applied sciences

MDPI

Article

Smart Rings vs. Smartwatches: Utilizing Motion Sensors for Gesture Recognition

Marc Kurz *, **Robert Gstoettner** and **Erik Sonnleitner**

Department of Mobility and Energy, University of Applied Sciences Upper Austria, 4232 Hagenberg, Austria; S1710455007@students.fh-hagenberg.at (R.G.); erik.sonnleitner@fh-hagenberg.at (E.S.)

* Correspondence: marc.kurz@fh-hagenberg.at; Tel.: +43-(0)50-8042-2827

Abstract: Since electronic components are constantly getting smaller and smaller, sensors and logic boards can be fitted into smaller enclosures. This miniaturization lead to the development of smart rings containing motion sensors. These sensors of smart rings can be used to recognize hand/finger gestures enabling natural interaction. Unlike vision-based systems, wearable systems do not require a special infrastructure to operate in. Smart rings are highly mobile and are able to communicate wirelessly with various devices. They could potentially be used as a touchless user interface for countless applications, possibly leading to new developments in many areas of computer science and human–computer interaction. Specifically, the accelerometer and gyroscope sensors of a custom-built smart ring and of a smartwatch are used to train multiple machine learning models. The accuracy of the models is compared to evaluate whether smart rings or smartwatches are better suited for gesture recognition tasks. All the real-time data processing to predict 12 different gesture classes is done on a smartphone, which communicates wirelessly with the smart ring and the smartwatch. The system achieves accuracy scores of up to 98.8%, utilizing different machine learning models. Each machine learning model is trained with multiple different feature vectors in order to find optimal features for the gesture recognition task. A minimum accuracy threshold of 92% was derived from related research, to prove that the proposed system is able to compete with state-of-the-art solutions.

Keywords: wearable computing; gesture recognition; human–computer interaction; machine learning

check for updates

Citation: Kurz, M.; Gstoettner, R.; Sonnleitner, E. Smart Rings vs. Smartwatches: Utilizing Motion Sensors for Gesture Recognition. *Appl. Sci.* **2021**, *11*, 2015. https://doi.org/10.3390/app11052015

Academic Editor: Jiro Tanaka

Received: 24 November 2020
Accepted: 19 February 2021
Published: 25 February 2021

1. Introduction

1.1. Overview

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. Mark Weiser [1] already predicted back in the year 1991 the modern age of smart devices we currently live in. About 15 years later, Steve Jobs revolutionized the smart phone [2] by replacing the user interface of styluses and hardware keyboards with multiple-finger gestures that allow even more natural computer interaction. However, these gestures require a surface to be performed against. This paper examines the process of gesture recognition, using gestures performed with the finger and/or the hand. Motion sensors of a smartwatch and a smart ring are used to record data, which are then used to train machine learning models in order to recognize gestures in real-time.

In the course of this paper a custom prototype of a smart ring is built to mirror the sensors that are already built into modern smartwatches. Both devices record data from accelerometer and gyroscope sensors [3] and send them wirelessly to a smartphone, which interprets the data and classifies the gesture. Additionally, different gesture recognition and machine learning aspects have to be considered. First, a common gesture catalogue is being established which feels intuitive for the users. The recorded data need to be analysed and preprocessed so that patterns can be found. Multiple models need to be trained and parameters need to be adjusted in order to select a model that is able to reliably classify gestures.

1.2. Motivation

The smart home [4,5] could be a major application domain for gesture recognition systems, because simple gestures could be used to control an entire home full of smart appliances, making the life of many people easier. However, this is only possible assuming the right infrastructure is established with compatibility between the gesture control system and the targeted smart appliance. Especially in a smart home a one-handed gesture recognition system is desirable. A smartwatch cannot provide one-handedness reliably, because it is impossible to interact with the watch using the hand it is mounted on. With a smart ring it would be possible since adjacent fingers can reach the ring. However, this depends on whether an interaction is necessary at all to activate the gesture recognition process.

Smartwatches and smart rings are ubiquitous [1] and blend into the environment like regular accessories, which makes them socially acceptable and therefore accessible in many environments. For example, they could be used as replacement for presenters to change slides during a talk, or more generally for performing actions depending on the context of the environment and available devices. Gesture controlled systems could also be combined with other classic input devices like keyboards to increase productivity or to create completely new interaction models. Virtual reality controllers are rather bulky and could be completely replaced [6] by multiple smart rings or a combination of watch and ring. Medical doctors could use gestures to control music playback or to switch screens in sterile environments. Speech impaired people could use a system that is as easy to use as voice control. Beyond the gesture catalogue, gestures are language independent and can be used for sign language [7] or for character recognition [8]. Elderly people having trouble with remote controls could benefit from gesture-controlled systems as well as visually impaired people [9].

Beyond, many more applications could benefit from gesture recognition systems. Especially, one-handed ubiquitous systems could replace or extend existing user interfaces and lead to completely new ways for computer interaction.

1.3. Scope and Focus

The scope of this paper encompasses gesture recognition, which is the process of detecting a specific gesture in an arbitrary sequence of datum points. Recognition is different from detection, which essentially is the process of detecting whether the data points represent a gesture or other arbitrary movements.

Data are recorded with motion sensors such as accelerometer, gyroscope and magnetometer. The gestures are performed with the hand, while sensor devices are mounted on the wrist and/or finger. The focus of the paper is on examining whether there is a difference between wrist-mounted and finger-mounted sensors when recognizing hand gestures with respect to different machine learning models. This difference is at first evaluated with equal conditions for both devices to establish a common ground. Later on, the finger device uses a different activation method to start and stop the data sampling. This different method has several advantages in terms of usability. Although the focus is on the technical approach, the different activation methods and the resulting differences in usability can influence the overall performance of the system. Therefore, the activation method should be taken into consideration as well.

The most important part of a gesture recognition system is the gestures it supports. These gestures should be universally accepted between different users and feel as intuitive as possible for specific tasks. Since the scope of the paper does not encompass specific applications for the gestures, a dictionary of gestures needs to be established, which is acceptable without respective applications in mind.

The entire machine learning process of the system is at the core of this paper and encompasses multiple steps from data recording to model training and gesture classification.

1.4. Goals and Research Focus

Card et al. [10] measured information rates of fingers and wrists, performing an empirical study on the design space of input devices. They conclude that fingers have an information rate of 40 bits/s and wrists have 25 bits/s. Although this study does not examine air-gestures, it can be concluded that in general fingers provide more information than wrists. Thus, it could be inferred that finger-mounted devices might produce better results than wrist-mounted devices for a gesture recognition task. This aspect encompasses the main research focus within this paper. Therefore, the definition of *better* in the gesture recognition context has to be defined. Based on the assumption that the amount of information correlates with the accuracy measure of a machine learning model, two systems can be ranked by comparing the accuracy of their models. However, the data recording of both sensor devices needs to be performed identically and at the same time with the same sampling rate.

The smartwatch has a display which can be used to start or stop the data sampling of the sensors. Since the smart ring has no display, the smartwatch needs to control the smart ring sensor activation in order to establish an identical recording setup. The sensors of the ring could be activated with a press of adjacent fingers, which could affect the recorded data. Assuming that the easier access to the activation of the ring sensors leads to almost perfectly segmented signals, the question arises whether the activation mechanism of the smart ring increases the accuracy—this aspect is part of future investigations.

This aspect can be evaluated by re-recording the data with a different activation method and then comparing the results with the previous recording. The goal of this paper was to create a system, which is capable of recording and recognizing gestures, aimed at evaluating the two research aspects.

2. Related Work

Starting with the first iPhone in 2007, accelerometers were considered a standard component of smartphones, with the intent to automatically rotate the screen when switching from portrait to landscape mode [2]. Today, most smart devices are equipped with sensors that make them capable of measuring movements. However, the primary goal of recent smartwatches, fitness trackers and smart rings is to recognize activity patterns and measure fitness related properties. Most market-ready products do not provide an API for accessing the built-in sensors with third-party software, which makes it difficult to acquire wearable sensors for research purposes.

Generally, for the scope of this paper, related work has to be discussed whereas it has to be differentiated between *finger-mounted systems* and *wrist-mounted systems*.

