Walking to the destination
[26]	2020	Reality	Activity recognition	CNN	Average Precision: 40.41%	26 common action classes
[45]	2015	Tabletop	Intention prediction	Support Vector Machine (SVM)	88% success rate	Drag Maximize Minimize Scroll Free-form drawing

Table 1. *Cont.*

Reference	Year	Platform	Scope	Classifier	Performance	Tasks/Activities/Intentions
[39]	2014	PC	Intention recognition	Nearest Neighborhood (NN) Support Vector Machine (SVM)	Average accuracy: 79.81 ± 4.93 Average accuracy: 85.26 ± 0.70	Navigational intent Informational intent
[46]	2019	VR	Intention prediction	Long Short-Term Memory (LSTM) Topology	Accuracy 99.94% Precision 99.92% Recall 99.96% F_1-Score 99.94%	Navigation: Needing navigation aid No need for navigation aid
[37]	2021	VR	Intention prediction	Logistic Regression	Average PR-AUC = 0.12 Average ROC-AUC = 0.77	Issue a command or not
[27]	2020	VR	Activity recognition	Support Vector Machine (SVM) Logistic Regression Random Forest	Prediction accuracy: SVM: 80.23% Logistic Regression: 74.74% Random Forest: 79.50%	Shopping Goal-directed search Exploratory search

Alghofaili et al. [46] classify whether users need navigation assistance in VR environments through Long Short-Term Memory (LSTM) topology. It determines whether the user loses his/her way by analyzing the eye-movement behavior of the user in VR roaming scenarios. Pfeiffer et al. [27] classify the type of search (goal or exploration based) when shopping in cave-based VR. Their study also relies mainly on eye-movement data for training and evaluating three classifiers: SVM, LR, and RF, where SVM has the highest accuracy of 80.2%.

The most similar work to our study is the work of Brendan et al. [37]. Their study predicts whether a user will make a selection interaction or not in VR. In their study, a separate LR classifier is trained for each participant, but the overall results are not very satisfactory, with an average PR-AUC of 0.12. However, in their study, they also find that the classifiers for participants are very similar in terms of feature selection, which to the extent indicates that the interaction intention of the user is common in eye movement-based features. There is some commonality in the eye movement-based features. Therefore, the training dataset in our study is composed of eye-movement data and controller data generated by multiple users during the two interaction tasks of selection and telepoting. We want the trained models to determine whether the user wants to interact or not and the interaction type (selection or telepoting).

The superiority of our work over the existing works that aim to classify user interaction intention in VR is twofold. First, many studies are content-related, since they focus on highly specific application scenarios such as VR navigation [46] and shopping [27]. Our work can be applied in all areas that utilize basic interaction tasks such as selecting and teleporting. Application areas can range from simple scene-roaming to more complicated game interactions. Second, our recognition model is more accurate than some existing works [27,37], making it a better candidate for practical use.

3. Materials and Methods

3.1. Data Collecting

3.1.1. Participants

Ten participants (five female and five male) volunteered for this experiment. Their ages ranged between 22 and 27. All participants had normal or corrected-to-normal vision by using glasses or lenses during the experiment. Most participants were either undergraduate or graduate students. All participants had used VR Head Mounted Display (HMD) before. A pretest was conducted before the formal experiment to help the participants prepare.

3.1.2. Physical Setup

The virtual environment was displayed through an HTC VIVE Pro Eye integrated with an eye tracker. The screen had a 1440 × 1600 pixels/eye resolution with a 110° field of view. The HMD's highest refresh rate was 90 Hz. The refresh rate of the built-in eye tracker was 120 Hz, which offered tracking precision of 0.5–1.1°. The experiment was conducted on a PC with an Intel Core i7-9700 CPU, an NVIDIA GeForce GTX 1070 8G GPU, and 16G

DDR4 2666 Hz RAM. The experimental platform was developed using Unity 2019.4 and C#.

3.1.3. Experiment Design

We designed two basic VR interactive tasks for experiments. One used ray casting to select the target sphere (Figure 1). The other was teleporting to the target location (Figure 2).There were two reasons for choosing these two tasks: first, these two primary tasks are relatively simple, but they are very similar in interaction behavior; second, they are often used in actual VR applications. The most complex interaction in the current VR application scenario was the game. For example, in the game "Half-life: Alyx" released in 2020, selecting an item from a distance and teleporting are the basic interaction tasks. Other, more straightforward scenes, such as the Home scenario of SteamVR, also included these two tasks. They are also used as experimental tasks in many studies [37,47].

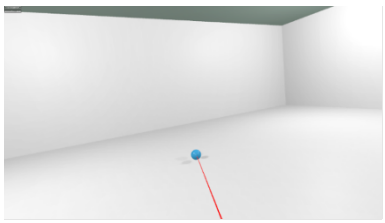

Figure 1. Using controllers to select the target sphere.

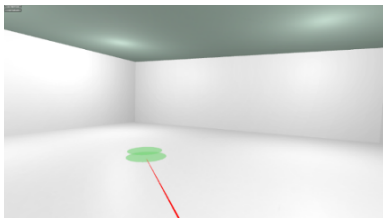

Figure 2. Using controllers to teleport to the target position.

The virtual environment was an empty room with the participant in the center. Participants were asked to repeat one of the two tasks 20 times in each session. The position of each target sphere or each target position was random. Each task was conducted in five sessions; that is, a total of 10 sessions for each participant.

3.1.4. Data Set

The raw data collected from the experiment consisted of gaze-related data, controller-related data, helmet-position coordinates, timestamps, and task types. Gaze-related data include the combined gaze-origin position, combined normalized gaze-direction vector, the corresponding timestamp and pupil diameter, and eye openness for either eye (Figure 3). In addition, we also acquired 3D gaze points in real-time with the help of a ray-based method [48]. The gaze direction vector and the corresponding gaze original position were used to find the intersection with the reconstructed 3D scene, representing the 3D gaze-points. The handle-related data were mainly the coordinates of the intersection points of the handle rays with the environment. One hundred tests were performed on ten subjects. After removing invalid data, 98 sets of valid data were obtained, i.e., a total of 250,380 raw data.

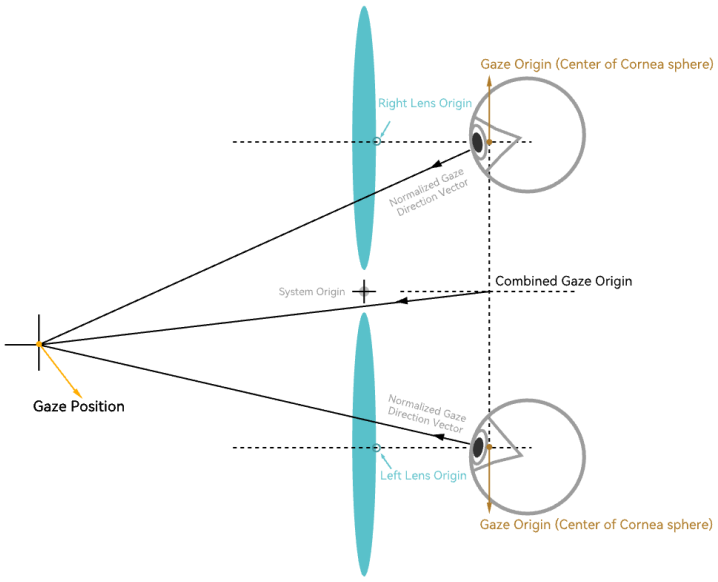

Figure 3. Eye tracker output data description.

