electronics

Article

Preventing Keratoconus through Eye Rubbing Activity Detection: A Machine Learning Approach

George Nokas and Theodore Kotsilieris *

Department of Business and Organizations Administration, University of the Peloponnese, GR 241-00 Kalamata, Greece
* Correspondence: t.kotsilieris@uop.gr

Abstract: Keratoconus is a non-inflammatory disease of the eyes diagnosed in more than 1/2000 people, making it significantly common. Among others, eye rubbing has been identified as a risk factor for the development of keratoconus. The severity of the disease strongly depends on the frequency and force of eye rubbing. Vast research efforts have focused on diagnosing keratoconus through the application of artificial intelligence techniques over optical coherence tomography images and corneal measurements. However, to the best of the authors' knowledge, no studies have been conducted which provide an eye rubbing detection and alert mechanism for keratoconus prevention. This study intends to help close this research gap. An inertial measurement unit that is dedicated to collecting hand motion data and machine learning techniques are jointly employed for the early detection of potential problems and complications. Four conventional classification methods (support vector machines, decision trees, random forest, and XGBoost) were evaluated and compared. All methods attain high-quality accuracy results, with SVMs, RF, and XGBoost slightly outperforming DTs. As the results reveal, the performance of all methods is remarkable, allowing the integration of such a solution in wearable devices such as smartwatches to be considered for the early detection of eye rubbing and keratoconus prevention.

Keywords: keratoconus; eye rubbing detection; support vector machines; decision trees

Citation: Nokas, G.; Kotsilieris, T. Preventing Keratoconus through Eye Rubbing Activity Detection: A Machine Learning Approach. *Electronics* **2023**, *12*, 1028. https://doi.org/10.3390/electronics12041028

Academic Editor: Hyunjin Park

Received: 14 December 2022
Revised: 15 February 2023
Accepted: 16 February 2023
Published: 18 February 2023

1. Introduction

Keratoconus (KC) is a cornea disease associated with its gradual thinning that makes it turn into a cone. It is considered to be rare, although several studies suggest that higher prevalence bias may be identified by different sample sizes, populations, or countries of origin. A recent meta-analysis study reported that the incidents are almost 0.138% of the total population [1]. However, most studies found that KC is more frequently diagnosed than 1/2000, making it a significantly common disease [2]. Comorbidities, histopathology, proteomics, biomechanics, genetics, and environmental factors are some of the dimensions of the disease considered in the literature. The disease treatment varies according to its severity. Mild cases may be treated with special purpose contact lenses that are hard to identify. Another procedure that fits to more severe KC cases is keratoplasty, which is a surgical procedure for cornea transplantation [2].

Its pathophysiology remains unknown, although it was first described in the 19th century. Among others, eye rubbing is assumed to be a contributing factor affecting the etiology of the disease. It is a common activity as a response to stress, itch, fatigue, or other eye disorders (such as dryness or allergies). No matter the origin of eye rubbing, its role in keratoconus prevalence is identified by several studies [3–6]. The data reveal that the occurrence of eye rubbing in KC patients varied from 45% to 83%.

The main contribution of this study is to propose and evaluate the accuracy of a machine-learning-based system for the detection and prevention of eye rubbing. The proposed system uses accelerometer and gyroscope data that are collected through wrist-mounted sensors. Sequential overlapping windows are applied over these data in order to extract relevant features. The detection has been tested over the widely adopted support vector machine (SVM), decision trees (DTs), random forest (RF), and XGBoost classification methods. The performance of all the classifiers was evaluated and the accuracy of the results was considered excellent. We hope that this study will be used as a reference for eye rubbing detection and widely adopted in wearable devices such as smartwatches.

The remaining sections of this paper are organized as follows. Section 1 performs a survey on recent studies on the detection of KC diagnosis and face touching activities through artificial intelligence techniques. Section 2 aims to familiarize the reader with the functional components of the system and presents the data collection and feature extraction process, as well as the experimental protocol. Section 3 depicts the experimental results, while Section 4 discusses the system performance through a relevant analysis of the findings. Finally, Section 5 outlines some concluding remarks and poses future research directions in the field.

1.1. Background and Motivation

The evolution of information and communication technology (ICT) has not only been induced by the advances in processing power, storage capacity, network bandwidth, and also the ability to capture and transmit huge amounts of even real-time data. It is also stimulated by the rapid progress of the last three decades in the field of artificial intelligence that lends additional forces towards learning, mastering, and becoming experts in various other scientific fields. Healthcare is one of the fields most affected by this progress. New instruments, models, algorithms, and techniques are being offered by ICT to facilitate better patient screening, early prognosis, accurate diagnosis, or even continuous monitoring. Wearable electrocardiogram (ECG) monitoring systems [7], cancer detection through artificial intelligence algorithms [8], and gesture recognition to avoid the spread of contagious diseases [9], to name a few, are some indicative examples of healthcare ICT.

The objective of this section is to provide the reader with up-to-date literature findings on artificial intelligence techniques for keratoconus disease diagnosis and the detection of face or eye touching. The latter is identified as a risk factor for the emergence of keratoconus. Despite the detailed literature review, we found no articles that apply any ICT methods for keratoconus prevention through eye rubbing detection. Thus, we initially focused on reviewing AI-based diagnosis efforts and aimed to present research findings of face touching prevention techniques that loomed during the COVID-19 pandemic era. Thus, we aimed to close this research gap and propose the first solution, to the best of our knowledge, for keratoconus prevention through eye rubbing detection.

1.2. Keratoconus Diagnosis through Artificial Intelligence Techniques

Keratoconus is a non-inflammatory disease of the eyes that has attracted joint research efforts from the fields of medicine, computer science, and engineering. The literature is rich in articles that provide sound solutions for accurate keratoconus diagnosis, mainly based on artificial intelligence techniques. Isaarti et al. [10] proposed a computer-aided diagnosis system that can detect keratoconus in early stages. The main objective of this research effort was to develop a stable and low-cost system for computer-aided diagnosis. It combines a Grossberg–Runge–Kutta architecture and a feedforward neural network. Corneal elevation and thickness data of 851 subjects were used for training and validation. The system achieved an accuracy of 96.56% versus 89% for Belin–Ambrosio deviation, and 79% for topographic keratoconus classification. The computation time was reduced by 70% with respect to traditional machine learning techniques.

A study conducted by Arbelaez et al. [11] considered 2243 keratoconus and 1259 healthy eyes in order to develop a classification method for disease diagnosis based on corneal measurements provided by a Scheimpflug camera combined with Placido corneal topography. The measurements were analyzed using a SVM classifier. Some of the main measures analyzed were front and back corneal curvature symmetry index, the best fit radius of the front corneal surface, the Baiocchi–Calossi–Versaci (BCV) front index, and the BCV back index. The accuracy was calculated with and without data generated from the posterior corneal surface but corneal thickness too. In both cases, the results surpassed 95%, while the SVM increased its precision when anterior and posterior corneal data were included with accurate diagnoses exceeding 97% for normal eyes.

The SVM was the algorithm of choice in a study conducted by Ruiz Hidalgo et al. [12]. The purpose of their study was to automatically identify corneal patterns on a combination of 22 parameters gathered through Scheimpflug tomography that was performed using Pentacam measurements. The dataset consisted of 860 eyes and was further divided in five groups. Three classification tasks were defined and tested in terms of accuracy. The discrimination of a keratoconus eye versus a normal one achieved the highest accuracy of 98.9%. The accuracy of forme fruste versus normal eye was 93.1%, while the five-group classification accuracy was 88.8%

Yousefi et al. [13] developed an unsupervised machine learning algorithm that was applied to optical coherence tomography (OCT) images of 12,242 eyes. The challenge was to develop a non-biased algorithm based on either the clinician or the patient, and consisted of three stages: (i) initially, principal component analysis was used for dimensionality reduction (from 420 to 8 components); (ii) the second step involved manifold learning so as to further reduce the components to two eigen-parameters, and (iii) density-based clustering was applied to the eigen parameters so as to identify keratoconus cases. Four clusters were determined, while the specificity of identifying healthy eyes from eyes with keratoconus was 94.1% and the sensitivity of identifying eyes with keratoconus from healthy eyes was 97.7%.