2.1. Finger-Mounted Systems

Roshandel et al. [11] present a gesture recognition system that uses a smart ring to record data. The ring consists of a 9-axis sensor, a battery and a proximity sensor to sense adjacent fingers. It transmits data via Bluetooth and does not have any mechanism to start or stop the transmission, which indicates that they performed the segmentation manually by hand. The results were evaluated by comparing four different machine learning models (multi-layer perceptron, decision tree, naive Bayes and support vector machine), using cross-validation, whereas the data were collected with 24 participants, each performing 9 different gestures. It is notable, that air-gestures perform better than gestures performed against a fixed surface—this supports the approach of this paper using air-gestures.

Xie et al. [12] propose a rule-based extensible gesture recognition system. This system is based on a smart ring with a 3-axis accelerometer that also contains a vibrator for tactile feedback. Data are transmitted via Bluetooth Low Energy with a sensor sampling rate of 50 Hz. The accelerometer records data within a range of −8 g to +8 g, which should suffice for the gesture catalog they propose. Since all gestures are defined as 2D gestures and the ring coordinate system is always aligned with the gestures, one axis is ignored, resulting in a 2D space. The gesture catalog encompasses eight basic gestures and 12 complex gestures composed of a combination of two to three basic gestures. The hamming distance is used as a measure for the similarity between two gestures. In contrast to machine learning-based approaches, the recorded data are only used to test the accuracy by counting the number of correct matches. The basic gesture matching resulted in an accuracy of 98.9% and the complex matching in 97.2%. However, some of the gestures were not detected by the segmentation algorithm, because some participants could not reach the minimum acceleration that is required for the thresholds. The advantage of this approach over machine learning approaches is that it is not necessary to train a model beforehand, which leads to an extensible system. However, it is not robust against different users and has strict constraints on the orientation of the ring and how the gestures have to be performed.

The magic ring [13] is a wired sensor that is mounted on the middle segment of the index finger. The mounting location was chosen, because the middle segment is the most flexible and should therefore deliver better results. The ring contains a 3-axis accelerometer which provides values in a range from −1.5 g to +1.5 g. This range is very small compared to most other systems, but the gesture catalog contains only very fine grained gestures. The gesture catalog consists of 12 simple gestures. Six of the gestures are targeted at finger movement while the hand is fixed and four are targeted at hand movements. The remaining two gestures represent bending and unbending movements of the finger. The data were recorded with 20 participants, each repeating every gesture five times, thus resulting in 100 samples per gesture and 1200 samples in total. Since there is no mechanism to activate or deactivate the recording between gestures, the data were segmented by using a fixed window with an empirically estimated size of one second. The results were evaluated by three different machine learning algorithms (C4.5, KNN and NB) using cross-validation. Instead of using all features, different groups of features were evaluated with every model to determine the feature quality. It is interesting that fewer features can produce better results. Therefore, a feature elimination strategy may be required for other systems as well. Relative features could be further improved and their impact on accuracy implies that experimentation may be necessary in that regard. The recognition accuracy could be further improved by using more complex models.

Zhu et al. [14] propose a system to control robots with gestures. A sensor is clipped onto the finger and connected to a PDA via a cable. The sensor provides data from an accelerometer and gyroscope with a sampling rate of 150 Hz. The five gestures are performed with the hand and resemble actions, which would be used to command dogs. Each gesture was repeated 15 times in 10 recording sessions, thus resulting in 150 samples per gesture. All of the data are recorded without any mechanism to identify the start or the end of the gesture. Therefore, segmentation is needed after the recording. The segmentation is realized with a 3-layer neural network (NN) to distinguish gestures and arbitrary movements. The NN is trained with simple statistical features, such as mean and variance. The same features are used for the gesture recognition, which is realized with a hierarchical hidden Markov model. The automatic segmentation approach using an NN seems to work very well, but it remains unclear how it performs on its own or how it affects the overall performance of the system. On a wireless real-time system, the segmentation would need to be performed directly on the sensor device, which could be problematic due to the low computing power of the device.

2.2. Wrist-Mounted Systems

Mace et al. [15] propose a gesture recognition system that utilizes a smartwatch. The watch contains a 3-axis accelerometer, which provides values in a range from −2 g to +2 g. This is a relatively small range compared to other systems. The data are sampled with 100 Hz and sent to a tablet for processing via a wireless RF-transmitter. The data were recorded from five people, each performing four different gestures. Each gesture was repeated five times, which results in 25 samples per gesture. This is a relatively small sample size compared to other systems. Nevertheless, the goal is to examine algorithms that require a small training data set. Twenty statistical features are derived from the data and evaluated with a feature weighted Naive Bayes and a Dynamic Time Warping algorithm. The results show that the NB algorithm performs best with 97% accuracy. The DTW performance is very similar with 95%.

Porzi et al. [9] propose a gesture-controlled user interface for visually impaired people. The system uses a smartwatch to recognize gestures, which instructs a smartphone to carry out various tasks. The watch contains a 3-axis accelerometer with a sampling rate of 10 Hz. This sampling rate originates from API restrictions of the smartwatch and is very low compared to other systems. The data are sent to the smartphone via Bluetooth to recognize the gestures. Eight different gestures were recorded by 15 different users, each repeating every gesture for 15 times, thus resulting in 225 samples per gesture and totaling 1800 samples. The data transmission starts when the user presses the display of the watch. The data are directly used as input for two support vector machine approaches and a dynamic time warping approach. The proposed kernel for the SVM approach is an approximated version of the global alignment kernel and has lower computational cost than the standard version. The optimal parameters for the kernels were estimated with grid search and cross-validation. The results show that the faster kernel approach delivers a slightly lower accuracy than the standard approach. The former results in an average accuracy of 91.08% and the latter in 92.33%. The DTW approach performed worse than both SVM approaches with 54.89%. The low accuracy of the DTW approach suggests that it is not suitable for a gesture recognition task without feature engineering.

Xu et al. propose a system, which is able to recognize multi-finger and hand gestures with a wrist-mounted sensor [16]. The sensor contains an accelerometer and gyroscope, which measure hand movements as well as tendon movements to recognize hand and finger gestures, respectively. The data are sampled with 128 Hz and are transmitted to a smartphone using Bluetooth. The gesture catalog consists of 37 gestures, where 13 are finger gestures, 14 are hand gestures and 10 are arm gestures. Only 10 samples were recorded per gesture, totaling in a data set with 370 gestures. During the recording, the finger, hand and arm were fixed to a chair depending on the gesture. Feature extraction is performed for various feature categories, such as motion energy, posture, motion shape and motion variation. The 10 best performing features are selected by determining the information gain. Three different classifiers (naive Bayes, linear regression and decision tree) are evaluated based on the selected features. The algorithms are only tested with a very small sample size of 10 samples per gesture and in a very controlled environment. To test the system's robustness, a much larger sample size, recorded with different users, would be necessary.

Serendipity [17] is a system to recognize fine finger gestures with a smartwatch. The smartwatch contains an accelerometer and a gyroscope and is capable of sensing rotation and gravity due to the sensor fusion between these two. The sensors are sampled with 50 Hz and the data are sent via Wi-Fi to a server for processing. Five fine-grained multi-finger gestures were recorded with 10 participants. Each participant performed each gesture 40 times. The same experiment was repeated for three different orientations and the whole experiment was repeated again, which results in 800 samples per gesture and orientation. The gesture segmentation is realized with a dynamic time warping algorithm and empirically estimated thresholds. The signal is split into windows with a width of one second. Each window is then compared to a template using DTW. Four different machine learning classifiers (support vector machine, naive Bayes, k-nearest neighbor and logistic regression) are used for validation. The results show that the average accuracy is 87%. However, data from half of the participants performed better with the SVM classifier and the other half performed better with the LR classifier. The false positive rate for the gesture segmentation is 25%, which means that every fourth time, noise is detected as a gesture. With an activation gesture instead of the automatic segmentation algorithm, the rate decreased to 8%.

2.3. Discussion

Table 1 shows an overview of the compared systems. Half of the systems use wrist-mounted sensors or smartwatches, while the other half uses finger-mounted sensors or smart rings. Every sensor device includes at least an accelerometer. Half of all systems additionally measure data with a gyroscope. However, the gesture catalog of some systems does not require a gyroscope because none of the gestures contain rotation movements. Nevertheless, some systems could benefit from an additional gyroscope.

Table 1. Overview of the compared systems.