One thing to note is that although the data collection frequency of the eye-tracking device was 120 Hz, our experimental platform was developed on Unity, so the actual data-collection frequency depended on the refresh frequency of the Update function. However, the increasing demand for GPU graphics rendering or the saturation of computing power led to a temporary decrease in the data collection frequency. The sampling frequency in this experiment fluctuates between 60 Hz and 40 Hz, with an average of 46 Hz. This will be taken into account in the subsequent feature extraction.

3.2. Proposed Method

3.2.1. Data Pre-Processing

Our processing pipeline is visualized in Figure 4. The first step filled the missing data mainly caused by blinking. The last valid data were directly filled in the blanks. There were 9552 blank data points, accounting for about 3.8%. The next step converted right-handed coordinates to left-handed. The eye-related data were obtained using the SDK (SRanpial) through a Unity script. According to the document of SRanpial, Gaze Original is the point in the eye from which the gaze ray originates, and Gaze Direction Normalized is the normalized gaze direction of the eye. They are both based on a right-handed coordinate system. However, Unity is based on a left-handed coordinate system. Therefore, we needed to multiply their X coordinates by -1 to convert the right-handed coordinate system to left-handed. Then, we transformed the Gaze Original vectors from the eye-in-head frame to the eye-in-world frame by adding the coordinates of the main camera to the Gaze Original vectors.

3.2.2. Ground Truth

We used the trigger/pad events from the hand controller to mark the ground truth of input datasets. It was uncertain how far in advance the intention could be predicted. We also needed to ensure sufficient training samples, so we chose two time thresholds to divide the data. The 20 or 40 sets of samples preceding a click were considered as positive samples; that is, the sampled data within 400 milliseconds as ground truth generation (GTG) type1 or 800 milliseconds as GTG type2 before the interaction occurred. In addition, we also tried to train two types of interaction-intention prediction models. One was a binary classifier, to predict whether users want to issue a command or not. The other was a three-class classifier which predicts whether users want to select, teleport, or execute no command at

all. Positive samples needed to be further divided into two types according to interaction tasks: selection or teleporting.

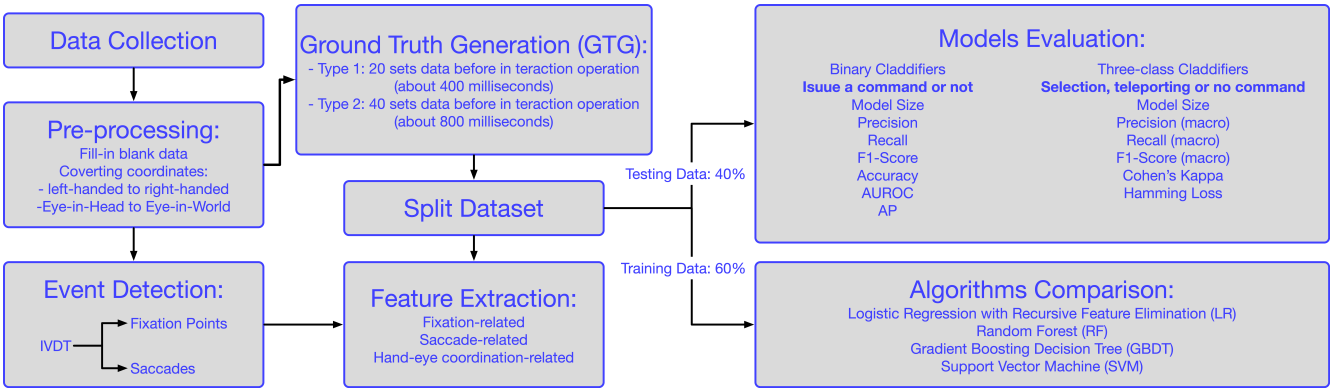

Figure 4. The pipeline to detect eye events, extract features, and train and evaluate models.

3.2.3. Eye Event Detection and Feature Extraction

Many previous studies selected eye-based features to capture spatiotemporal characteristics based on two fundamental eye movements—fixation points and saccades. Our method utilizes four types of features for interaction-intention prediction: fixation, saccade, pupil, and hand-eye coordination. We extracted them from each fixation and saccade. We summarize these features in Table 2. Therefore, eye event detection is required before feature extraction to classify these two types of eye movements.

Komogortsev and Karpov [49] proposed a ternary classification algorithm called velocity and dispersion threshold identification (I-VDT). We chose it to classify the two types of eye movements. It first identifies saccades by the velocity threshold. Subsequently, it identifies smooth pursuits from fixation by a modified dispersion threshold and duration. The original algorithm needs an initial time window to carry out. However, in a VR environment, due to increasing graphic rendering requirements or the limited computing power of GPUs, the data collection frequency is unstable and often reduced. Since the raw data is obtained using the SDK (SRanpial) through a Unity script, the data-collection frequency depends on the graphic engine's processing rate. To solve this problem, we adjusted the algorithm. Instead of setting an initial window, we checked whether it met the minimum fixation duration after determining a group of fixation points. In addition, we also checked the dispersion distance between the centroids of two adjacent fixation groups. They merged if they were too close (below the dispersion threshold). Moreover, the smooth pursuit was not one of our classification categories, so we modified the algorithm.

The I-VDT algorithm in this paper employs three velocity, dispersion, and minimum fixation-duration thresholds. The specific values of these three parameters are determined by previous research [50]. The velocity threshold is 140 degrees per second. The minimum fixation duration is 110 milliseconds. The maximum dispersion angle is 5.75 degrees. I-VDT begins by calculating point-to-point velocities for each eye-data sample. Then, I-VDT classifies (Algorithm A1) each point as a fixation or saccade point based on a simple velocity threshold: if the point's velocity is below the threshold, it is a fixation point; otherwise, it is a saccade point. Then, we check whether each fixation group meets the minimum fixation duration and whether the dispersion distance between adjacent fixation groups meets the maximum dispersion distance. If both are met, it is regarded as a fixation at centroid (x, y, z) of the fixation group points with the first point's timestamp as fixation start timestamp and the duration of the points as the fixation duration.

Each gaze sample should belong to fixation or saccade after classification by I-VDT. So, to represent all these features as a continuous-time series, we set the value for each gaze sample as the feature value from the most recent fixation or saccade event, i.e., each was carried forward in time until the next detected event. Pupil-related and hand-eye-coordination-related features were all calculated based on the fixation or scanning data

group to which the sample belonged. As for hand-eye coordination, related features were based on the distance between points of gaze and controller-ray intersection with the virtual environment at the same time. Specifically, let $G_t < x, y, z >$ be the positions of gaze in the virtual environment at time t during the execution of a particular task; let $C_t < x, y, z >$ represent the position of the intersection point of the controller ray with the virtual environment at time t. We argue that the distance between these points $D_t = |G_t - C_t|$ strongly correlates with whether the user executes interaction. Çığ, Ç and Sezgin [45] confirmed that the distance between strokes and gaze in pen-based touch-screen interaction is related to task types, and different task types have completely different rise/fall characteristics. We assume the same in VR controller interaction, so we choose this feature type. See Table 2 for specific features.

Table 2. Features derived from fixation, saccade, pupillary responses, and hand-eye coordination.

Types	Features
Fixation Related	Fixation detection: Sample-level boolean indicating whether a sample was part of a fixation or not Fixation duration Standard deviation of gaze position on x-axis, y-axis, and z-axis during fixation Skewness of gaze position on x-axis, y-axis, and z-axis during fixation Kurtosis of gaze position on x-axis, y-axis, and z-axis during fixation Average velocity of gaze samples during fixation Path length of gaze samples during fixation Dispersion of gaze samples during fixation
Saccade Related	Saccade duration Standard deviation of gaze position on x-axis, y-axis, and z-axis M3S2K of gaze velocity during saccade Saccadic ratio: peak velocity/saccade duration Saccade amplitude
Pupil Related	M3S2K of left-eye pupil during a fixation or a saccade M3S2K of right eye pupil during a fixation or a saccade
Hand-Eye-Coordination-related	M3S2K of the distance between gaze position and the hit point of the controller ray during a fixation or saccade

Note: M3S2K refers to the computation of mean, median, maximum, standard deviation, skewness, and Kurtosis values.