In another study by Yousefi et al. [14], artificial intelligence techniques were adopted to predict future needs for keratoplasty intervention. They used 12,242 corneal OCT images from 3162 subjects, and 3318 measurements were gathered from the visits of each patient. Unsupervised machine learning was applied and five distinct eyes clusters were identified so that each one corresponded to a different likelihood of future keratoplasty. The main contribution of this study is that the physician may be supported to identify patients with a higher likelihood for keratoconus disorder and consequent need for surgical care in the future.

A convolutional neural network is applied in the KeratoDetect algorithm [15] to support the diagnostic process of keratoconus (i.e., to determine whether an eye is affected or not by keratoconus). The algorithm is trained to analyze the corneal topography of the eye and consequently extract and learn about keratoconus eye features. The accuracy of the algorithm is reported to be 99.33% and it sought to be a reliable screening tool at the ophthalmologist's disposal. The dataset used in this study resulted from the SyntEye KTC model as it was difficult to gather real eye topographies. However, this was not considered to pose a bias in the algorithm's assessment.

Hallett et al. [16] compare the accuracy of machine learning models for the prediction of disease progression. Data from 124 keratoconus patients were processed. Classification was based on supervised multilayer perceptron and unsupervised variational autoencoder, and both methods achieved high accuracy. Specifically, in the case of the multilayer perceptron (MLP), the AUC for classes 1, 2, 3, and 4 (early to late stages) obtained values 0.92, 0.86, 0.97, and 0.94, respectively, while in the case of the variational autoencoder (VAE), the respective AUC values for the same classes were 0.91, 0.87, 0.79, and 0.99, indicating better clustering accuracy.

A study that intended to contribute to the knowledge on keratoconus detection based on corneal imaging data was proposed by Lavric et al. [17]. The challenge was to provide a model that was able to detect keratoconus in early stages. Several machine learning algorithms were applied and benchmarked over real medical data (i.e., corneal topography, elevation, and corneal pachymetry) from OCT-based corneal topography. The results of a comparison of 25 different models achieved accuracy that varied from 62% to 94%. The SVM obtained the better accuracy score of 94% and the algorithm is considered as a candidate for integration into corneal imaging devices.

EMKLAS [18] was developed as a predictive model that uses demographic, optical, and geometric variables in order to quickly and correctly classify keratoconus cases. The limitations of this study include the subjective diagnosis of ophthalmological physicians that was used for the model training, as well as the small sample size, as only subclinical KC eyes were included in the dataset. EMKLAS is intended to classify early and mild cases and was trained on data gathered from 178 subjects (104 keratoconus patients, i.e., 61 early cases and 43 mild cases, and 74 healthy patients). Their classifier used an ordinal logistic regression model that combined 27 parameters and an overall accuracy of 73% that is low but promising enough taking into account that it was used to diagnose early- and mild-stage patients.

1.3. Face Touching Detection

The publications reported in the Section 1.2 present solutions that achieve high accuracy, yet are able to discriminate between healthy and unhealthy eyes (i.e., eyes already suffering from keratoconus). However, in medicine, prevention is more important than a cure. Consequently, special research efforts are devoted to preventive actions rather than diagnostic ones. As a result, young populations can be prevented from developing habits associated with keratoconus risk factors. As the available technology allows for the continuous monitoring of human subjects through small factor wearable devices, various solutions have been developed in order to avoid movements that may place people at risk. Hands and face are of special interest in various creative solutions that intend to protect people against infectious diseases, such as COVID-19, influenza, etc.

Accordingly, Sudharsan et al. [19] developed a 3D motion sensor that can identify hand-to-face motion with an ultimate objective of avoiding eye, nose, and mouth touching, and can consequently eliminate the chance of spreading viral infections, such as COVID-19, etc. The main contributions of their study are as follows: the 3D multi-sensor data for 2071 dynamic hand-to-face movements, the feature extraction process, and the COVID-away one-class classification model that is smartwatch-aware and with minimal loss (and was evaluated by factory workers in a manufacturing plant). For evaluation purposes, the authors collected a dataset using their sensor and experimental results, and revealed that the minimum covariance method produced the best F1 score of 0.93 using 39 features extracted by the accelerometer sensor.

FaceOff [20] is another project that makes use of a wrist-worn device to detect face touching. The device can detect the motion of raising one's hand towards the face. Although the authors only collected and made use of accelerometer data, they employed a concrete protocol of data collection. The author himself was the experiment subject, and the activities gathered through a smartwatch were divided in two wide categories (based on touch and no touch). As a touch activity, the author considered touching an eye, ear, mouth, nose, etc., while in the second category, based on no touch, activities were classified as page flipping, typing, writing, etc. For testing purposes, three more subjects participated and the results were encouraging, as 82 out of 89 touched their faces (92% success), with a false-positive rate of 0.59%. Random forest models were used in the training phase of the project.

Another study on face touching detection was proposed by Bai et al. [21]. The control of COVID-19 and respiratory disease against spontaneous hand motions on the face is the main objective of this research. To this end, smartwatches were employed through a relevant application that can identify motion signatures and map them to face touching. It included 10 participants (each of them wore a smartwatch) and classified 10 actions as activities based on face touching and no face touching. The data gathered were analyzed through logistic regression (LR), support vector machines (SVMs), decision trees, and random forest methods, and were evaluated based on the F1 score. Although logistic regression explores the linear relationship between features, it outperformed all other methods that can explore non-linear relationships. It achieved a mean F1 score of 0.90, random forest achieved an F1 score of 0.88, while support vector machines and decision trees achieved F1 scores of 0.86 and 0.84, respectively.

Artificial intelligence is the method of choice in a system [22] that intends to become another weapon in the fight against COVID-19. The proposed system was designed to pose no specific hardware requirements and rely on a relatively simple software. Furthermore, it is platform independent and is easy to implement. It employs the BodyPix 2.0 machine learning tool based on the ResNet50 convolutional neural network. Its objective is to implement face touching recognition software that detects the intersection of hands and face through a web camera with an accuracy of 91.3%.

The COVID-19 pandemic was the trigger for another research project [23] that made use of radio frequency technology to capture special characteristics of human activities. Consequently, the gathered data were analyzed through a set of fuzzy logic IF-THEN rules that can recognize different subjects entering or leaving a room, coughing, sneezing, and face touching through walls. The system achieved an overall accuracy score of 96%.

A relevant approach [24] proposed a computer vision system that, apart from mask detection and social distance measuring, detects face–hand interactions. Additionally, the authors addressed the issues of distinguishing between proper and improper mask wearing by investigating several CNN models over a dataset that included more than 40,000 images that depict relevant activities (e.g., wearing a mask or rubbing a nose). System validation took place on existing datasets with no adaptation. In the case of face–hand interaction detection, the best accuracy achieved was 93.35% using the EfficientNet-b2 convolutional neural network model. Face mask detection achieved an accuracy of 98.20% through the Inception-v3 model, while the social distancing control capability of the system was evaluated over six videos and the best reported accuracy was 96.51%.

CovidAlert [25] is a smartwatch system that intends to predict face touching and alert users in an energy-efficient manner by limiting the execution of algorithms during idle periods (i.e., periods of no motion). Accelerometer and gyroscope data are used to train an ensemble composed of the STA/LTA algorithm and the random forest model with an accuracy of 88.40%.

A classifier based on dynamic time warping was used to classify the activities of the participants, recorded through smartwatches, in a research effort conducted by Chen et al. [26]. The two activities groups were based on face touching and no face touching, and the authors achieved high-accuracy results up to 99.07%. Due to within-category confusions being favored over between-category confusions in multiclass classification, the DTW classifier performed better in binary classification. Moreover, the authors did not consider the differentiation between face touching activities, as the main objective was to classify data into one of two groups, while the spread COVID-19 infection posed health risks.