Publication	Mounting Location	Nr. of Gestures	Nr. of Samples	Sensors	Model	Accuracy
Roshandel et al. [11]	Finger	9	120	A,G	MLP	97.80%
Xie et al. [12]	Finger	8	70	A	SM	98.90%
Jing et al. [13]	Finger	12	100	A	DT	86.90%
Zhu et al. [14]	Finger	5	150	A,G	HMM	82.30%
Mace et al. [15]	Wrist	4	25	A	DTW	95.00%
Porzi et al. [9]	Wrist	8	225	A	SVM	93.33%
Xu et al. [16]	Wrist	14	10	A,G	NB	98.57%
Wen et al. [17]	Wrist	5	800	A,G	KNN	87.00%

Generally, the accuracy ranges from 82.3% to 98.57% and leads to an average accuracy of about 92%. It is not clear from the comparison whether wrist- or finger-mounted sensors produce better results. Moreover, this cannot be directly compared because different systems use different recognition techniques and gesture catalogs. Furthermore, some systems base the performance of the models on user-independent and some base it on user-dependent experiments and facilitate different feature sets or no features at all. To compare two mounting positions, the experiments need to be identical in every possible way, so that unknown influences do not disturb the results.

All the findings of this section directly influence the methodology of this paper. In order to compare the mounting points, the experimental setup needs to be identical. Therefore, the data for both locations will be recorded in parallel, using a button-based mechanism to start and stop the gesture. The same gesture set will be used for both devices, containing only hand gestures. In order to examine all aspects of gestures, the catalog will contain gestures that are performed in a 3D space, utilizing all axes of the sensors. Both devices will use 9-axis sensors, containing an accelerometer, gyroscope and magnetometer. The latter will be ignored to prevent interference with other smart devices. To support the design aspects of a real-time system, both devices will transmit the data wirelessly to a smartphone for processing. A sampling rate of 100 Hz will be used, which should be low enough to be transmitted in real-time and high enough to capture most of the movement patterns.

For the data acquisition, 10 participants will provide 10 repetitions of 12 gestures, which results in 100 samples per gesture for each mounting location and is consistent with the gesture and sample size that is used in the proposed systems. The feature derivation process will produce different features, which will be ranked by importance and the unimportant ones will be removed. Only some models in the proposed systems, will be used for the evaluation, since the models depend on the compatibility with machine learning frameworks.

Since all systems achieved an average accuracy of 92%, the desired minimum for this paper matches this number. To examine alternative manual segmentation techniques, the whole recognition process is repeated with a different button-based mechanism.

3. Approach and Methodology

3.1. System Overview

In the course of this paper, a custom smart ring has been developed (see Figure 1) taking into account well-known challenges and aspects for wearable systems [18]. The ring consists of different modules, which can be stacked onto each other, adding certain capabilities to a microcontroller. The stacking point between modules also serves as a connector between them. All modules were bought from the company TinyCircuits [19] and are designed as extensions for the TinyDuino [19] microcontroller. Each module measures about 20 × 20 mm and all required modules stack in height to about 17 mm. The battery has the same form factor as the modules and increases the height to about 20 mm when stacked on top of the modules. The accelerometer board is as close to the bottom as possible to minimize the distance between the finger and the actual sensor. For attaching the stack of modules to the finger, a comfortable elastic ring is used.

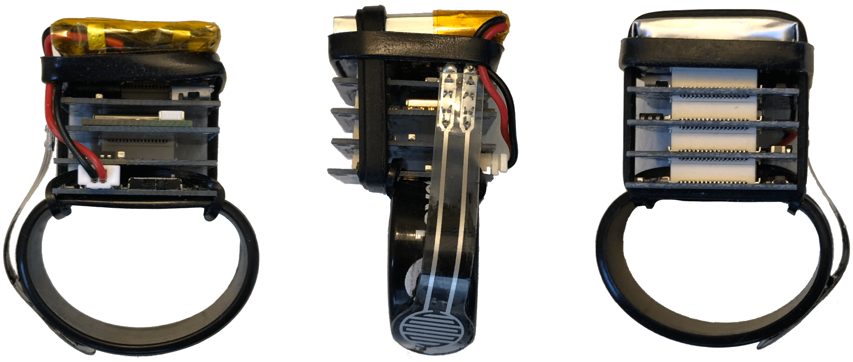
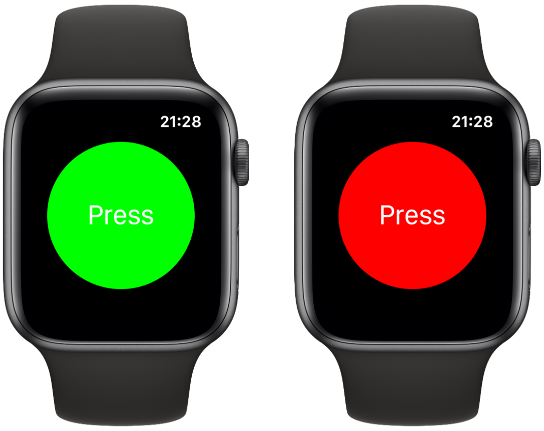

(**a**) front (**b**) side (**c**) back

Figure 1. Custom built smart ring with an elastic mounting and a force sensitive resistor on the side of the ring. From bottom to top the modules are the processor board, the sensor module, the Bluetooth module, the USB shield, the prototyping board and the battery.

In addition to the custom hardware developments, software had to be implemented for both, the smart ring as well as the smartwatch. The purpose of both implementations is mainly to read the sensor data and send it to the smartphone. However, other tasks have to be considered too, such as timing the sensor readings, enabling/disabling the sensors, locking/unlocking the activation mechanism and managing power consumption. Regarding the smart ring, an attached force sensitive resistor enables/disables the sensors, but in order to compare different activation methods, the sensors must be able to be triggered externally. Therefore, the smart ring supports remote activation of the sensors via Bluetooth. Additionally, the manual activation mechanism can be locked remotely to prevent false activations while being controlled remotely. Furthermore, the ring periodically sends status updates to the smartphone, which inform its user about the condition of the ring.

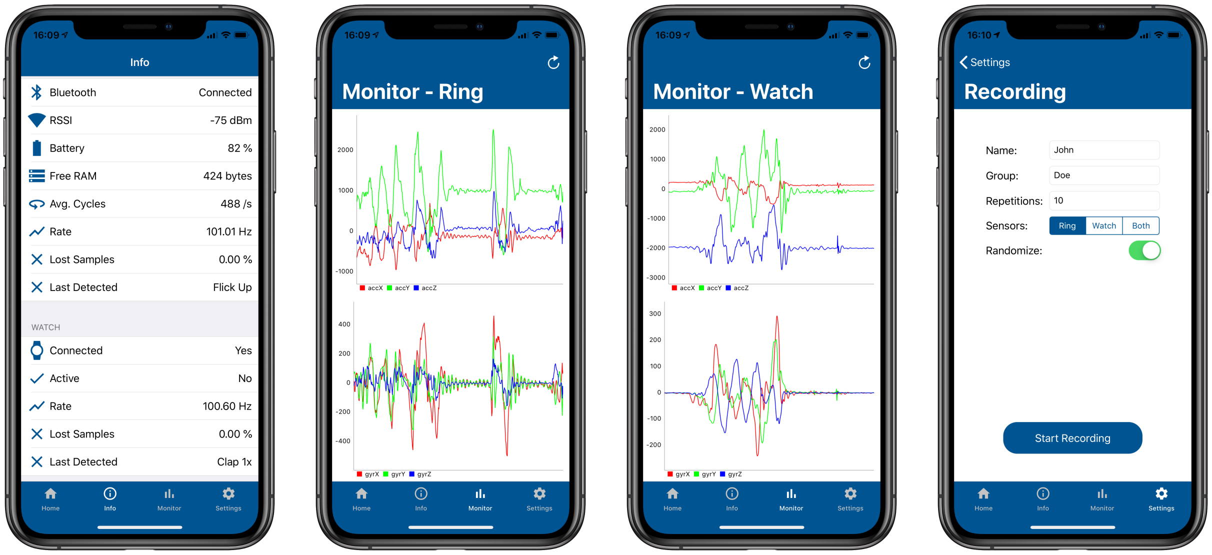
The purpose of the smartwatch application is similar to the smart ring's: read sensor data and then send them to the smartphone application. The user interface consists of only a single screen. This screen provides a software button as activation mechanism. Figure 2 shows the user interface of the smartwatch application. In contrast to the activation mechanism of the ring, it is not required to keep pressing the button after the initial press. Instead, the button changes its color to the color red, which informs the user that the sensors are enabled.