3.2.4. Metrics

We chose accuracy, precision, recall, F_1-Score, and model size to evaluate binary classifiers.

Accuracy is the ratio of correct predictions. If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value, then the ratio of correct predictions over $n_{samples}$ samples is defined as

$$\text{Accuracy}(y, \hat{y}) = \frac{\sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i)}{n_{samples}} \tag{1}$$

where $1(x)$ is an indicator function.

Precision is the ability of the classifier not to label negative samples as positive, and recall is the ability of the classifier to find all positive samples. The calculation formulas are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

where TP, FP, and FN are the numbers of true positives, false positives, and false negatives, respectively.

F_1-Score is the weighted harmonic mean of precision and recall with equal importance. The F_1-Score is defined as

$$F_1 = \frac{2 * (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{4}$$

In addition to the above metrics, for binary classification, we also use average precision (AP) and AUROC (the area under the receiver operating characteristic curve) to evaluate binary classifiers.

The value of AP is between 0 and 1 and higher is better. AP is defined as

$$\mathrm{AP} = \sum_n (R_n - R_{n-1}) P_n \tag{5}$$

where P_n and R_n are the precision and recall at the n-th threshold. With random predictions, the AP is the ratio of positive samples.

A receiver operating characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a binary classifier as its discrimination threshold varies. It is created by plotting the ratio of true positives to all positives (TPR = true positive rate) versus the ratio of false positives to all negatives (FPR = false positive rate), at various threshold settings. By computing the area under the ROC curve (AUROC), the curve information is summarized in one number. The closer to 1, the better.

As for three-class classifiers, we chose Hamming loss, Cohen's kappa, model size, and the macro average of precision, recall, and F_1-Score.

Let n_{labels} be the number of classes or labels, the Hamming loss $L_{Hamming}$ is defined as:

$$L_{Hamming}(y, \hat{y}) = \frac{\sum_{i=0}^{n_{labels}-1} 1(\hat{y}_i \neq y_i)}{n_{labels}} \tag{6}$$

The closer to zero, the better.

The calculation formulas of macro average metrics are as follows:

$$\mathrm{Precision}_{macro} = \frac{\sum_{l \in L} P(y_l, \hat{y}_l)}{|L|} \tag{7}$$

$$\mathrm{Recall}_{macro} = \frac{\sum_{l \in L} R(y_l, \hat{y}_l)}{|L|} \tag{8}$$

$$F_{1_{macro}} = \frac{\sum_{l \in L} F_1(y_L, \hat{y}_l)}{|L|} \tag{9}$$

where L is the set of labels, and $P(y_l, \hat{y}_l), R(y_l, \hat{y}_l), F_1(y_l, \hat{y}_l)$ are the Precision, Recall, F_1-Score of class or label l, respectively.

A kappa score is a number between -1 and 1. Scores above 0.8 are generally considered good agreement; zero or lower means no agreement (practically random labels).

3.2.5. Classifiers

We used the features described in the previous sections to build models that automatically classify observations as positive (interaction intention) or negative. There are plenty of candidate classification algorithms. We explored LR models, RF, GBDT, and SVM (which are commonly used for gaze data (Table 1)) to predict interaction intention in VR. All the above algorithms are implemented by Scikit-learn (https://github.com/scikit-learn/scikit-learn, accessed on 1 April 2022) [51], an open source machine-learning library in Python. We performed parameter tuning to find the optimal parameters for each classifier with F_1-Score. The optimal parameters for each classifier are given in Appendix B Table A1.

4. Results

All evaluations were performed using Scikit-learn. The evaluations were measured in line with the standard three-step machine-learning pipeline, where we first extracted features from the dataset and split the data into training and test datasets, then trained classifier models using training data, and finally measured all metrics using test data. We evaluated the hyper-parameters of each model using a grid search with two-fold cross-validation based on F_1-Score.

4.1. Performance of Binary Classifiers

Table 3 presents an overview of the main results of the best classification performance for each combination of algorithms and GTG methods for binary classification.

We compare the performance across all combinations of four classifiers, two GTG methods, and two feature sets. Table 3 shows the performance using LR, SVM, RF, and GBDT . The LR classifier performed poorly for both feature sets. As our dataset is highly complex and multi-dimensional, the LR classifier proved unsuitable for our purpose. The F_1-Scores of the other three classifiers are higher than 86%, which is worthy of further analysis.

We can see an improvement in the F1-Score when hand-eye-coordination-related features were used. The F1-Scores of the other three classifiers were improved by 1–3% by incorporating hand-eye-coordination-related features. Table 3 also shows that the GTG methods influenced the classifiers' performance for the Whole Feature Set. When using the Whole Feature Set, the GBDT classifier achieved a maximum *F*1-Score of 92.4% using 20 sets of data before interaction operation (400 milliseconds, GTG type1) as positive samples and 87.3% with 40 sets of data before interaction operation (800 milliseconds, GTG type2) as positive samples. However, the difference between the two GTG methods was less significant when using the Eye-Only Feature Set. One possible explanation can be related to the fact that hand-eye-coordination-related features are more sensitive to time. In other words, the relevant features have substantial differences only when they are very close to the time of interaction.

Table 3. Binary classification results for the combinations of four classifiers (RF, GBDT, LR, and SVM), two feature sets, and two GTG methods.

GTG		20 Sets Data before Interaction Operation (400 Milliseconds)				40 Sets Data before Interaction Operation (800 Milliseconds)			
Algorithm		**RF**	**GBDT**	**RFE + LR**	**SVM**	**RF**	**GBDT**	**RFE + LR**	**SVM**
Whole Feature Set	**Accuracy**	0.976	0.976	0.838	0.964	0.949	0.947	0.801	0.928
	Precision	0.954	0.947	0.476	0.894	0.962	0.947	0.679	0.908
	Recall	0.890	0.902	0.172	0.880	0.874	0.882	0.709	0.859
	F_1-Score	0.921	0.924	0.253	0.887	0.916	0.914	0.693	0.883
	AUROC	0.994	0.993	0.875	0.980	0.987	0.981	0.850	0.962
	AP	0.980	0.970	0.460	0.940	0.980	0.970	0.660	0.950
	Size	83 MB	54.5 MB	37KB	10.3 MB	139.2 MB	32.9 MB	19 KB	21.8 MB
Eye-Only Feature Set (No Hand-Eye Coordination-related Feature)	**Accuracy**	0.972	0.974	0.836	0.959	0.945	0.943	0.758	0.927
	Precision	0.958	0.949	0.465	0.881	0.966	0.947	0.653	0.899
	Recall	0.862	0.884	0.190	0.854	0.857	0.868	0.508	0.868
	F_1-Score	0.908	0.916	0.270	0.868	0.908	0.906	0.571	0.883
	AUROC	0.993	0.990	0.853	0.973	0.985	0.977	0.817	0.960
	AP	0.980	0.950	0.430	0.920	0.980	0.970	0.600	0.940
	Size	100.1 MB	66.1 MB	37KB	8.4 MB	156.4 MB	41.2 MB	39 KB	18.2 MB

In addition to standard evaluation metrics in machine learning, we also chose the model size as a reference because the ultimate goal of our research is to achieve real-time classification, so the smaller the model, the better. RF and GBDT had similar classification performances, but the GBDT model was relatively small. RF and GBDT are ensemble classifiers, which means the final models contain many decision trees. The SVM classifier only needed to record the final classification hyperplane so that the model was smaller than the other two.