Another study in the joint field of artificial intelligence and inertial sensors was reported by Alesmaeil and Sehirli [27] who employed smartwatches and performed detection through convolutional neural networks. Their dataset consisted of 28,000 training samples containing various hand motion types, body positions, and hand orientations. Two different datasets were collected for left and right hands, and the main challenge was to overcome precision issues of hand movement in 3D space. The system alerts the user to the user through a vibration signal and an F1 score of 95.85% was achieved.

Similarly, Rizk et al. [28] proposed TouchAlert that can predict face touch activity and consequently warn the user before touching his/her face. The sensors involved are a gyroscope and an accelerometer, while the classification module consisted of a deep and fully connected neural network that was trained using adaptive moment estimation. The F1 score reported by the experimental results exceeded 0.98 when there were three location layers involved.

D' Aurizio et al. [29] also tried to depict the social role of hygiene rules and the potential to enforce them through a combination of new technology artifacts. The main challenge of the authors was to employ widespread technologies, while the system implementation should remain simple with no complex installation and/or hardware assembly procedures needed. In fact, they employ smartwatches and magnetometers so as to distinguish between safe (not compatible with face touching) and unsafe (potential face touching) movements. The system described was validated under two scenarios. The first one used a magnetometer, while the second one detected safe and unsafe movements with an accelerometer. The first algorithm achieved correct detection in 91.3% of cases, while the latter one correctly detected 92.60% of cases. However, in the first case, the false-positive incidents detected 3.20% of cases, while the second detected 38.10% of cases.

The unconscious nature of face touching is tackled in the study conducted by Michelin et al. [9]. The main contribution of this study is based on the real-time understanding of hand motion in order to predict and prevent face touching. For that reason, an inertial measurement unit was applied to obtain features of hand movement and face touching. Hand movement prediction was performed through 1D-CNN filters over 4800 trials from 40 participants. The results reveal that although the time needed to touch the face is 1.2 seconds, the proposed system achieved accuracy of 92% in 0.55 s, which is enough to take preventive actions such as smartwatch vibration etc.

As stated previously, this study intends to build on top of mature research fields by blending diagnosis through artificial intelligence methods (as depicted in Section 1.2) and eye rubbing sensing techniques (according to solutions described in Section 1.3). Visual detection systems are not within the scope of our study, as current research efforts either focus on static images and recorded video or make use of web cameras. In both cases, the visual detection device cannot be wearable and should be in front of the subject (and not on it).

2. Materials and Methods

2.1. High-Level Architecture of the Proposed System

This section describes the high-level architecture of the system that comprises an inertial measurement unit (IMU) and the detection component that involves machine learning algorithms. The IMU is the sensing device dedicated to collecting gyroscope and accelerometer data through a SparkFun MPU6050 sensor [30], mounted on a Raspberry-Pi minicomputer through a wired connection. The IMU data were used to measure hand movement, and machine learning algorithms were used to detect whether a hand movement involves eye rubbing in real time. Sonic feedback was used to alert the user to stop their hand movement.

Figure 1 depicts a detailed description of the proposed system. The detection module consisted of three discrete stages: data processing, feature extraction, and a machine learning layer. As a prevention component, we applied a sonic feedback module. The input of the detection component was a sensor which incorporated an accelerometer along with a gyroscope that constantly provided six different types of data (i.e., x, y, and z components for both the accelerometer and the gyroscope). The acquired data were segmented by sliding overlapping windows, a commonly used technique used to capture transition effects. For each window, appropriate features were extracted, resulting in the feature vector. Finally, the feature vector was fed in a machine learning model and classified independently (i.e., instantaneous classification). As soon as eye rubbing was detected, the response module provided the user with a feedback action in order to remind the subject to stop the hand movement. The algorithm flow for data collection, feature extraction, and classification processes presented in this paper is depicted in Figure 2.

All the technical information on the dataset, the experimental setup, and the necessary information of the sensing device are provided in the subsequent sections.

2.2. Eye Rubbing Detection through Machine Learning

2.2.1. Experimental Protocol and Evaluation Measures

An extensive search of relevant databases on eye rubbing activity IMU signals revealed that no such dataset is available, which meant that one needed to be created. Two participants participated in the data collection process. The first participant was a healthy male, 51 years old, while the second one was also male, recently recovered from keratoconus through crosslinking operation, and was 15 years old. The acquisition was repeated five times (i.e., 50 min) for each of the participants. The collected data were randomly split and used for the training and testing phases. An important aspect of the experimental protocol was that the second participant had recently recovered from keratoconus, as a result of eye rubbing according to the diagnosis. Consequently, the resulting dataset had a duration of 100 min with 500 s of eye rubbing activity. The sampling frequency was set to 50 Hz according to similar reported setups [31].

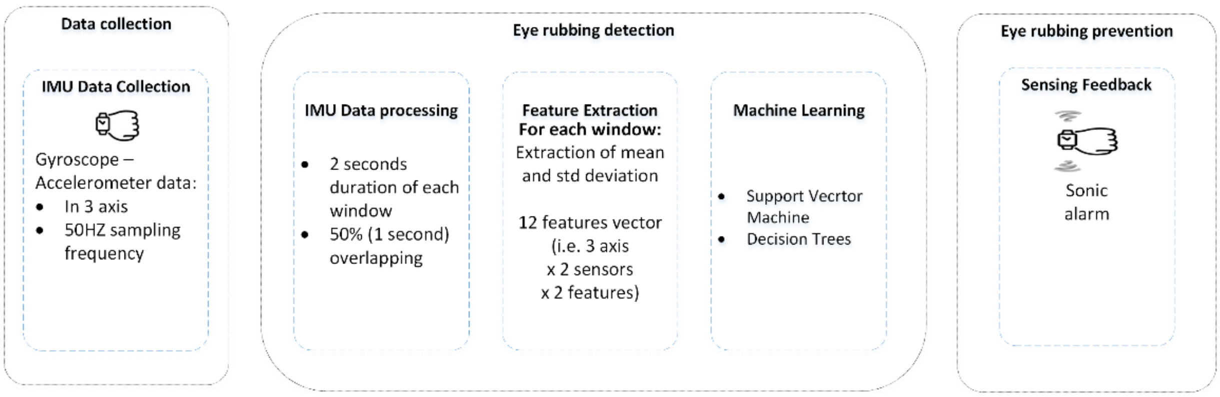


Figure 1. High-level architecture and case scenario of the proposed system.

BEGIN

Initialize the sampling frequency, the amount of IMU data used, and the buzzer duration

Initialize the IMU sensor data and feature vector X_{Data}, $V_{Features}$ = *Null*

repeat

X_{Data} ← *Read the last 100 IMU sensor data with a 50 Hz sampling frequency. Each measurement contains 3 acceleration and 3 rotation values*

$V_{Features}$ ← *Extract the features vector from the* X_{Data}

E_r ← *Perform classification on the* $V_{Features}$ *vector and return true if eye rubbing activity is detected (otherwise, return false)*

if E_r *is true* **then**

Activate the buzzer for 0.5 seconds to alarm the user

else

Do not activate the buzzer

end if

until *application is terminated*

Figure 2. The algorithm outline for eye rubbing detection.

Among machine learning classifiers, support vector machines, decision trees, random forest, and XGBoost were selected for the experimental evaluation. These classifiers were selected because of their low computational cost and their suitability for wearable applications. Nowadays, the processing and memory requirements of these methods can be fulfilled by smartphones with satisfactory recognition performance. Furthermore, high-accuracy results have been achieved in similar activity recognition problems [32–34].