(**a**) sensors deactivated (**b**) sensors activated

Figure 2. User interface of the smartwatch. A simple button as an activation method. When the button is pressed the color changes to red and the sensors begin to transmit data until it is pressed again.

Finally, the two major responsibilities of the smartphone application are (i) recording data and (ii) recognizing gestures using this data. Moreover, it provides information about connected sensor devices and the data they are transmitting. Exemplary screens of this application can be seen in Figure 3.



Bluetooth	Connected
RSSI	-75 dBm
Battery	82 %
Free RAM	424 bytes
Avg. Cycles	488 /s
Rate	101.01 Hz
Lost Samples	0.00 %
Last Detected	Flick Up
WATCH	
Connected	Yes
Active	No
Rate	100.60 Hz
Lost Samples	0.00 %
Last Detected	Clap 1x

Name: John

Group: Doe

Repetitions: 10

Sensors: Ring Watch Both

Randomize:

(**a**) (**b**) (**c**) (**d**)

Figure 3. Info screen (**a**), ring monitor screen (**b**), the watch monitor screen (**c**) and the recording screen (**d**).

Regarding the intended real-time aspect of the proposed system, we specifically aim to meet the requirement that the system is able to compute and return the result (i.e., the recognized gesture) within a maximum delay of 1–2 s (this seems to be an appropriate time-span in terms of user interaction)—thus, speaking of a soft-real-time system (i.e., the missing of a deadline does not fail the system but degrades the usefulness of the result) [20,21]. The answer of the system could also be defined in terms of response-time, turnaround-time or recognition-time. With our approach and the chosen hardware and software configurations, we achieve this real-time requirement invariably as the results Section 4 shows. The exact details regarding the response-time of the system are out of scope of this paper but we plan to release these findings in a future article.

3.2. Relevant Gestures

Since there is no gesture catalog that is universally applicable and intuitive for different users, a set of 12 different gestures is proposed, inspired by Gheran et al. [22,23]. All considered gestures based on direction information implicitly introduce a gesture in the opposite direction. Some gestures are based on rotation movements and others combine rotation information with direction information. Moreover, the gesture catalog contains gestures that produce low as well as high acceleration values and gestures that utilize different dimensions of the sensor reference coordinate system. In order to be able to compare finger-mounted and wrist-mounted sensors, all gestures are performed using the whole hand. Since most users are right-handed and the smartwatch is (usually) worn on the left hand, the smart ring is also worn on the left hand. Therefore, all gestures are performed with the left hand.

As depicted in Figure 4, both sensor devices provide measurements inside their own reference coordinate system. Both systems are very similar in regard to the axis labeling. However, the systems' axes are not parallel due to the anatomy of the hand and might be different from user to user. The gravity vector always points along the negative Z-axis, which indicates that the participants can either stand or sit when performing gestures.

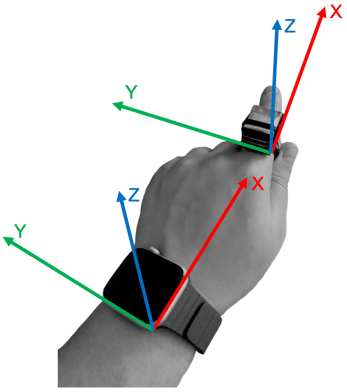


Figure 4. Reference coordinate systems of the smartwatch and the smart ring.

Figure 5 shows four directional gestures. Each gesture is performed in the YZ-plane and resembles a quick movement in one of four directions. Therefore, the acceleration on the X-axis is relatively low compared to accelerations in the plane. Due to the lack of rotation movements, the gyroscope measurements contain mainly noise.

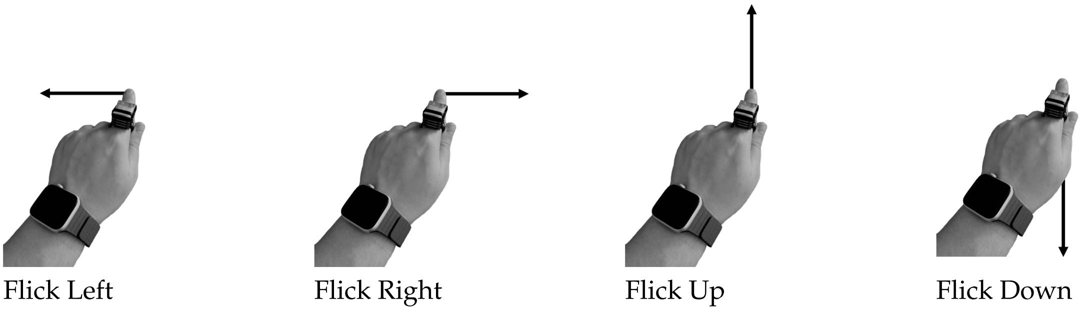
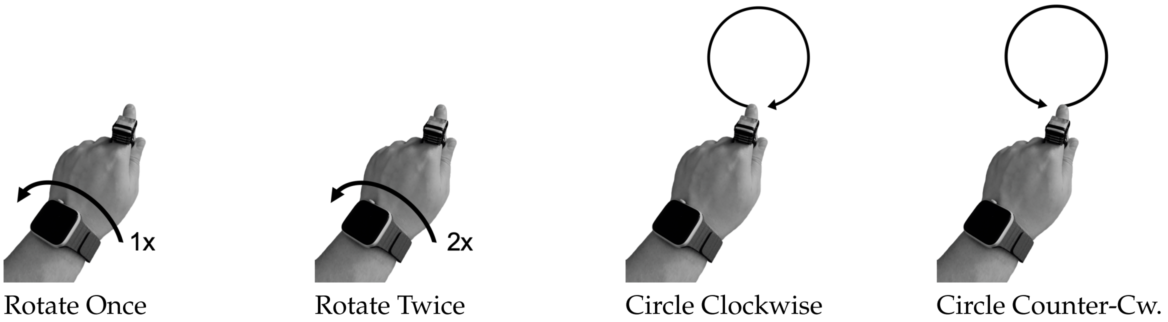


Figure 5. Gestures based on direction information in the YZ-plane.

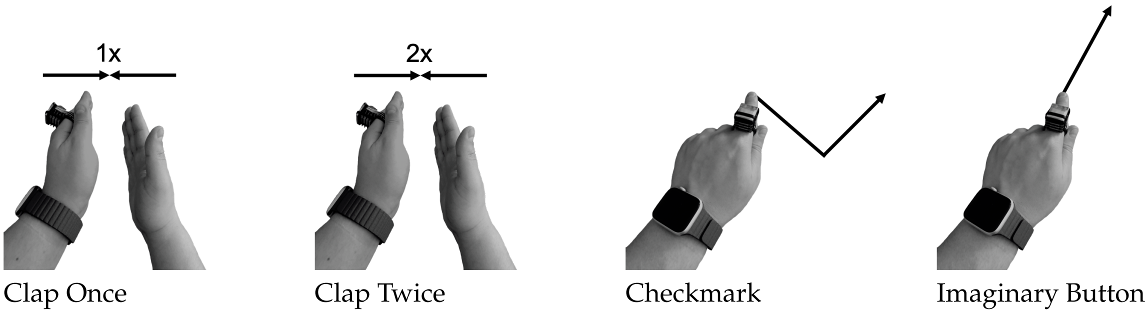
The first two gestures in Figure 6 are very similar to the wake gesture of the Apple Watch and resemble either one or two quick rotation movements around the X-axis. The delay between the rotations of the *Rotate Twice* gesture might vary between users. The other two gestures are performed in the YZ-plane and resemble a circle which is drawn into the air, either clockwise or counter-clockwise. The *Circle* gestures are the only gestures that smoothly transition between acceleration axes. The measurements of the former gestures are mainly visible on the gyroscope and the latter gestures mainly affect the accelerometer.



Rotate Once Rotate Twice Circle Clockwise Circle Counter-Cw.

Figure 6. Gestures based on rotation information around the X-axis or inside the YZ-plane.