Table 4 lists the top-ten features according to RF and GBDT importance scores when predicting whether users want to issue a command or not with the Eye-only Feature Set or Whole Feature Set.

Table 4. Top-ten importance features based on RF and GBDT feature importance scores of binary classifiers.

Algorithm		RF		GBDT	
GTG	**Feature Set**	**Features**	**Importance**	**Features**	**Importance**
20 sets data before interaction operation (400 milliseconds)	Whole Feature Set	[C] Min of distance	0.060	[C] Median of distance	0.120
		[F] Fixation duration	0.051	[S] Saccade duration	0.064
		[C] Median of distance	0.023	[C] Skewness of distance	0.055
		[C] Mean of distance	0.013	[F] Fixation duration	0.040
		[F] Average velocity of gaze samples during fixation	0.007	[S] Average velocity of gaze samples during saccades	0.040
		[C] Max of distance	0.006	[C] Max of distance	0.036
		[F] Dispersion of gaze samples during fixation	0.006	[C] Min of Distance	0.034
		[F] Fixation detection	0.005	[C] Standard deviation of distance	0.026
		[S] Average velocity of gaze samples during saccades	0.004	[C] Mean of distance	0.026
		[S] Max velocity of gaze samples During saccades	0.003	[F] Dispersion of gaze samples during fixation	0.024
	Eye-only Feature Set (No Hand-Eye Coordination-related Feature)	[F] Fixation duration	0.071	[F] Fixation duration	0.089
		[F] Average velocity of gaze samples during fixation	0.020	[F] Average velocity of gaze samples during fixation	0.081
		[F] Dispersion of gaze samples during fixation	0.020	[P] Kurtosis of right-eye pupil diameter	0.051
		[F] Fixation detection	0.011	[F] Path length of gaze samples during fixation	0.050
		[S] Max velocity of gaze samples during saccades	0.010	[F] Dispersion of gaze samples during fixation	0.037
		[S] Average velocity of gaze samples during saccades	0.009	[S] Average velocity of gaze samples during saccades	0.036
		[F] Path length of gaze samples during fixation	0.006	[P] Mean of left-eye pupil diameter	0.031
		[F] Standard deviation of z-axis coordinate of the gaze position during fixation	0.006	[P] Standard deviation of right-eye pupil diameter	0.028
		[S] Median velocity of gaze samples during saccades	0.006	[F] Standard deviation of z-axis coordinate of the gaze position during fixation	0.028
		[F] Standard deviation of x-axis coordinate of the gaze position during fixation	0.005	[P] Mean of right-eye pupil diameter	0.026
40 sets data before interaction operation (800 milliseconds)	Whole Feature Set	[C] Min of distance	0.043	[C] Median of distance	0.074
		[F] Fixation duration	0.026	[S] Saccade duration	0.064
		[F] Fixation detection	0.012	[C] Min of distance	0.046
		[C] Median of distance	0.010	[C] Standard deviation of distance	0.043
		[C] Mean of distance	0.010	[F] Kurtosis of y-axis coordinate of the gaze position during fixation	0.037
		[S] Average velocity of gaze samples during saccades	0.008	[F] Average velocity of gaze samples during fixation	0.034
		[S] Max velocity of gaze samples during saccades	0.007	[C] Mean of distance	0.031
		[C] Max of distance	0.007	[P] Kurtosis of right-eye pupil diameter	0.030

Table 4. *Cont.*

GTG	Algorithm / Feature Set	RF Features	RF Importance	GBDT Features	GBDT Importance
		[F] Dispersion of gaze samples during fixation	0.007	[F] Skewness of y-axis coordinate the gaze position during fixation	0.030
		[F] Average velocity of gaze samples during fixation	0.006	[S] Max Velocity of gaze samples during saccades	0.029
	Eye-Only Feature Set (No Hand-Eye Coordination-related Feature)	[F] Fixation Duration	0.064	[F] Fixation duration	0.136
		[S] Average velocity of gaze samples during saccades	0.017	[S] Standard deviation of y-axis coordinate of the gaze position during saccade	0.064
		[F] Fixation detection	0.013	[F] dispersion of gaze samples during fixation	0.055
		[S] Max velocity of gaze samples during saccades	0.013	[S] Min velocity of gaze samples during saccades	0.047
		[S] Median velocity of gaze samples during saccades	0.010	[P] Skewness of left-eye pupil diameter	0.038
		[S] Min velocity of gaze samples during saccades	0.009	[F] Skewness of x-axis coordinate of the gaze position during fixation	0.031
		[S] Saccade amplitude	0.009	[P] Mean of left-eye pupil diameter	0.028
		[F] Average velocity of gaze samples during fixation	0.009	[F] Average velocity of gaze samples during fixation	0.027
		[F] Dispersion of gaze samples during fixation	0.009	[S] Average velocity of gaze samples during saccades	0.027
		[S] Saccadic ratio	0.009	[S] Median velocity of gaze samples during saccades	0.023

Note: [F] stands for fixation-related feature; [S] stands for saccade-related feature; [P] stands for pupil-related feature; [C] stands for hand-eye-coordination-related feature.

For the Whole Feature Set, taking the example of the GBDT classifier with the highest *F*1-Score using GTG type1, the top-10 important features consisted of six hand-eye-coordination-related features, two fixation-related features, and one saccade-related feature. The top-10 features of other classifiers were highly consistent with this one. The four hand-eye-coordination-related features—min, max, median, and mean of distance—received high importance. As for eye-only features, three features about the velocity of gaze samples, such as the average velocity of gaze samples during fixation or saccade and the maximum velocity of gaze samples during saccade, also scored high in importance, the same as fixation-related features—fixation duration and dispersion of gaze samples during fixation.

For the Eye-Only Feature Set, taking the example of the GBDT classifier with the highest F_1-Score using GTG type1, the top-10 important features consisted of five fixation-related features, four pupil-related features, and one saccade-related feature. Overall, the important eye-only features were the same as the classifiers that used the Whole Feature Set.

4.2. Performance of Three-Class Classifiers

For three-class classifiers, except LR, the F1-Scores of the other three algorithms are above 0.9. The GBDT is still the best classification algorithm, followed by RF and SVM. Table 5 shows an overview of the main results for three-class classifiers.

Table 5. Three-class classification results for the combinations of four classifiers (RF, GBDT, LR, and SVM), two feature sets, and two GTG methods.

GTG		20 Sets Data before Interaction Operation (400 Milliseconds)				40 Sets Data before Interaction Operation (800 Milliseconds)			
Algorithm		**RF**	**GBDT**	**RFE + LR**	**SVM**	**RF**	**GBDT**	**RFE + LR**	**SVM**
Whole Feature Set	**Precision (macro)**	0.963	0.964	0.280	0.936	0.956	0.969	0.633	0.922
	Recall (macro)	0.916	0.923	0.333	0.930	0.893	0.939	0.557	0.909
	F_1-Score (macro)	0.939	0.943	0.305	0.933	0.921	0.953	0.582	0.915
	Cohen's kappa	0.906	0.912	0.000	0.866	0.879	0.927	0.395	0.830
	Hamming loss	0.026	0.024	0.159	0.036	0.056	0.034	0.263	0.072
	Size	107.5 MB	86.2 MB	4 KB	10.3 MB	180 MB	148.4 MB	5 KB	21.8 MB
Eye-Only Feature Set (No Hand-Eye Coordination-related Feature)	**Precision (macro)**	0.964	0.964	0.280	0.953	0.957	0.964	0.582	0.907
	Recall (macro)	0.899	0.907	0.333	0.898	0.885	0.915	0.476	0.895
	F_1-Score (macro)	0.929	0.934	0.304	0.923	0.917	0.938	0.492	0.901
	Cohen's kappa	0.891	0.899	0.000	0.846	0.871	0.902	0.274	0.801
	Hamming loss	0.029	0.027	0.159	0.039	0.059	0.046	0.290	0.085
	Size	118.9 MB	87.9 MB	3 KB	34.6 MB	175.6 MB	144.6 MB	4 KB	15.9 MB

In terms of GTG, for the GBDT algorithm, the two GTGs had little difference in classification performance, while for RF and SVM, the result of GTG type1 was better than that of type2. For the feature sets, as we estimated, the classification performance of the Eye-Only Feature Set was worse than the Whole Feature Set by 0.006–0.016 (F_1-Score). As for the model size, the GBDT had a better classification performance with a smaller model size than the RF. SVM was the smallest model, the same as binary classifiers.