2.2.2. Data Collection

The experimental dataset for eye rubbing detection consists of accelerometer and gyroscope data. The SparkFun MPU6050 sensor was adopted as it provides both accelerometer and gyroscope sensing capabilities. The sensor transmits the data to a Raspberry-Pi minicomputer through wired connection (Figure 3). To the best of our knowledge, there is not database that exists with rubbing eye activity data. This equipment was properly mounted on the participant's right wrist (Figure 4). The acquisition test had a 10 min duration. The participants were asked to perform usual free activities in an uncontrolled environment. A buzzer was actuated every one minute for five seconds, indicating the participants' need to rub their eyes.

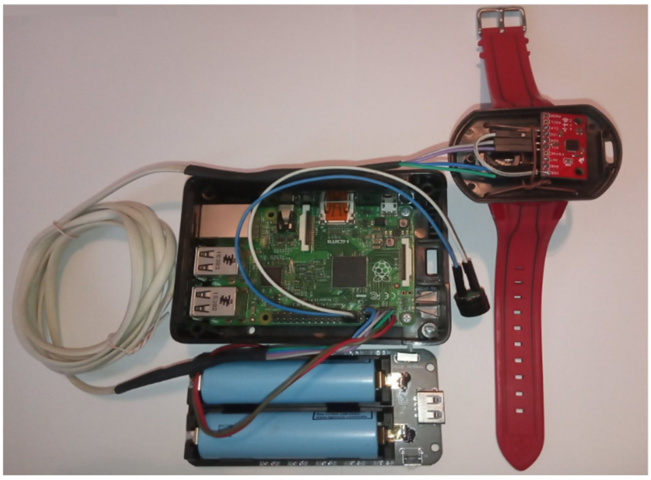


Figure 3. The experimental equipment with the SparkFun MPU6050 accelerometer, gyro sensor, and the Raspberry-Pi minicomputer.

Figure 4. The sensing device on the participant's wrist.

Figure 5 depicts a segment of the dataset. One second corresponded to 50 samples, since the sampling frequency was 50 Hz. The raw dataset consisted of acceleration values (accel-x, accel-y, and accel-z) and rotation values (gyro-x, gyro-y, and gyro-z) in the 3 axes. The output scale for each value was [−32,768, +32,767]. The signals corresponded to one-minute random activities, while an eye rubbing activity occurred in the last five seconds (samples 3000–3250). Figures 5 and 6 focus on the eye rubbing activity.

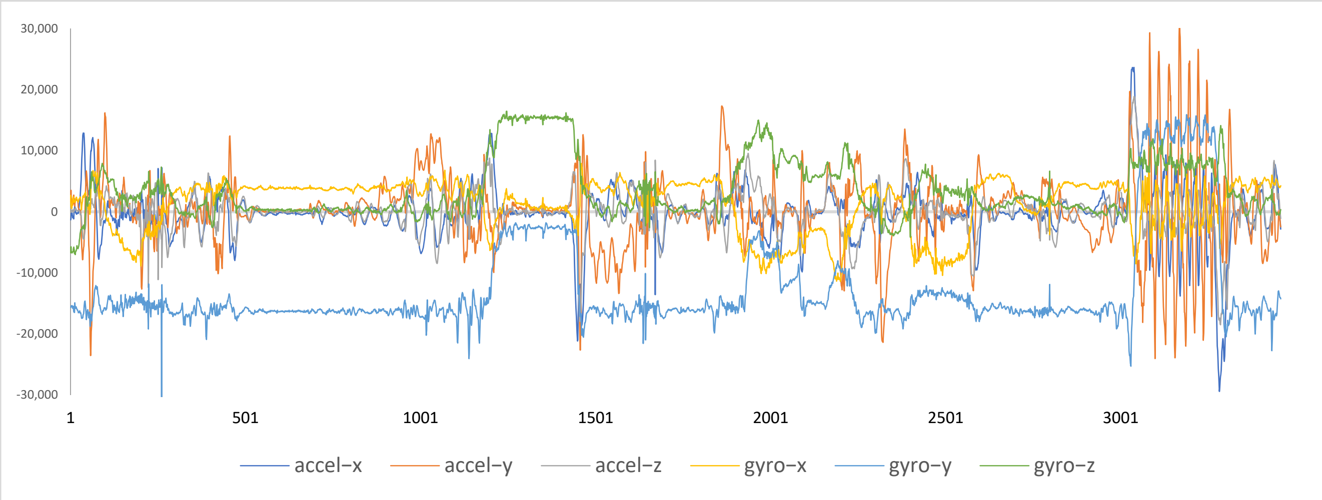


Figure 5. Accelerometer and gyroscope data.

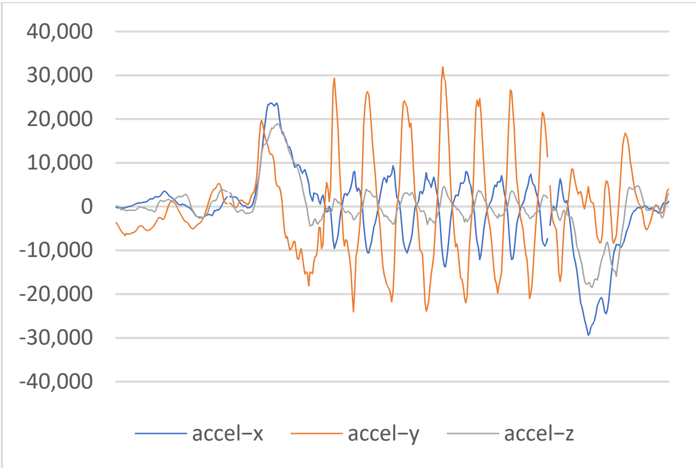


Figure 6. Eye rubbing accelerometer data.

Figures 6 and 7 focus on the eye rubbing activity. Figure 6 presents the acceleration data and Figure 7 presents the gyroscope data. The rotary movement of the hand during eye rubbing can be deduced by the periodicity of the acquired signals.

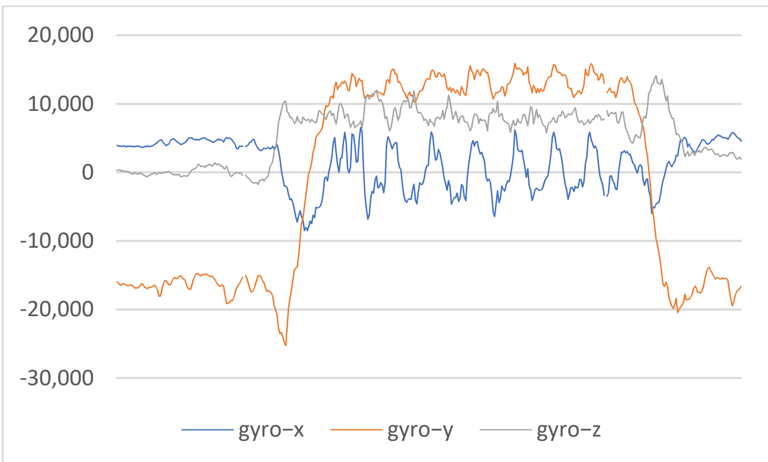


Figure 7. Eye rubbing gyroscope data.

Table 1 reports the set of attributes used in our study, including information on hand acceleration and rotation in space.

Table 1. Attributes and dataset statistics used in our study.

	Acceleration			Rotation		
	x	**y**	**Z**	**x**	**y**	**z**
min	−32,768	−32,768	−32,768	−32,768	−32,768	−20,564
max	32,767	32,767	32,767	32,767	32,767	32,767
mean	−328.3	330.8	37.7	94.6	−6943.7	9115.7
stdev	3184	6400.2	2952	4836.1	7942.1	6488.5
median	−293	211	−12	48	−3976	11,780
kurtosis	24.5	14.5	23.6	3	−0.3	−1.1
skewness	−0.7	0.1	0.8	0.4	0.4	−0.6

2.2.3. Features Extraction

The time domain data are processed over sliding overlapping windows in order to consider transition effects. A window of 2 s was used with 50% (1 s) overlapping. The mean value and standard deviation were used as features. The feature extraction setup is depicted in Figure 8. Thus, 2 features were extracted for each axis and for each sensor (accel. and gyro), resulting in a vector consisting of 12 features. The specific setup was chosen as it was proven to be more effective in a relevant study [31].