The first two gestures of Figure 7 resemble clap movements, which lead to relatively high accelerations on the Z-axis compared to other gestures. Similar to the *Rotate Twice* gesture, the time delay between claps might be different from user to user. Furthermore, the initial rotation of the hand, which is necessary to get into a clapping position, affects the gyroscope. The *Checkmark* gesture contains movements in multiple directions inside the YZ-plane and is the only gesture that abruptly changes acceleration axes. The *Imaginary Button* resembles a gesture pressing an imaginary button in front of the user. It is the only gesture performed along the X-axis and therefore, transitions the gesture catalog from a 2D to a 3D space.



Clap Once Clap Twice Checkmark Imaginary Button

Figure 7. Gestures based on high acceleration values along the Z-axis, multiple directions in the YZ-plane or direction information along the X-axis.

It is notable to mention that all persons have performed the gestures with the whole left hand and that the devices have not been worn reversed or generally wrong for the experiments. Since the gestures are performed with the whole hand, the finger on which the ring is worn does not significantly influence the performance of the system (i.e., the coordinate system and thus the movement patterns do not change).

3.3. Algorithmic Methodology

In the following Sections 3.3.1–3.3.4, the algorithmic methodologies applied in our approach are presented. Figure 8 provides an overview about the sub-sequential steps starting from the gathering of the gesture data, to data preprocessing, feature extraction and gesture recognition.

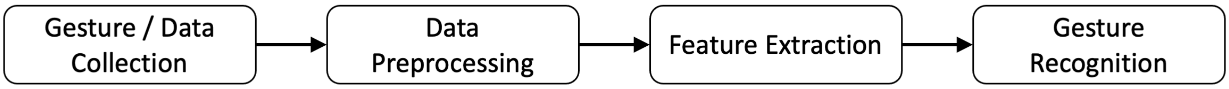


Figure 8. An overview of the different algorithmic steps applied in our approach.

3.3.1. Data Analysis and Preprocessing

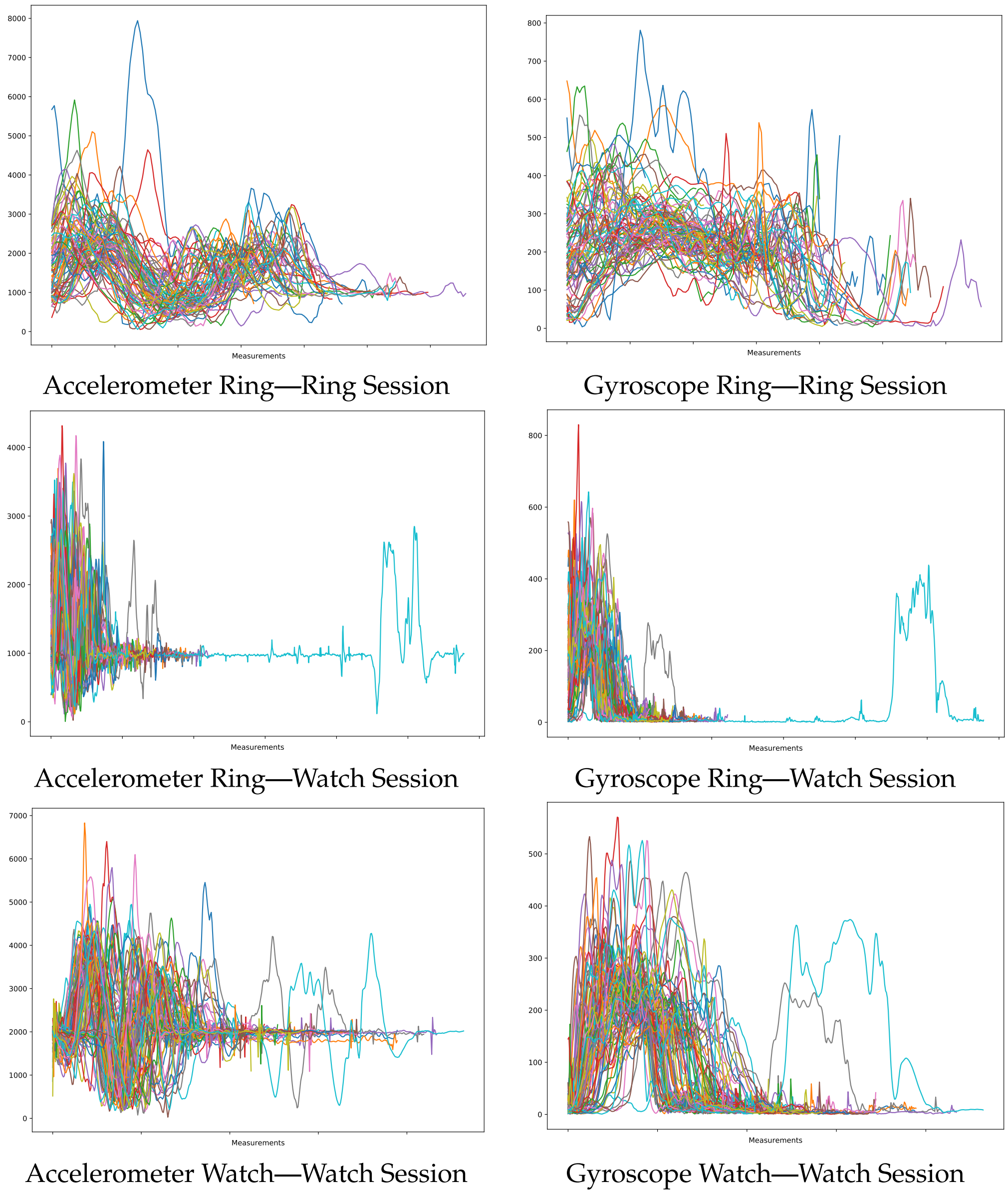
Every gesture produces different signals which can be compared to other instances of the same gesture. Multiple instances of the same gesture might look similar when visualized and others might look rather different. The data analysis stage is a tool for observing the behavior of data transformations, which are performed during the preprocessing stage. Thus, the preprocessing stage refines the data with the goal of making different instances of the same gesture more similar.

Since the gesture catalog contains 12 different gestures and every gesture contains eight signals, only the magnitudes of the *Circle Clockwise* gesture are used for the analysis. The gesture is chosen as a reference, because it utilizes the entire YZ-plane and also contains rotation movements. The magnitudes are calculated by applying the Euclidean norm for all three axes of accelerometer and gyroscope. Figure 9 shows the unprocessed magnitude data of the *Circle Clockwise* gesture for each of the three data sets originating from two sessions and is used as a reference for discussing the preprocessing algorithms. Every algorithm is applied to multiple gesture instances and must be independent from other instances, because in a real-time applications only one instance is available at a time. Figure 10 shows the results for all three data sets after applying all the algorithms.

While not visible in Figure 9, some instances contain a low number of measurements. Especially, instances of the ring session show this behavior, because the gesture can end with or without moving the hand back to the initial position, or because of technical errors originating from the remote activation. The fastest gesture contains at least 20 measurements according to empirical assumptions. Therefore, all instances that contain a lower number of measurements must be outliers and are discarded.

When comparing the watch session graphs with the ring session graphs of Figure 9, it can be observed that the watch session contains individual signals, which are relatively long compared to others. These signals result from a delayed deactivation of the sensors and might originate from technical errors or human mistakes. Therefore, they are considered outliers and need to be removed. Due to the size of the data set, a method has been developed that shortens signals with atypical lengths automatically. This method does not remove the signals but rather limit the length of a signal depending on the majority of lengths of all other signals of the same gesture type. The median of the lengths is calculated, and all signals that exceed the median value are limited to it, resulting in a collection of signals with about the same length, as shown in Figure 10.

Most instances of the gesture in Figure 9 have different maximum amplitudes, because different users prefer different scales of the same gesture. Faster gestures result in shorter signals with lower amplitudes. The signals can be normalized to a range between 0.0 and 1.0 by dividing every measurement of the signal by the maximum amplitude of the signal. This would result in gestures that share the same maximum amplitude of 1.0 across all dimensions. However, when the signal of every dimension has the same maximum amplitude, direction information is lost implicitly. To circumvent this problem, every measurement on each sensor axis is divided by the corresponding magnitude value, instead of the maximum value. Using this approach, the direction information of individual axes is not lost and all values range from 0.0 to 1.0.