Table 6 lists the top ten features of three-class classifiers using RF and GBDT. The features related to hand-eye coordination are still of high importance. However, some new features, especially those related to the y-axis distribution of fixation points, have a significant difference between the two interactive tasks of selection and blinking. However, it may also indicate that these indicators may be related to the design of the interactive interface.

Table 6. Top-ten importance features based on Random Forest and GBDT feature importance scores of three-class classifiers.

Algorithm		RF		GBDT	
GTG	**Feature Set**	**Features**	**Importance**	**Features**	**Importance**
20 sets data before interaction operation (400 milliseconds)	Whole feature set	[F] Fixation duration	0.062	[F] Fixation duration	0.078
		[C] Min of distance	0.058	[C] Min of distance	0.058
		[C] Median of distance	0.046	[C] Median of distance	0.054
		[C] Mean of distance	0.043	[F] Standard deviation of y-axis coordinate of the gaze position during fixation	0.051
		[F] Dispersion of gaze samples during fixation	0.042	[F] Dispersion of gaze samples during fixation	0.048
		[F] Standard deviation of y-axis coordinate of the gaze position during fixation	0.041	[C] Mean of distance	0.045
		[F] Path length of gaze samples during fixation	0.037	[F] Kurtosis of y-axis coordinate of the gaze position during fixation	0.032
		[F] Average velocity of gaze samples during fixation	0.032	[F] Average velocity of gaze samples during fixation	0.031
		[F] Kurtosis of y-axis coordinate of the gaze position during fixation	0.027	[F] path length of gaze samples during fixation	0.027
		[F] Skewness of y-axis coordinate of the gaze position during fixation	0.024	[F] Skewness of y-axis coordinate of the gaze position during fixation	0.026
	Eye-Only Feature Set (No Hand-Eye Coordination-related Feature)	[F] Fixation duration	0.084	[F] Fixation duration	0.082
		[F] Dispersion of gaze samples during fixation	0.058	[F] Path length of gaze samples during fixation	0.062
		[F] Standard deviation of y-axis coordinate of the gaze position during fixation	0.054	[F] Standard deviation of y-axis coordinate of the gaze position during fixation	0.053
		[F] Path length of gaze samples during fixation	0.046	[F] Dispersion of gaze samples during fixation	0.045
		[F] Average velocity of gaze samples during fixation	0.039	[F] Average velocity of gaze samples during fixation	0.043
		[F] Skewness of y-axis coordinate of the gaze position during fixation	0.039	[F] Skewness of y-axis coordinate of the gaze position during fixation	0.037
		[F] Kurtosis of y-axis coordinate of the gaze position during fixation	0.036	[F] Kurtosis of y-axis coordinate of the gaze position during fixation	0.033
		[S] Average velocity of gaze samples during saccades	0.030	[S] Max Velocity of Gaze samples during saccades	0.030
		[S] Max velocity of gaze samples during saccades	0.028	[S] Average velocity of gaze samples during saccades	0.029
		[F] Standard deviation of z-axis coordinate of the gaze position during fixation	0.025	[F] Standard deviation of z-axis coordinate of the gaze position during fixation	0.026
		[C] Min of distance	0.067	[C] Min of distance	0.173
		[F] Fixation duration	0.054	[F] Standard deviation of y-axis coordinate of the gaze position during fixation	0.116
		[F] Standard deviation of y-axis coordinate of the gaze position during fixation	0.048	[F] Fixation duration	0.086

Table 6. *Cont.*

Algorithm		RF		GBDT	
GTG	**Feature Set**	**Features**	**Importance**	**Features**	**Importance**
40 sets data before interaction operation (800 milliseconds)	Whole Feature Set	[C] Median of distance	0.043	[C] Max of distance	0.027
		[F] Dispersion of gaze samples during fixation	0.043	[C] Mean of distance	0.027
		[C] Mean of distance	0.041	[S] Average velocity of gaze samples during saccades	0.027
		[F] Path Length of gaze samples during fixation	0.036	[S] Max velocity of gaze samples during saccades	0.025
		[F] Kurtosis of y-axis coordinate of the gaze position during fixation	0.028	[F] Fixation detection	0.025
		[F] Average velocity of gaze samples during fixation	0.028	[P] Max of right-eye pupil diameter	0.023
		[C] Max of distance	0.026	[S] saccadic ratio	0.022
	Eye-Only Feature Set (No Hand-Eye Coordination-related Feature)	[F] Fixation duration	0.075	[F] Fixation duration	0.090
		[F] Standard deviation of y-axis coordinate of the gaze position during fixation	0.055	[F] Standard deviation of y-axis coordinate of the gaze position during fixation	0.047
		[F] Dispersion of gaze samples during fixation	0.044	[F] Skewness of y-axis coordinate of the gaze position during fixation	0.043
		[F] Path length of gaze samples during fixation	0.039	[F] Dispersion of gaze samples during fixation	0.042
		[S] Average velocity of gaze samples during saccades	0.035	[F] Path length of gaze samples during fixation	0.042
		[S] Max velocity of gaze samples during saccades	0.033	[S] Max velocity of gaze samples during saccades	0.038
		[F] Average velocity of gaze samples during fixation	0.032	[F] Kurtosis of y-axis coordinate of the gaze position during fixation	0.037
		[F] Kurtosis of y-axis coordinate of the gaze Position during fixation	0.030	[F] Average velocity of gaze samples during fixation	0.033
		[F] Skewness of y-axis coordinate of the gaze position during fixation	0.030	[S] Average velocity of gaze samples during saccades	0.032
		[S] Median velocity of gaze samples during saccades	0.027	[S] Median velocity of gaze samples during saccades	0.029

Note: [F] stands for fixation-related feature, [S] stands for saccade-related feature, [P] stands for pupil-related feature, [C] stands for Hand-Eye-Coordination-Related feature.

5. Discussion

The research of binary classifiers mainly explores which features can separate intentional behavior from unintentional behavior. The research of classifiers is to explore which features may be particularly relevant to the two tasks in our experiment. It can be said that binary classifiers can play a comparative role to three-class classifiers. In general, the features in binary classifiers are independent of the coordinate axis. The y-axis—that is, the vertical gaze coordinate distribution in three-class classification—plays a vital role in distinguishing the two types of tasks. It should be noted that when we select features at the beginning, we avoid features related to absolute coordinates and retain features related to the distribution law of coordinates. The above phenomenon may be because the selection task requires the user to keep staring at the target until visual feedback indicates that the interaction is completed. However, the teleporting task only requires clarification of the destination, so there is no need to keep staring at destination but to prepare for the change of perspective after teleporting. This phenomenon needs to be further explored in later research.