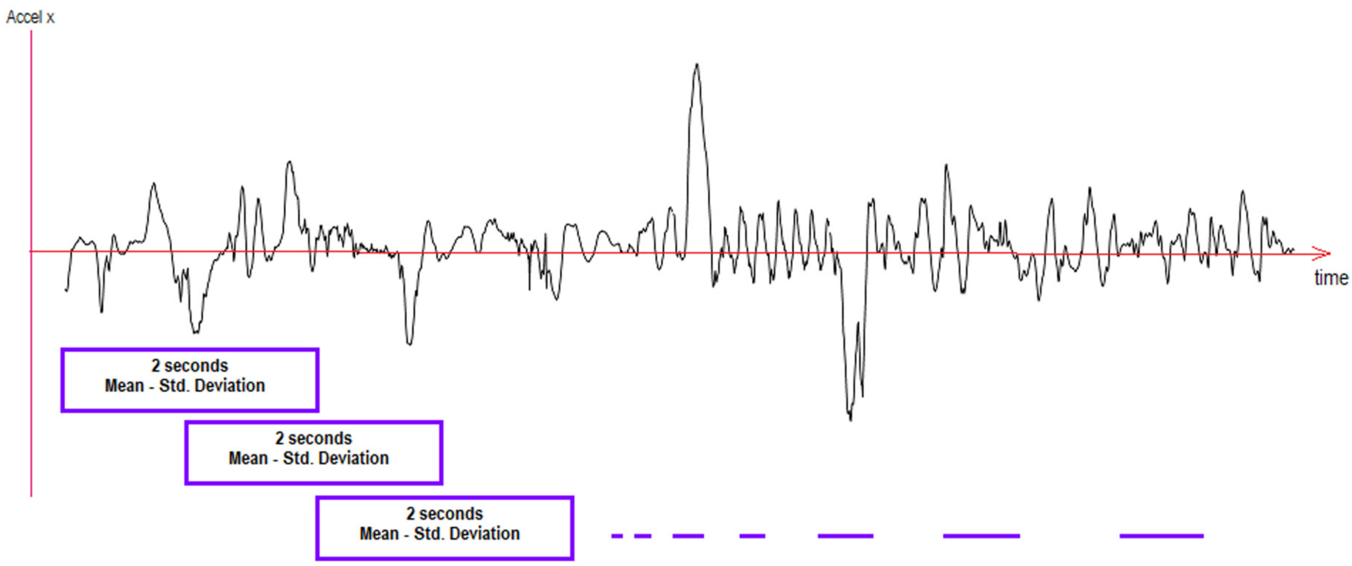
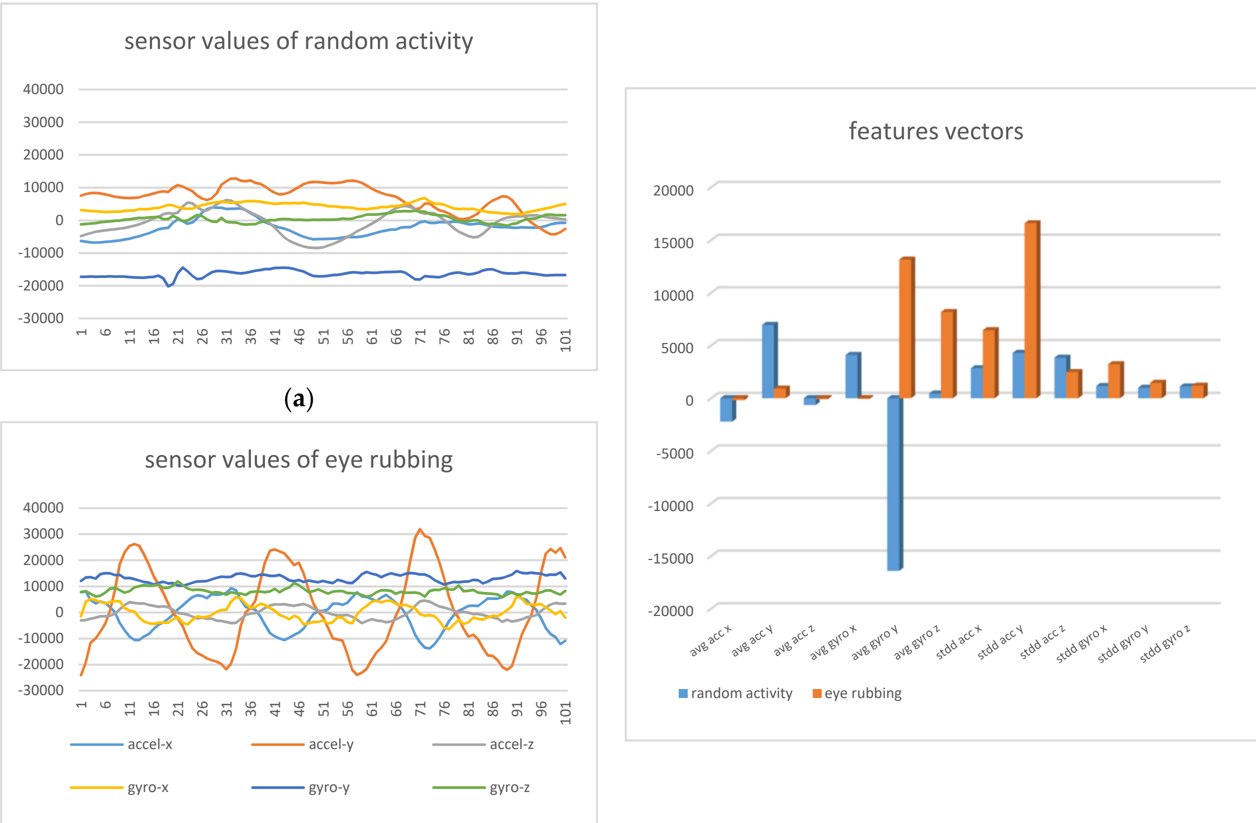


Figure 8. Feature extraction along accelerometer data on x axis.

Figure 9 depicts the sensors' raw data on random and eye rubbing activity and demonstrates the relationship between these data and the feature vector. A sample instance of images was used since the feature vector needs a large amount of data (600 values) to be calculated.



(**b**) (**c**)

Figure 9. IMU sensor data during: (**a**) 2 s random activity; (**b**) 2 s eye rubbing; (**c**) respective feature vector values.

3. Results

As already mentioned, four conventional classification methods (support vector machines, decision trees, random forest, and XGBoost) were evaluated and compared. The experiments were implemented and conducted within the Python Scikit-Learn environment [35]. The SVM is a popular machine learning classification algorithm introduced in the 1960s, and has provided excellent results since then. The main concept of the algorithm is to find a boundary that separates the data in such a way that the classification error is minimized. The main hyperparameter of the method is the kernel. The experiments were implemented and conducted within the Python Scikit-Learn environment [35] using radial kernel functions. On the other hand, DTs create a model that predicts the value of a target variable by learning simple decision rules from the data features. The Scikit-Learn environment was used, while the Gini impurity hyperparameter was selected as the splitting algorithm. Random forests [36] are made up of tree predictors, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. In our setup, the forest consisted of 100 trees and the Gini impurity hyperparameter was also selected as the splitting algorithm. Finally, the XGBoost [37] classifier was used to implement gradient-boosted decision trees designed for speed and performance. A learning rate of 0.1 was applied in this classifier. The hyperparameters of the models used are provided as supplementary material.