Accelerometer Ring—Ring Session

Gyroscope Ring—Ring Session

Accelerometer Ring—Watch Session

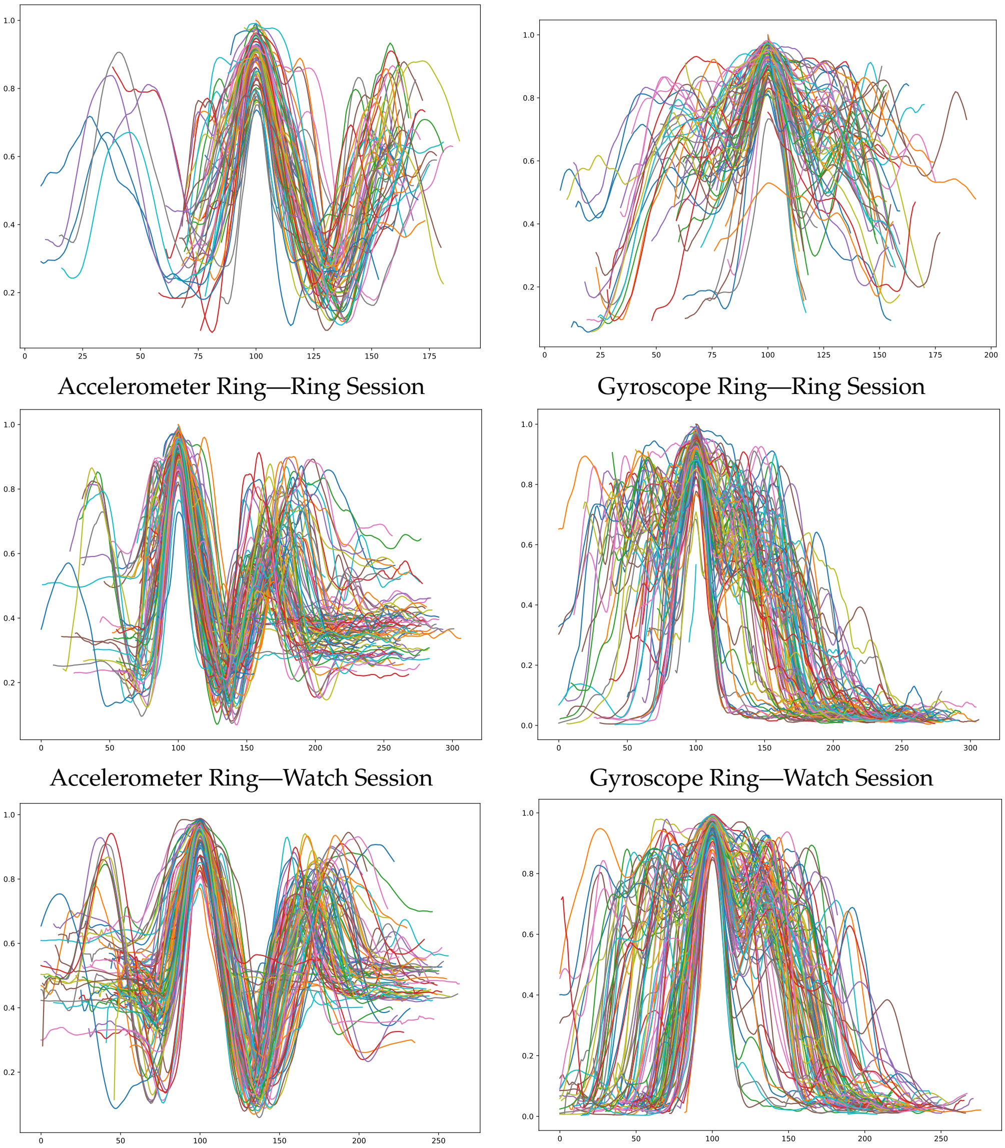
Gyroscope Ring—Watch Session

Accelerometer Watch—Watch Session

Gyroscope Watch—Watch Session

Figure 9. Unprocessed magnitude data for the *Circle Clockwise* gesture. The two columns contrapose accelerometer and gyroscope data and each row shows data originating from a different session. The data contain a few outliers and each signal is scaled differently. Additionally, some signals are shifted in time and the data contain high frequency noise.

Some instances in Figure 9 contain high frequency noise. Since humans are not able to perform movements with a high frequency, a simple smoothing filter is applied to the data. The algorithm is based on a list with an empirically estimated size of five entries. This size keeps most signal characteristics and eliminates enough high frequency noise. While iterating over the measurements, every new iteration step adds a new measurement to the list and removes the oldest measurement, if the maximum size is exceeded. The mean of measurements inside the list is calculated and replaces the current value of the iteration step. This type of filter is called a moving average filter and was proven to be effective in a gesture recognition context [12].



Accelerometer Ring—Ring Session

Gyroscope Ring—Ring Session

Accelerometer Ring—Watch Session

Gyroscope Ring—Watch Session

Accelerometer Watch—Watch Session

Gyroscope Watch—Watch Session

Figure 10. Preprocessed magnitude data for the *Circle Clockwise* gesture. The two columns contrapose accelerometer and gyroscope data and each row shows data originating from a different session. Every signal is scaled equally, and the outliers are removed. Additionally, the time shift is corrected, and the high frequency noise is filtered.

Since every gesture is recorded with an activation mechanism and since the user is free to perform the gesture at any time between the start and the stop event, some instances are shifted in time. This effect is visible in all sessions of Figure 9. In order to correct the shift, a characteristic which defines the location of the gesture in the signal must be found. Additionally, the characteristic must be independent from other instances. Different characteristics, such as the maximum value, the minimum value or one of both depending on the smaller index, were empirically evaluated. The best performing characteristic was the position of the maximum value. Thus, every instance is shifted by the index of the maximum value of the signal, leading to aligned instances as shown in Figure 10.

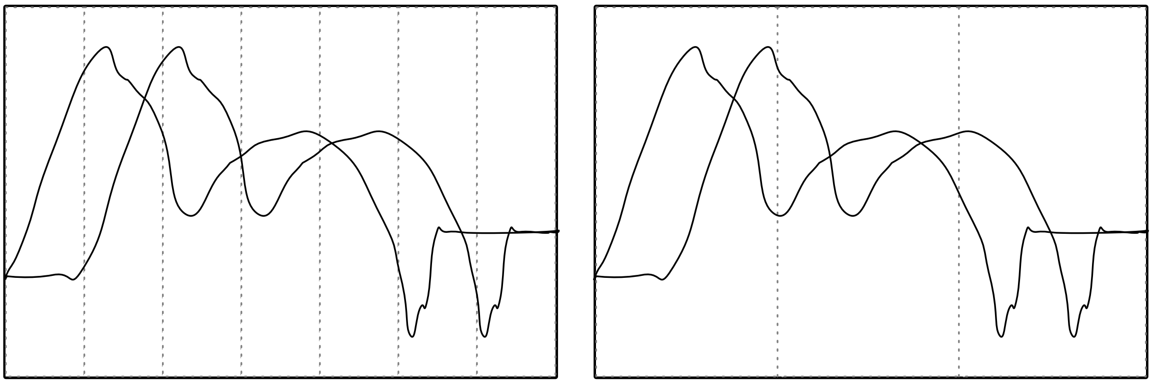
3.3.2. Feature Derivation and Selection

The goal of the feature derivation stage is to extract multiple characteristics from the data in order to differentiate gesture types. The best feature is one that uniquely identifies a gesture type and is identical on every gesture instance. However, this is not possible for the gesture catalog, since different gestures share similar characteristics. Thus, multiple features are extracted for each gesture instance and then combined with the assumption that a collection of features describes a gesture better than a single feature. Each gesture instance provides eight different signals, and each signal can be used to calculate multiple features. Therefore, the total number of features depends on the dimensionality of the preprocessed data.

Figure 10 shows that different instances of the same gesture contain a different number of measurements, even after being preprocessed. Since most machine learning models can only handle feature vectors with a fixed size, the feature derivation stage must transform signals with different lengths into feature vectors with fixed lengths. Therefore, each feature derivation algorithm must be able to handle signals with variable input sizes.

Each signal can be described by the change of its values over time, but every used feature derivation algorithm reduces the signal to a single value. Thus, a technique is required to extract time information. To solve this challenge, a generic windowing approach with dynamic sized windows is facilitated. This approach divides the signal into a fixed number of windows and then applies a feature derivation algorithm to every window, essentially producing the same feature multiple times for successive parts of the signal.