In the selection of features, we used two feature sets. The major difference was whether to include the hand-eye-coordination-related features. On the one hand, we wanted to verify whether the features of hand-eye coordination can improve the accuracy of interaction intention recognition in a multimodal interaction system, including controller and eye movement. The results show that the hand-eye-coordination index is important in predicting interaction intention. On the other hand, we should also consider whether the interaction intention of users can be effectively predicted with only eye-movement data and without controller-related data. Our study shows that only the features related to eye movement can be used to classify the interaction intention and the classification performance is also acceptable.

We used two kinds of methods to generate datasets. The main difference was how many groups of sampled data were included before the interaction occurred. We expected the system to deduce the interaction intention in advance. We selected 400 milliseconds and 800 milliseconds for comparative analysis. The classification result of the 800-millisecond classifier was slightly inferior to that of the 400-millisecond classifier, which is understandable. The generation time of real interaction intention was short, especially for our experiment's simple interaction tasks. If a long period is selected for data generation, the difference of features under different categories will not be significant, and the classification performance will naturally decline. However, it is not always good to use a shorter period. The shorter the period is, the fewer data we can generate in the dataset. In that way, the robustness of the trained model may decline. The choice of this time length needs to be determined through further experimental research and combined with the user's expectation of the intention prediction system.

As for the selection of algorithms, GBDT had the best performance. Its classification performance was not inferior to RF, and its model size was smaller than RF's. When we transformed the model into a real-time classifier, it was more likely to reduce latency. The model size of the SVM was small enough, but the overall classification performance still lagged behind the other two algorithms. In addition, SVM is more dependent on hyperparameters and takes the longest time to train.

We declare several limitations of our work, despite our best efforts to minimize them. First, the dataset is not entirely naturalistic. The number of participants was limited, so it was necessary to use data from new participants to verify the performance of the models. The experimental environment was also relatively simple. Whether more complex interaction scenarios will impact the classification performance still needs to be verified by follow-up research.

In the light of promising findings reported in this paper, we envision several immediate follow-ups to our work, as well as long-term research directions to explore. An immediate extension might involve conducting experiments to see if our classification models apply to other more complex interaction environments rather than a concise experiment

environment only. We want to explore two factors. One is whether the targets of different dimensions will affect the prediction results of the classifier (the selection target in this experiment is a sphere if it is replaced by a plane). The other is whether the interface complexity will affect the prediction results of the classifier (if there are multiple targets or locations in the environment at the same time). We also want to build an online prediction system to verify the performance of classifiers. Further experiments would evaluate the usability aspects of this setup and compare it to state-of-the-art online interaction intention prediction mechanisms in the literature. Another possible direction might involve conducting experiments to see if our prediction system can successfully recognize other interaction tasks.

6. Conclusions

This paper explored hand-eye-coordination-related features to improve interaction intention recognition in a VR environment. We collected a dataset of eye-movement data and controller-related data from 10 participants as they performed two basic interaction tasks: selection and teleporting. We extracted a Whole Feature Set, including fixation-related, saccade-related, pupil-related, and hand-eye-coordination-related features, and an Eye-Only Feature Set without hand-eye-coordination-related features. We obtained a high binary classification performance score (F_1-Score = 0.924) using the combination of the Whole Feature Set, GTG method type1, and the GBDT classifier, as well as a high three-class classification performance score (F_1-Score = 0.953) using the combination of the Whole Feature Set, GTG method type2, and the GBDT classifier. The results show that hand-eye-coordination-related features improve interaction intention recognition in VR environments. The GBDT had the best classification performance among the four classifiers, and its model size was smaller than the RF's. Generally, this work provides the groundwork for its exploration and towards building a robust and generalizable model for eye-based interaction-intention recognition in VR. We believe that predicting the interaction intention will eventually enable us to build systems that save users the trouble of switching during basic interaction tasks.

Author Contributions: Conceptualization, X.-L.C. and W.-J.H.; methodology, X.-L.C. and W.-J.H.; software, X.-L.C.; validation, X.-L.C. and W.-J.H.; formal analysis, X.-L.C.; data curation, X.-L.C.; writing—original draft preparation, X.-l.C.; writing—review and editing, W.-J.H.; supervision, W.-J.H.; funding acquisition, X.-L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by BUPT Excellent Ph.D. Students Foundation (Grant No.: CX2019112). This work was also funded by Beijing University of Posts and Telecommunications-China Mobile Research Institute Joint Center (Research Title: Research on 6G new business models and business communication quality indicators).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://www.scidb.cn/s/nQN7fm, accessed on 1 April 2022.

Acknowledgments: Thanks to all participants.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

VR	Virtal Reality
HMD	Head Mounted Display
LR	Logistic Regression
SVM	Support Vector Machine
RF	Random Forest
GBDT	Gradient Boosting Decision Tree
GTG	Ground Truth Generation

Appendix A

The I-VDT algorithm in this paper employs three thresholds of velocity, dispersion, and minimum fixation duration. The specific values of these three parameters are determined by previous research. The velocity threshold is 140 degrees per second. The minimum fixation during is 110 milliseconds. The maximum dispersion angle is 5.75 degrees. See Algorithm A1 below for details.

Algorithm A1 Velocity and Dispersion-Threshold Identification

Require: p_i:3D gaze position with timestamps, (x, y, z, t); V_i:normalized gaze direction vector with timestamps, Vel:velocity threshold; DD_{max}: maximum fixation dispersion distance threshold; $Duration_{min}$: minimum fixation duration threshold;

Ensure: f_i:representative coordinates corresponding to fixations groups, and the the starting time and duration of these fixations groups, $(x_f, y_f, z_f, t_{start}, d)$

// calculate the instantaneous visual angle

for $i = 0 \rightarrow n - 1$ **do**

$v_i = \frac{\arccos \frac{V_i \cdot V_{i+1}}{\|V_i\| \|V_{i+1}\|}}{|t_{i+1} - t_i|} \times 5.73 \times 10^4$

end for

Initialize Previous fixation group PFG and current fixation group CFG

save p_0 into PFG

save p_1 into CFG

for $i = 2 \rightarrow n - 1$ **do**

  Calculate the CFG centroid coordinates (x, y, z)

  Calculate the dispersion distance (DD) between CFG centroid coordinates and p_i coordinates

  if $v_i < Vel$ **then**

    save p_i into CFG

  else

    if CFG is not empty **then**

      Calculate the duration d of the points in CFG

      if $d > Duration_{min}$ **then**

        Calculate the dispersion distance (DD) between the first point in CFG and the last point in PFG

        if $DD < DD_{max}$ **then**

          Merge CFG into PFG

        else

          Calculate the PFG centroid coordinates (x_f, y_f, z_f)

          Save the timestamp t of the first point in PFG as t_{start}

          Calculate the duration d of points in PFG

          Initialize PFG

          Merge CFG into PFG

          Initialize CFG

          save p_i into CFG

        end if

      else

        Initialize CFG

        save p_i into CFG

      end if

    end if

  end if

end for

Appendix B

Table A1 shows the optimal parameters for each classifier discussed in this paper.

Table A1. The optimal parameters of each classifier are selected by grid search.