The dataset was randomly divided in a training and a testing set; 60 min of free activities (including 60 eye rubbing activities) formed the training set, while the other 40 min of free activities (including 40 eye rubbing activities) formed the testing set. The evaluation was based on performance metrics, confusion matrices, and ROC curves.

Figure 9 depicts IMU sensor raw data during (a) a random activity and (b) eye rubbing activity. The raw dataset consisted of acceleration values (accel-x, accel-y, and accel-z) and rotation values (gyro-x, gyro-y, and gyro-z) in the three axes. The respective features of these segments are presented in (c). More specifically, the average values of acceleration and rotation formed the first six features (avg acc x, avg acc y, avg acc z, avg gyro x, avg gyro y, and avg gyro z), while the other six features were formed by the standard deviation of the acceleration and rotation data (stdd acc x, stdd acc y, stdd acc z, stdd gyro x, stdd gyro y, and stdd gyro z). The relative position of the wrist (gyroscope values) and its movement characteristics (acceleration values) formed a pattern that is reflected in these features. The rotary movement of the hand during eye rubbing can be noticed by the periodicity of accel-x and accel-y in (b). This information is reflected in the feature vector through the standard deviation of the acceleration in axes x and y (stdd_acc_x and stdd_acc_y).

The test set consisted of 2400 vectors of random activity and 160 vectors of eye rubbing. Table 2 presents and compares the experimental results of the classifiers through confusion matrices and Table 3 depicts the precision, recall, and F1 score metrics. The precision metric demonstrates the accuracy of positive predictions and our experiments show that SVM, RF, and XGBoost methods achieved satisfactory results with values exceeding 95%. On the other hand, DT performance hardly exceeded 68%. Recall, also known as sensitivity, is the fraction of correctly identified positive predictions. For eye rubbing activity, all methods achieved satisfactory performance on this metric with values exceeding 94%. F1 score, precision, and recall helped to find the harmonic mean of the two values. The accuracy metric measured the accuracy of all predictions (positive and negative). Our experimental results denote that all methods attained high accuracy; SVMs, RF, and XGBoost achieved an accuracy score of 0.99, while DTs achieved an accuracy score 0.96. The macro-averaged F1 score was computed using the arithmetic mean of all the per-class F1 scores, while the weighted-average F1 score was calculated by taking the mean of all per-class F1 scores while considering each class's support.

Table 2. Combined confusion matrices of experimental results.

SVM		**Actual**		**DT**		**Actual**	
		Random Activity	**Eye Rubbing**			**Random Activity**	**Eye Rubbing**
Predicted	Random activity	2398	6	Predicted	Random activity	2328	8
	Eye rubbing	2	154		Eye rubbing	72	152
RF		Actual		XGBoost		Actual	
		Random activity	Eye rubbing			Random activity	Eye rubbing
Predicted	Random activity	2399	7	Predicted	Random activity	2392	10
	Eye rubbing	1	153		Eye rubbing	8	158

Furthermore, the AUC refers to the area under the sensitivity–specificity curve and is a metric of the model's ability to make distinctions between classes [38,39]. Higher AUC values indicate a better accuracy of the model. Figure 10 presents the AUC achieved by both methods. The AUC score for the SVM was 0.999, while the relevant score for DTs was 0.956.

Table 3. The precision, recall, and F1 score metrics.

	Precision				Recall				F1 Score			
	SVM	**DTs**	**RF**	**XGB**	**SVM**	**DTs**	**RF**	**XGB**	**SVM**	**DTs**	**RF**	**XGB**
Random activity	1.00	1.00	1.00	1.00	1.00	0.97	1.00	1.00	1.00	0.98	1.00	1.00
Eye rubbing	0.99	0.68	0.99	0.95	0.96	0.95	0.96	0.94	0.97	0.79	0.97	0.94
Accuracy									0.99	0.96	0.99	0.99
Macro avg	0.99	0.84	1.00	0.97	0.98	0.96	0.98	0.97	0.99	0.89	0.99	0.97
Weighted avg	1.00	0.98	1.00	0.99	1.00	0.97	1.00	0.99	1.00	0.97	1.00	0.99

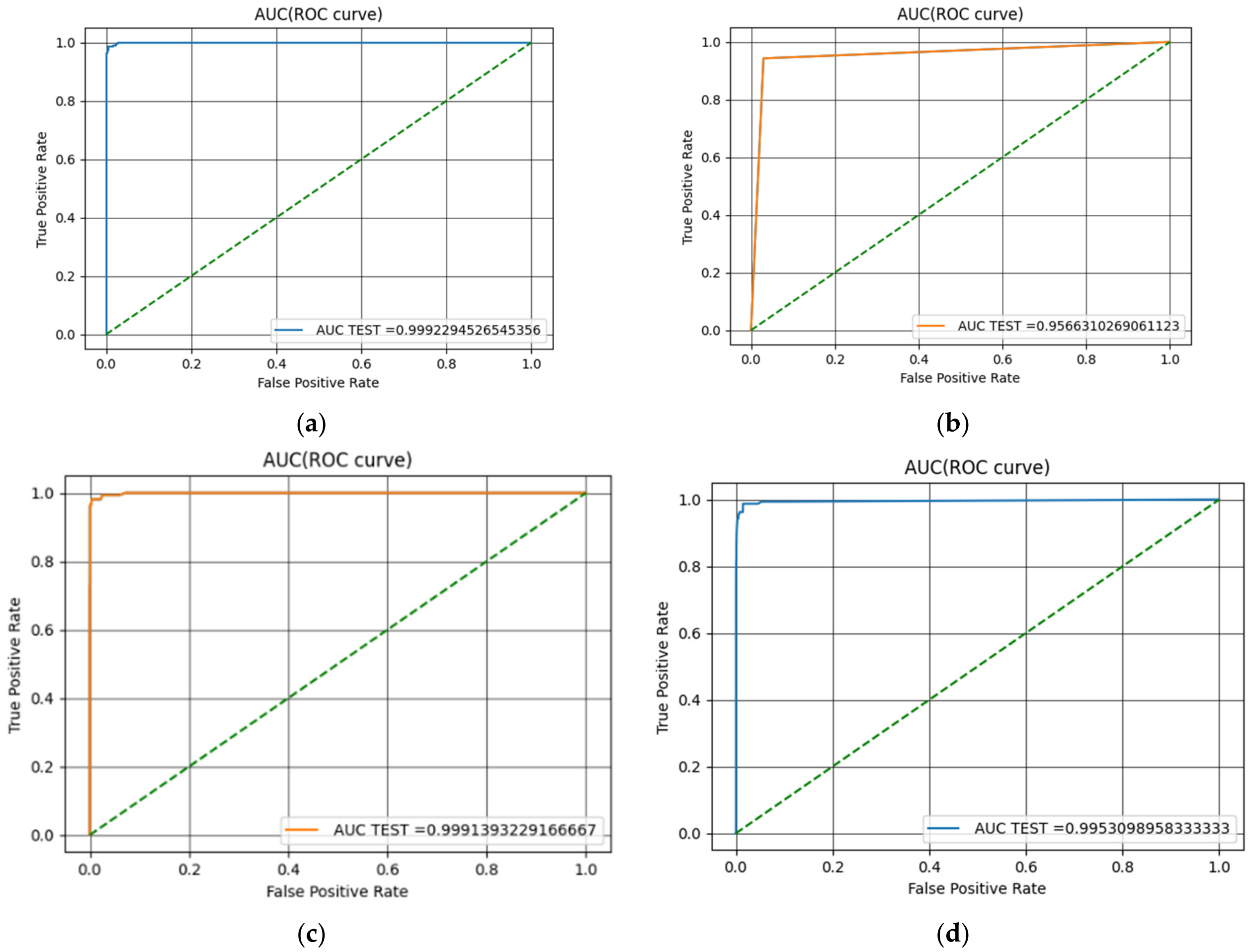


Figure 10. The AUC for the (**a**) SVM, (**b**) DTs, (**c**) RF, and (**d**) XGBoost classifiers.