A low number of windows produces a coarse description of the signal, while a high number of windows describes the signal rather precisely. Since the goal is to create features that are very similar inside a gesture class, the number of windows must be low enough to capture the descriptive signal information in the same window. However, too few windows might oversimplify the signal and result in weak features. Figure 11 demonstrates this problem by comparing two similar signals with different window sizes. Furthermore, using windowed feature derivation algorithms with a low number of windows eliminates the need for signal centering in the preprocessing step, since the time shift is implicitly corrected by the windowing approach. The optimal number of windows per signal can be empirically determined by repeatedly training a model with different window sizes and was estimated to be five.



(**a**) small windows (**b**) large windows

Figure 11. Comparison of window sizes with two similar signals. The peak values of the signals on the left reside in different windows. In contrast, the peak values on the right reside in the same window.

In total, the following features are considered: (i) crossing (i.e., counting how often the amplitude crosses a certain threshold inside the signal), (ii) min–max (i.e., applied to each window; calculation of the min and max value inside each window), (iii) sum (i.e., sum of all values in a signal is proportional to its area), (iv) average (i.e., mean and median of measurements in each window—possibly contributes 80 features to the feature vector when five windows and eight signals are considered), (v) frequency (i.e., window approach not applied since whole signal is required; discrete Fourier transformation) and (vi) length.

In total—when all features are concatenated—the feature vector consists of 793 features. The feature vector eventually contains weak features or features that contain redundant information about the signals. Therefore, feature selection is performed to identify strong features, remove redundant features and reduce the size of the feature vector. Large feature vectors produce complex models and, depending on the algorithm, might lead to overfitting [24]. Essentially, overfitted models achieve very high prediction accuracies as long as the input data for the prediction was already used to train the model. The Mutual Information (MI) score is used to measure the relatedness and the dependency between a feature and the gesture class by quantifying the amount of information that a feature provides when observing the gesture class. The chosen MI-score algorithm is based on entropy estimation from k-nearest neighbor distances [25]. Features with a score of zero are independent from the target gesture and therefore, contribute no useful information. Whereas, a high score indicates a strong relationship with the target variable. Therefore, all features are ranked by the MI-score and only the highest ranked features are selected for the model training stage.

3.3.3. Machine Learning Models

In the course of this paper, four different machine learning models (i.e., classifiers) are considered. The selection of those specific models is grounded on the criteria that (i) the models should be rather easy to train (e.g., Kim et al. [26] show that deep learning methods need to be profoundly adapted to support real-time processing), (ii) be executable on a common smart-phone in real-time and (iii) promising for our specific task as stated by related work—the differences in the recognition performance/accuracy with respect to our research interests are also an interesting aspect and will be tackled in the results Section 4.

Each classifier can be refined with hyperparameters, which depend on the data set. Since three different data sets are compared and each data set is reduced to three categories of feature vectors, the model parameters must be estimated nine times per model type. Using four different model types, this totals in 36 iterations. However, each iteration follows the same process and is independent from other iterations. The accuracy score for evaluation of the classifiers is done by a 10-fold cross-validation (see Section 4). In detail, the following four algorithm are considered:

- Random forest (RF) [27]: based on multiple decision trees; can be tuned with various hyperparameters (e.g., number of trees, depth of the trees, no. of samples per leaf, etc.).
- Radial support vector machine (SVM) [28]: based on multi-dimensional lines (i.e., hyperplanes), which separate different classes in a multi-dimensional feature-space.
- k-nearest neighbor (KNN) [29]: based on a distance measure between a feature vector and the k nearest feature vectors (i.e., the neighbors).
- Gaussian naive Bayes (NB) [30]: based on the Bayes theorem for predicting probabilities.

3.3.4. Experiment and Data Recording

In order to gather a data set consisting of our relevant gestures with respect to a variance in the subjects performing the gestures in multiple sessions, various variables had to be considered. The experiment facilitated the established gesture catalog consisting of 12 gestures. The gestures have been performed by 10 participants, each being between 21 and 53 years old and having a technical or non–technical educational background. All participants were familiar with touch controlled smart devices.

Each gesture was performed 10 times by every participant. The resulting 120 gestures were performed sequentially by repeating the whole gesture catalog. The gestures were not randomized and therefore, the same gesture never repeated twice in succession. Half of the participants performed the gestures in a sitting posture and the other half in a standing posture. The sitting participants were not allowed to rest their elbow on a surface. Thus, all gestures were performed without restricting the movements in any way.

All gestures had to be performed with the whole hand, even if it was possible to perform the gesture with the finger while the hand was fixed. This constraint is important to be able to compare the mounting positions of the sensor devices. The duration and the scale of the gesture were not restricted, but every gesture had to be performed with a changing velocity in order to support the accelerometer readings. Every gesture had been presented to every participant before the experiment started to prevent confusion during the recording session.

The data recording was performed during two sessions. Both sessions recorded a total number of 1200 gesture instances for each targeted sensor device. The sessions were performed in succession on the same day. In total, each participant contributed a total number of 360 gesture instances to the gesture recognition system.

4. Evaluation and Results

For evaluation, three different data sets are utilized to cross-validate four different machine learning model types (random forest, support vector machine, k-nearest neighbor, naive Bayes), each with three different feature vector sizes (small, medium, large). Two of these data sets were recorded during the watch session, utilizing the data from the smart ring and the smartwatch, while sharing a common activation mechanism. The third data set was recorded during the ring session. The ring data set of the watch session is reused, and both data sets are compared, whereas both sets utilize a different activation mechanism.

4.1. Watch Session Results

The idea of the watch session is to establish a baseline for comparing the smart ring with the smartwatch. The data for both devices was recorded simultaneously, while the activation mechanism of the watch controlled both devices. Therefore, both data sets contain measurements of the exact same gestures and should differ only in terms of the mounting locations of sensor devices.

4.1.1. Smartwatch Observations

Figure 12 shows the results for 12 different models. Each model was trained with data from the smartwatch using cross-validation and compared against the held-back test set to estimate realistic accuracy scores. The held-back test was split apart from the data set in advance to the model training stage. Therefore, testing against this test set provides a realistic estimate for the model performance on unseen data. The training process was repeated for each of the three feature vector sizes, where the small feature vector contains only the most important features and the large vector misses only the weakest features. The different vectors contain 10%, 50% and 90% of the best features, respectively. For each type of model, a model representative is selected by comparing the scores originating from the three different feature vector configurations. From the four resulting representatives, the best and worst models are further inspected in detail.

The random forest (RF) representative performed best with 98.8% accuracy, while the k-nearest neighbor (KNN) representative performed the worst with 94.6% accuracy. The difference between the best and the worst representatives is only 4.4%.

The naive Bayes (NB) and the RF classifier seem to be able to handle different feature vectors about equally well considering that the maximum difference in accuracy between feature vectors sizes is only 1.1% for the RF and 2.0% for the NB. In contrast, the difference for the support vector machine (SVM) and the k-nearest neighbor (KNN) classifiers is 9.6% and 8.4%, respectively. Generally, the results show that feature vectors containing more than 10% of the best features lead to better results and that feature vectors with more than 50% of the best features lead to worse results. This suggests, that the models could be optimized further by introducing additional feature vector sizes around the medium sized category.