	Ground Truth Generation	Algrithm	Random Forest	Gradient Boosting Decision Tree	Logistic Regression with Recursive Feature Elimination	Support Vector Machine
2-class	20 sets data before interaction operation (400 milliseconds)	Whole Feature Set	max_depth: 29 max_features: 0.1 min_samples_leaf: 1 min_samples_split: 2 n_estimators: 100 criterion: entropy	max_depth: 24 max_features: 0.1 min_samples_leaf: 2 min_samples_split: 7 n_estimators: 100 learning_rate: 1.0	Optimal number of features: 38	C: 100.0 gamma: 0.1 kenel:RBF
		Eye-Only Feature Set (No Hand-Eye Coordination-related Feature)	max_depth: 30 max_features: 0.1 min_samples_leaf: 1 min_samples_split: 2 n_estimators: 100 criterion: entropy	max_depth: 25 max_features: 0.1 min_samples_leaf: 3 min_samples_split: 2 n_estimators: 100 learning_rate: 1.0	Optimal number of features: 39	C: 100.0 gamma: 0.1 kenel:RBF
	40 sets data before interaction operation (800 milliseconds)	Whole Feature Set	max_depth: 35 max_features: 0.1 min_samples_leaf: 1 min_samples_split: 2 n_estimators: 100 criterion: entropy	max_depth: 17 max_features: 1 min_samples_leaf: 19 min_samples_split: 8 n_estimators: 100 learning_rate: 1.0	Optimal number of features: 16	C: 10.0 gamma: 0.1 kenel:RBF
		Eye-Only Feature Set (No Hand-Eye Coordination-related Feature)	max_depth: 30 max_features: 0.1 min_samples_leaf: 1 min_samples_split: 2 n_estimators: 100 criterion: entropy	max_depth: 17 max_features: 1.0 min_samples_leaf: 14 min_samples_split: 2 n_estimators: 100 learning_rate 1.0	Optimal number of features: 41	C: 100.0 gamma: 0.1 kenel:RBF
3-class	20 sets data before interaction operation (400 milliseconds)	Whole Feature Set	max_depth: 27 max_features: 0.1 min_samples_leaf: 1 min_samples_split: 2 n_estimators: 100 criterion: gini	max_depth: 22 max_features: 0.1 min_samples_leaf: 12 min_samples_split: 2 n_estimators: 100 learning_rate: 0.1	Optimal number of features: 1	C: 100.0 gamma: 0.1 kenel:RBF

Table A1. *Cont.*

Ground Truth Generation	Algrithm	Random Forest	Gradient Boosting Decision Tree	Logistic Regression with Recursive Feature Elimination	Support Vector Machine
40 sets data before interaction operation (800 milliseconds)	Eye-Only Feature Set (No Hand-Eye Coordination-related Feature)	max_depth: 29 max_features: 0.1 min_samples_leaf: 1 min_samples_split: 2 n_estimators: 99 criterion: entropy	max_depth: 12 max_features: 0.1 min_samples_leaf: 1 min_samples_split: 3 n_estimators: 100 learning_rate: 0.1	Optimal number of features: 1	C: 10.0 gamma: 1.0 kenel:RBF
	Whole Feature Set	max_depth: 29 max_features: 0.1 min_samples_leaf: 1 min_samples_split: 2 n_estimators: 98 criterion: gini	max_depth: 18 max_features: 0.9 min_samples_leaf: 12 min_samples_split: 9 n_estimators: 99 learning_rate 0.3	Optimal number of features: 48	C: 10.0 gamma: 0.1 kenel:RBF
	Eye-Only Feature Set (No Hand-Eye Coordination-related Feature)	max_depth: 29 max_features: 0.1 min_samples_leaf: 1 min_samples_split: 2 n_estimators: 88 criterion: gini	max_depth: 17 max_features: 0.1 min_samples_leaf: 14 min_samples_split: 2 n_estimators: 100 learning_rate: 0.1	Optimal number of features: 40	C: 10.0 gamma: 0.1 kenel:RBF

References

1. Biocca, F.; Delaney, B. Immersive Virtual Reality Technology. In *Communication in the Age of Virtual Reality*; L. Erlbaum Associates Inc.: Mahwah, NJ, USA, 1995; pp. 57–124.
2. Burdea, G.C.; Coiffet, P. *Virtual Reality Technology*; John Wiley & Sons: Hoboken, NJ, USA, 2003.
3. Duchowski, A.T. A breadth-first survey of eye-tracking applications. *Behav. Res. Methods Instrum. Comput.* **2002**, *34*, 455–470. [CrossRef] [PubMed]
4. Duchowski, T. Gaze-based interaction: A 30 year retrospective. *Comput. Graph.* **2018**, *73*, 59–69. [CrossRef]
5. Jacob, R., Eye Tracking in Advanced Interface Design. In *Virtual Environments and Advanced Interface Design*; Oxford University Press, Inc.: Oxford, MS, USA, 1995; pp. 258–288.
6. Hansen, J.; Rajanna, V.; MacKenzie, I.; Bækgaard, P. A Fitts' Law Study of Click and Dwell Interaction by Gaze, Head and Mouse with a Head-Mounted Display. In Proceedings of the Workshop on Communication by Gaze Interaction (COGAIN '18), Warsaw, Poland, 14–17 June 2018; Association for Computing Machinery: New York, NY, USA, 2018. [CrossRef]
7. Blattgerste, J.; Renner, P.; Pfeiffer, T. Advantages of Eye-Gaze over Head-Gaze-Based Selection in Virtual and Augmented Reality under Varying Field of Views. In *Proceedings of the Symposium on Communication by Gaze Interaction*; ACM: New York, NY, USA, 2018.
8. Rajanna, V.; Hansen, J. Gaze Typing in Virtual Reality: Impact of Keyboard Design, Selection Method, and Motion. In Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications (ETRA '18), Warsaw, Poland, 14–17 June 2018; Association for Computing Machinery: New York, NY, USA, 2018. [CrossRef]
9. Pai, Y.; Dingler, T.; Kunze, K. Assessing hands-free interactions for VR using eye gaze and electromyography. *Virtual Real.* **2019**, *23*, 119–131. [CrossRef]
10. Piumsomboon, T.; Lee, G.; Lindeman, R.; Billinghurst, M. Exploring natural eye-gaze-based interaction for immersive virtual reality. In Proceedings of the 2017 IEEE Symposium on 3D User Interfaces (3DUI), Los Angeles, CA, USA, 18–19 March 2017; pp. 36–39. [CrossRef]
11. Qian, Y.; Teather, R. The Eyes Don't Have It: An Empirical Comparison of Head-Based and Eye-Based Selection in Virtual Reality. In Proceedings of the 5th Symposium on Spatial User Interaction (SUI '17), Brighton, UK, 16–17 October 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 91–98. [CrossRef]
12. Kytö, M.; Ens, B.; Piumsomboon, T.; Lee, G.; Billinghurst, M., Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, QC, Canada, 21–27 April 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1–14.
13. Luro, F.; Sundstedt, V. A Comparative Study of Eye Tracking and Hand Controller for Aiming Tasks in Virtual Reality. In Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications (ETRA '19), Denver, CO, USA, 25–28 June 2019; Association for Computing Machinery: New York, NY, USA, 2019. [CrossRef]
14. Scott, N.; Zhang, R.; Le, D.; Moyle, B. A review of eye-tracking research in tourism. *Curr. Issues Tour.* **2019**, *22*, 1244–1261. [CrossRef]
15. Kim, S.J.; Laine, T.H.; Suk, H.J. Presence Effects in Virtual Reality Based on User Characteristics: Attention, Enjoyment, and Memory. *Electronics* **2021**, *10*, 1051. [CrossRef]
16. Wolfe, J.M.; Horowitz, T.S. Five factors that guide attention in visual search. *Nat. Hum. Behav.* **2017**, *1*, 0058. [CrossRef]
17. Wolfe, J.M. Guided Search 6.0: An updated model of visual search. *Psychon. Bull. Rev.* **2021**, *28*, 1060–1092. [CrossRef]
18. McNally, R.J. Attentional bias for threat: Crisis or opportunity? *Clin. Psychol. Rev.* **2019**, *69*, 4–13. [CrossRef]
19. Anobile, G.; Arrighi, R.; Castaldi, E.; Burr, D.C. A Sensorimotor Numerosity System. *Trends Cogn. Sci.* **2021**, *25*, 24–36. [CrossRef]
20. Liu, X.; Chen, T.; Xie, G.; Liu, G. Contact-Free Cognitive Load Recognition Based on Eye Movement. *J. Electr. Comput. Eng.* **2016**, *2016*, 1–8. [CrossRef]
21. Kamińska, D.; Smółka, K.; Zwoliński, G. Detection of Mental Stress through EEG Signal in Virtual Reality Environment. *Electronics* **2021**, *10*, 2840. [CrossRef]
22. Al-Moteri, M.O.; Symmons, M.; Plummer, V.; Cooper, S. Eye tracking to investigate cue processing in medical decision-making: A scoping review. *Comput. Hum. Behav.* **2017**, *66*, 52–66. [CrossRef]
23. Brunyé, T.T.; Gardony, A.L. Eye tracking measures of uncertainty during perceptual decision making. *Int. J. Psychophysiol.* **2017**, *120*, 60–68. [CrossRef] [PubMed]
24. Srivastava, N.; Newn, J.; Velloso, E. Combining Low and Mid-Level Gaze Features for Desktop Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 1–27. [CrossRef]
25. Liao, H.; Dong, W.; Huang, H.; Gartner, G.; Liu, H. Inferring user tasks in pedestrian navigation from eye movement data in real-world environments. *Int. J. Geogr. Inf. Sci.* **2019**, *33*, 739–763. [CrossRef]
26. Xu, B.; Li, J.; Wong, Y.; Zhao, Q.; Kankanhalli, M.S. Interact as You Intend: Intention-Driven Human-Object Interaction Detection. *IEEE Trans. Multimed.* **2020**, *22*, 1423–1432. [CrossRef]
27. Pfeiffer, J.; Pfeiffer, T.; Meißner, M.; Weiß, E. Eye-Tracking-Based Classification of Information Search Behavior Using Machine Learning: Evidence from Experiments in Physical Shops and Virtual Reality Shopping Environments. *Inf. Syst. Res.* **2020**, *31*, 675–691. [CrossRef]
28. Iqbal, S.T.; Bailey, B.P. Using Eye Gaze Patterns to Identify User Tasks. *He Grace Hopper Celebr. Women Comput.* **2004**, *6*, 2004.