The SVM slightly outperformed DTs for reasons that are further discussed in the next section. The macro-averaged F1 score was computed using the arithmetic mean of all the per-class F1 scores, while the weighted-average F1 score was calculated by taking the mean of all per-class F1 scores while considering each class's support. The execution time for the detection of one frame was about 0.2 msec for both methods, thus making the real-time implementation feasible.

The FaceGuard system [9], which elaborates on understanding hand motion in real time and prevents face touching, is closely related to what is proposed in our study. The main difference is that we focused on a specific part of the face—the eyes—in order to avoid keratoconus, while FaceGuard makes no distinction between the nose, mouth, or eyes, as its objective is to avoid COVID-19 transmission. Both studies attain high-accuracy results that exceed 92% in the case of FaceGuard, while the reported accuracy of our system is 99%.

4. Discussion

The correlation between eye rubbing and eye disorders (such as keratoconus) has already been justified in the literature. In order to prevent the development of such diseases, it is of critical importance to avoid activities that are common and mostly instinctive. In this paper, the potential of eye rubbing detection and its forestallment was investigated.

The high-level architecture of the proposed system was thoroughly described, and several implementation details were provided. The inertial measurement unit, the features extraction stage, and the machine learning processes were our main points of interest. Accelerometer and gyroscope data were collected using a wrist-mounted sensor that is practically installed in every popular smartwatch in the market. In total, 60% of the data were randomly selected for training purposes and the rest was used for testing purposes.

The main challenge concerned the implementation of a solution of low computational resources that could fit to small-sized wearable devices. Consequently, SVMs, RF, XGBoost, and DTs were selected as generally simple methods and were further evaluated.

As far as detection ability is concerned, the classification results provided by the selected machine learning algorithms were analyzed in terms of precision, recall, F1 score, and area under the curve (AUC). After comparing the resulting confusion matrices for the eye rubbing class, the best results were obtained with SVMs and RF, with a precision value = 099, a recall value = 0.96, and an F1 score = 0.97. The XGBoost classifier follows with a precision value = 0.95, a recall value = 0.94, and an F1 score = 0.94. As suggested by the results, DTs had a lower performance with a precision value = 0.68, a recall value = 0.95, and an F1 score = 0.79. In the random activity class, the metrics of all four methods were almost perfect, exceeding 0.97 for recall and 0.98 for the F1 score in case of DTs. Recall is important in cases where the actual positive cases should not go undetected, while a false alarm is not that important. Thus, these results indicate that the proposed system is a valuable tool in detecting eye rubbing and consequently preventing the occurrence of keratoconus disease. The tool's value is further upgraded by the small execution time overhead implied for the detection of one frame, which lends support to the claim that it is appropriate for real-time implementations.

In our study, the accuracy of all methods was exceptionally high—over 96%, and the main differences were noticed in the recall and precision metrics. The reason of this diversity mainly lies in the nature of the dataset and its features. The dimensionality of the feature vector (12 features) was better handled by the SVM and RF classifiers, as suggested by the results. Moreover, this is also an indication that more experiments with a larger dataset are necessary.

The conducted experiments reveal that the proposed system attains high accuracy for the detection of eye rubbing. Moreover, the system is highly expandable due to its modular design and implementation on the popular Raspberry Pi minicomputer. Consequently, additional functionalities can be easily added according to the user's needs.

The limitations of this study include discrimination among similar actions, such as face and nose rubbing, ear or head scratching, etc. For that reason, a larger sample with more subjects would allow for a dataset to be used to evaluate more classification methods and become appropriate for deep learning implementations. Moreover, more detection techniques should be evaluated and considered for inclusion in the system. Visual systems could be a candidate solution, but this would lead to a tradeoff between low-cost and wearable techniques versus sophisticated devices, but small-form gadgets are more expensive and harder to integrate.

5. Conclusions and Future Work

In this study, we focused on detecting and preventing eye rubbing, which is a major risk factor for keratoconus disease. No previous studies have been conducted in this field.

The proposed system employs an inertial measurement unit for the collection of hand motion data, while machine learning techniques are used for eye rubbing detection. The SVM, RF, XGBoost, and DT classification methods were applied and all of them attained high-accuracy results. Two participants were monitored during an acquisition test to assess eye rubbing activity in an uncontrolled environment. Each test had a duration of 10 min and was repeated 10 times for each subject. A major limitation of study work was the dataset size and the number of activities monitored. Future directions should focus on engaging more participants and expanding the activity types that are monitored (i.e., nose rubbing, head scratching, etc.). Consequently, more classification methods could be evaluated and obtain high-accuracy results in more complicated environments.

Finally, in our future study, we aim to enhance our proposed approach by adding the explainability/interpretability property [40].

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics12041028/s1. Table S1: The system's hyperparameters.

Author Contributions: Conceptualization, G.N. and T.K.; Methodology, T.K.; Software, G.N.; Data curation, G.N.; Writing – original draft, G.N. and T.K.; Writing – review and editing, G.N. and T.K.; Supervision, T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the authors, upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Totan, Y.; Hepşen, I.F.; Çekiç, O.; Gündüz, A.; Aydın, E. Incidence of keratoconus in subjects with vernal keratoconjunctivitis:: A videokeratographic study. *Ophthalmology* **2001**, *108*, 824–827. [CrossRef] [PubMed]
2. Kenney, M.C.; Gaster, R.N. Keratoconus. In *Ocular Disease*; Elsevier: Amsterdam, The Netherlands, 2010; pp. 42–48. [CrossRef]
3. Ferrari, G.; Rama, P. The keratoconus enigma: A review with emphasis on pathogenesis. *Ocul. Surf.* **2020**, *18*, 363–373. [CrossRef] [PubMed]
4. Hashemi, H.; Heydarian, S.; Hooshmand, E.; Saatchi, M.; Yekta, A.A.; Aghamirsalim, M.; Valadkhan, M.; Mortazavi, M.; Hashemi, A.; Khabazkhoob, M. The Prevalence and Risk Factors for Keratoconus: A Systematic Review and Meta-Analysis. *Cornea* **2020**, *39*, 263–270. [CrossRef] [PubMed]
5. Najmi, H.; Mobarki, Y.; Mania, K.; AlTowairqi, B.; Basehi, M.; Mahfouz, M.S.; Elmahdy, M. The correlation between keratoconus and eye rubbing: A review. *Int. J. Ophthalmol.* **2019**, *12*, 1775–1781. [CrossRef]
6. Sahebjada, S.; Al-Mahrouqi, H.H.; Moshegov, S.; Panchatcharam, S.M.; Chan, E.; Daniell, M.; Baird, P.N. Eye rubbing in the aetiology of keratoconus: A systematic review and meta-analysis. *Graefe's Arch. Clin. Exp. Ophthalmol.* **2021**, *259*, 2057–2067. [CrossRef]
7. Yang, Z.; Zhou, Q.; Lei, L.; Zheng, K.; Xiang, W. An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare. *J. Med. Syst.* **2016**, *40*, 286. [CrossRef]
8. Saba, T. Recent advancement in cancer detection using machine learning: Systematic survey of decades, comparisons and challenges. *J. Infect. Public Health* **2020**, *13*, 1274–1289. [CrossRef]
9. Michelin, A.M.; Korres, G.; Ba'Ara, S.; Assadi, H.; Alsuradi, H.; Sayegh, R.R.; Argyros, A.; Eid, M. FaceGuard: A Wearable System To Avoid Face Touching. *Front. Robot. AI* **2021**, *8*, 612392. [CrossRef]
10. Issarti, I.; Consejo, A.; Jiménez-García, M.; Hershko, S.; Koppen, C.; Rozema, J.J. Computer aided diagnosis for suspect keratoconus detection. *Comput. Biol. Med.* **2019**, *109*, 33–42. [CrossRef]
11. Arbelaez, M.C.; Versaci, F.; Vestri, G.; Barboni, P.; Savini, G. Use of a Support Vector Machine for Keratoconus and Subclinical Keratoconus Detection by Topographic and Tomographic Data. *Ophthalmology* **2012**, *119*, 2231–2238. [CrossRef]
12. Ruiz Hidalgo, I.; Rodriguez, P.; Rozema, J.J.; Ní Dhubhghaill, S.; Zakaria, N.; Tassignon, M.-J.; Koppen, C. Evaluation of a Machine-Learning Classifier for Keratoconus Detection Based on Scheimpflug Tomography. *Cornea* **2016**, *35*, 827–832. [CrossRef] [PubMed]
13. Yousefi, S.; Yousefi, E.; Takahashi, H.; Hayashi, T.; Tampo, H.; Inoda, S.; Arai, Y.; Asbell, P. Keratoconus severity identification using unsupervised machine learning. *PLoS ONE* **2018**, *13*, e0205998. [CrossRef] [PubMed]