29. Courtemanche, F.; Aïmeur, E.; Dufresne, A.; Najjar, M.; Mpondo, F. Activity recognition using eye-gaze movements and traditional interactions. *Interact. Comput.* **2011**, *23*, 202–213. [CrossRef]
30. Steichen, B.; Carenini, G.; Conati, C. User-adaptive information visualization: Using eye gaze data to infer visualization tasks and user cognitive abilities. In Proceedings of the 2013 international conference on Intelligent user interfaces-IUI '13, Santa Monica, CA, USA, 19–22 March 2013; ACM Press: New York, NY, USA, 2013. [CrossRef]
31. Yang, J.J.; Gang, G.W.; Kim, T.S. Development of EOG-Based Human Computer Interface (HCI) System Using Piecewise Linear Approximation (PLA) and Support Vector Regression (SVR). *Electronics* **2018**, *7*, 38. [CrossRef]
32. Paing, M.P.; Juhong, A.; Pintavirooj, C. Design and Development of an Assistive System Based on Eye Tracking. *Electronics* **2022**, *11*, 535. [CrossRef]
33. Bulling, A.; Ward, J.A.; Gellersen, H.; Tröster, G. Eye movement analysis for activity recognition. In Proceedings of the 11th International Conference on Ubiquitous Computing, Orlando, FL, USA, 30 September–3 October 2009; ACM: New York, NY, USA, 2009. [CrossRef]
34. Bulling, A.; Roggen, D.; Tröster, G. What's in the Eyes for Context-Awareness? *IEEE Pervasive Comput.* **2011**, *10*, 48–57. [CrossRef]
35. Ogaki, K.; Kitani, K.M.; Sugano, Y.; Sato, Y. Coupling eye-motion and ego-motion features for first-person activity recognition. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012. [CrossRef]
36. Bednarik, R.; Eivazi, S.; Vrzakova, H. A Computational Approach for Prediction of Problem-Solving Behavior Using Support Vector Machines and Eye-Tracking Data. In *Eye Gaze in Intelligent User Interfaces*; Springer: London, UK, 2013; pp. 111–134. [CrossRef]
37. Brendan, D.; Peacock, C.; Zhang, T.; Murdison, T.S.; Benko, H.; Jonker, T.R. Towards Gaze-Based Prediction of the Intent to Interact in Virtual Reality. In Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA '21 Short Papers), Stuttgart, Germany, 25–29 May 2021; Association for Computing Machinery: New York, NY, USA, 2021. [CrossRef]
38. Van-Horenbeke, F.A.; Peer, A. Activity, Plan, and Goal Recognition: A Review. *Front. Robot. AI* **2021**, *8*, 106. [CrossRef]
39. Jang, Y.M.; Mallipeddi, R.; Lee, S.; Kwak, H.W.; Lee, M. Human intention recognition based on eyeball movement pattern and pupil size variation. *Neurocomputing* **2014**, *128*, 421–432. [CrossRef]
40. Jang, Y.M.; Mallipeddi, R.; Lee, M. Identification of human implicit visual search intention based on eye movement and pupillary analysis. *User Model. User-Adapt. Interact.* **2014**, *24*, 315–344. [CrossRef]
41. Lisha, M.A.; Jian, L.V.; Pan, W.; Shan, J.; Ping, Z. Research on Implicit Intention Recognition and Classification Based on Eye Movement Pattern. *J. Graph.* **2017**, *38*, 332.
42. Bednarik, R.; Vrzakova, H.; Hradis, M. What do you want to do next: A novel approach for intent prediction in gaze-based interaction. In Proceedings of the Symposium on Eye Tracking Research and Applications, Santa Barbara, CA, USA, 28–30 March 2012; ACM Press: New York, NY, USA, 2012. [CrossRef]
43. Liang, Y.; Wang, W.; Qu, J.; Yang, J. Application of Eye Tracking in Intelligent User Interface. In Proceedings of the 2018 3rd International Conference on Communication, Image and Signal Processing, Sanya, China, 16–18 November 2018; pp. 333–340.
44. Huang, C.; Andrist, S.; Sauppé, A.; Mutlu, B. Using gaze patterns to predict task intent in collaboration. *Front. Psychol.* **2015**, *6*, 1049. [CrossRef]
45. Çığ, Ç.; Sezgin, T.M. Gaze-based prediction of pen-based virtual interaction tasks. *Int. J. Hum.-Comput. Stud.* **2015**, *73*, 91–106. [CrossRef]
46. Alghofaili, R.; Sawahata, Y.; Huang, H.; Wang, H.; Shiratori, T.; Yu, L. Lost in Style: Gaze-Driven Adaptive Aid for VR Navigation. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1–12.
47. Zagata, K.; Gulij, J.; Halik, Ł.; Medyńska-Gulij, B. Mini-Map for Gamers Who Walk and Teleport in a Virtual Stronghold. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 96. [CrossRef]
48. Mansouryar, M.; Steil, J.; Sugano, Y.; Bulling, A. 3D Gaze Estimation from 2D Pupil Positions on Monocular Head-Mounted Eye Trackers. In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16), Charleston, SC, USA, 14–17 March 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 197–200. [CrossRef]
49. Komogortsev, O.V.; Karpov, A. Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behav. Res. Methods* **2013**, *45*, 203–215. [CrossRef]
50. Chen, X.; Hou, W. Identifying Fixation and Saccades in Virtual Reality. *arXiv* **2002**, arXiv:2205.04121.
51. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.