14. Yousefi, S.; Takahashi, H.; Hayashi, T.; Tampo, H.; Inoda, S.; Arai, Y.; Tabuchi, H.; Asbell, P. Predicting the likelihood of need for future keratoplasty intervention using artificial intelligence. *Ocul. Surf.* **2020**, *18*, 320–325. [CrossRef] [PubMed]
15. Lavric, A.; Valentin, P. KeratoDetect: Keratoconus Detection Algorithm Using Convolutional Neural Networks. *Comput. Intell. Neurosci.* **2019**, *2019*, 8162567. [CrossRef] [PubMed]
16. Hallett, N.; Yi, K.; Dick, J.; Hodge, C.; Sutton, G.; Wang, Y.G.; You, J. Deep Learning Based Unsupervised and Semi-supervised Classification for Keratoconus. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020. [CrossRef]
17. Lavric, A.; Popa, V.; Takahashi, H.; Yousefi, S. Detecting Keratoconus From Corneal Imaging Data Using Machine Learning. *IEEE Access* **2020**, *8*, 149113–149121. [CrossRef]
18. Velázquez-Blázquez, J.S.; Bolarín, J.M.; Cavas-Martínez, F.; Alió, J.L. EMKLAS: A New Automatic Scoring System for Early and Mild Keratoconus Detection. *Transl. Vis. Sci. Technol.* **2020**, *9*, 30. [CrossRef]
19. Sudharsan, B.; Sundaram, D.; Breslin, J.G.; Ali, M.I. Avoid Touching Your Face: A Hand-to-face 3D Motion Dataset (COVID-away) and Trained Models for Smartwatches. In Proceedings of the 10th International Conference on the Internet of Things Companion, Malmi, Sweden, 6–9 October 2020. [CrossRef]
20. Chen, X.A. FaceOff: Detecting Face Touching with a Wrist-Worn Accelerometer. *arXiv* **2008**, arXiv:2008.01769.
21. Bai, C.; Chen, Y.-P.; Wolach, A.; Anthony, L.; Mardini, M.T. Using Smartwatches to Detect Face Touching. *Sensors* **2021**, *21*, 6528. [CrossRef]
22. Patel, S.; Madhani, H.; Garg, S.; Chauhan, M. An AI-Based Solution to Reduce Undesired Face-Touching as a Precautionary Measure for COVID-19. In Proceedings of the Computing Science, Communication and Security: Second International Conference, COMS2 2021, Gujarat, India, 6–7 February 2021; pp. 30–45. [CrossRef]
23. Miller, E.; Banerjee, N.; Zhu, T. Smart homes that detect sneeze, cough, and face touching. *Smart Health* **2021**, *19*, 100170. [CrossRef]
24. Eyiokur, F.I.; Ekenel, H.K.; Waibel, A. Unconstrained face mask and face-hand interaction datasets: Building a computer vision system to help prevent the transmission of COVID-19. *Signal Image Video Process.* **2022**, 1–8. [CrossRef]
25. Roy, M.; Seethi, V.D.R.; Bharti, P. CovidAlert—A Wristwatch-Based System to Alert Users from Face Touching. In *Pervasive Computing Technologies for Healthcare. PH 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*; Springer: Cham, Switzerland, 2022; pp. 489–504. [CrossRef]
26. Chen, Y.-P.; Bai, C.; Wolach, A.; Mardini, M.; Anthony, L. Detecting Face Touching with Dynamic Time Warping on Smartwatches: A Preliminary Study. In Proceedings of the Companion Publication of the 2021 International Conference on Multimodal Interaction, Montreal, QC, Canada, 18–22 October 2021; pp. 19–24. [CrossRef]
27. Sehirli, E.; Alesmaeil, A. Detecting Face-Touch Hand Moves Using Smartwatch Inertial Sensors and Convolutional Neural Networks. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 122–128. [CrossRef]
28. Rizk, H.; Amano, T.; Yamaguchi, H.; Youssef, M. Smartwatch-Based Face-Touch Prediction Using Deep Representational Learning. In *Mobile and Ubiquitous Systems: Computing, Networking and Services. MobiQuitous 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*; Springer: Cham, Switzerland, 2022; pp. 493–499. [CrossRef]
29. D'Aurizio, N.; Baldi, T.L.; Paolocci, G.; Prattichizzo, D. Preventing Undesired Face-Touches With Wearable Devices and Haptic Feedback. *IEEE Access* **2020**, *8*, 139033–139043. [CrossRef] [PubMed]
30. InvenSense. *MPU-6000 and MPU-6050 Product Specification, 8 2013. Rev. 3.4.*; InvenSense: San Jose, CA, USA, 2013.
31. Shoaib, M.; Bosch, S.; Scholten, H.; Havinga, P.J.M.; Incel, O.D. Towards detection of bad habits by fusing smartphone and smartwatch sensors. In Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), St. Louis, MO, USA, 23–27 March 2015; pp. 591–596. [CrossRef]
32. Khan, A.M.; Tufail, A.; Khattak, A.M.; Laine, T.H. Activity Recognition on Smartphones via Sensor-Fusion and KDA-Based SVMs. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 503291. [CrossRef]
33. Lu, H.; Yang, J.; Liu, Z.; Lane, N.D.; Choudhury, T.; Campbell, A.T. The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, Zürich, Switzerland, 3–5 November 2010; pp. 71–84. [CrossRef]
34. Ravi, N.L.M.; Dandekar, N.; Mysore, P. Activity recognition from accelerometer data. In Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence—Volume 3, IAAI'05, Pittsburgh, PA, USA, 9–13 July 2005; pp. 1541–1546.
35. Pedregosa, F.; Varoquau, G.; Gramfo, 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.* **2012**, *12*, 2825–2830. [CrossRef]
36. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
37. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the KDD '16: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794. [CrossRef]
38. Livieris, I.E.; Kiriakidou, N.; Kanavos, A.; Vonitsanos, G.; Tampakas, V. Employing Constrained Neural Networks for Forecasting New Product's Sales Increase. In Proceedings of the Artificial Intelligence Applications and Innovations: AIAI 2019 IFIP WG 12.5 International Workshops: MHDW and 5G-PINE 2019, Hersonissos, Crete, Greece, 24–26 May 2019; pp. 161–172. [CrossRef]

39. Livieris, I.E.; Pintelas, E.; Kiriakidou, N.; Stavroyiannis, S. An Advanced Deep Learning Model for Short-Term Forecasting U.S. Natural Gas Price and Movement. In Proceedings of the Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops: MHDW 2020 and 5G-PINE 2020, Neos Marmaras, Greece, 5–7 June 2020; pp. 165–176. [CrossRef]
40. Pintelas, E.; Liaskos, M.; Livieris, I.E.; Kotsiantis, S.; Pintelas, P. A novel explainable image classification framework: Case study on skin cancer and plant disease prediction. *Neural Comput. Appl.* **2021**, *33*, 15171–15189. [CrossRef]