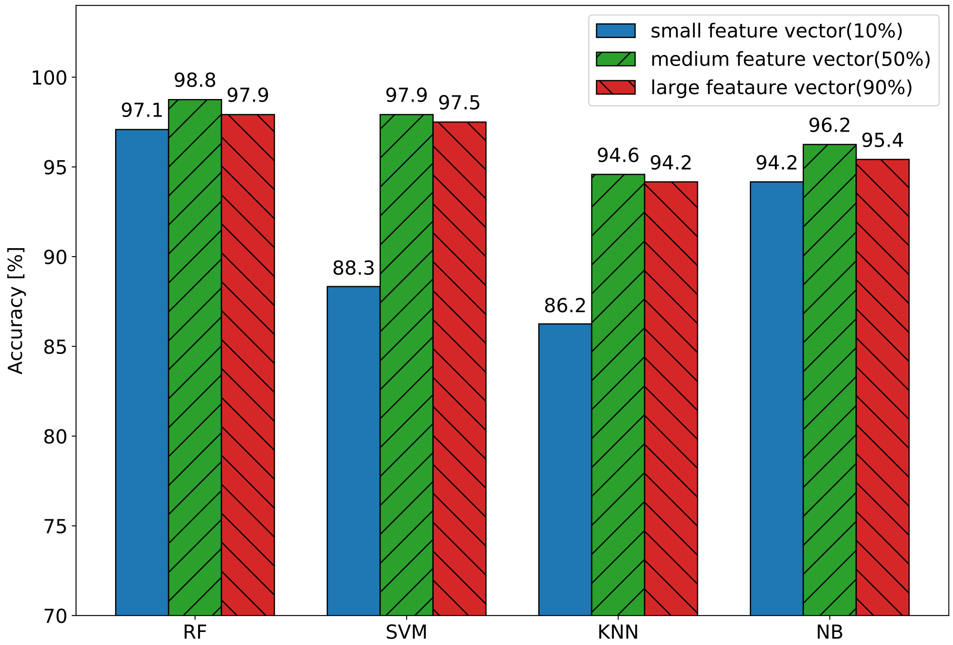
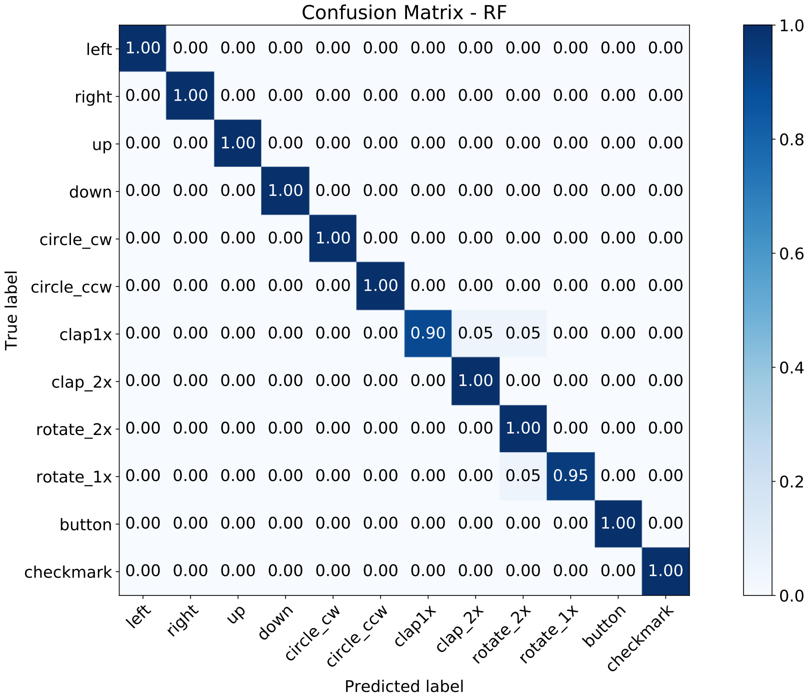


Figure 12. Accuracy scores of all 16 models trained by the watch data set of the watch session. Each model type was trained with three different feature vector sizes and tested against the held-back test data set.

The differences between the best and the worst performing model representatives can be compared in further detail with their confusion matrices. Figure 13 shows the matrix for the best performing RF model. The misclassified gestures for the RF model are the *Clap Once* and the *Rotate Once* gestures with a true positive rate of 90% and 95%, respectively. The confusion matrix suggests that the only difficulty for the RF is to decide whether the gesture is a clap or a rotation gesture and how often it is performed.

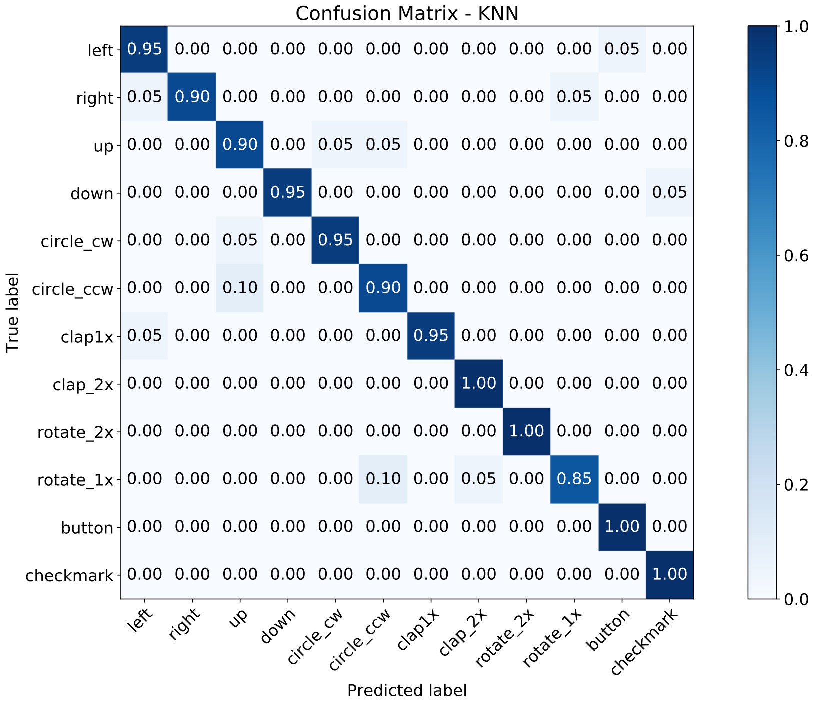


Confusion Matrix - RF

True label \ Predicted label	left	right	up	down	circle_cw	circle_ccw	clap1x	clap_2x	rotate_2x	rotate_1x	button	checkmark
left	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
right	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
up	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
down	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
circle_cw	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
circle_ccw	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
clap1x	0.00	0.00	0.00	0.00	0.00	0.00	0.90	0.05	0.05	0.00	0.00	0.00
clap_2x	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
rotate_2x	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
rotate_1x	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.95	0.00	0.00
button	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
checkmark	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Figure 13. Confusion matrix for the best performing model trained by the watch data set of the watch session. The RF model achieved an accuracy of 98.8%, while utilizing the medium sized feature vector.

The confusion matrix in Figure 14 for the best KNN model shows many misclassifications. The most misclassified gesture is the *Rotate Once* gesture with a true positive rate of 85%. Only four gestures are perfectly classified, and the remaining gestures are recognized with a true positive rate of either 90% or 95%. The distribution of the false positives suggests that the model has difficulties deciding between circle gestures and the *Flick Up* gesture.



	left	right	up	down	circle_cw	circle_ccw	clap1x	clap_2x	rotate_2x	rotate_1x	button	checkmark
left	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00
right	0.05	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00
up	0.00	0.00	0.90	0.00	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00
down	0.00	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05
circle_cw	0.00	0.00	0.05	0.00	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00
circle_ccw	0.00	0.00	0.10	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.00
clap1x	0.05	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.00
clap_2x	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
rotate_2x	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
rotate_1x	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.05	0.00	0.85	0.00	0.00
button	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
checkmark	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Figure 14. Confusion matrix for the worst performing model representative trained by the watch data set of the watch session. The KNN model achieved an accuracy of 94.6%, while utilizing the medium sized feature vector.

4.1.2. Smart Ring Observations

Figure 15 shows the results for the 12 different models that were trained with the smart ring data. The smart ring was controlled remotely by the smartwatch and the activation mechanism on the ring was temporarily disabled to ensure that the data are independent of the activation method. Each model was cross-validated with the same feature vector sizes that were used in the watch data section and tested with the held-back test set.

The model representative is chosen in the same way as in the smartwatch section, by comparing all three models of the same model type and using the highest accuracy score as a criterion. However, in this case, the scores for the SVM models are equal for the medium and large feature vector sizes. Since a lower number of features leads to lower complexity models, the model that was trained on the medium sized feature vector is chosen. The best and the worst performing model representatives are analyzed in detail by comparing their confusion matrices.

The RF representative performed the best with 96.6% accuracy while the NB representative performed the worst with a score of 88.6%. The difference between these two model representatives is 8.0%, which is rather significant. The KNN and NB model representatives are very close to each other with 89% accuracy and 88.6% accuracy, respectively. However, both are unable to satisfy the defined minimum requirement of 92% accuracy.

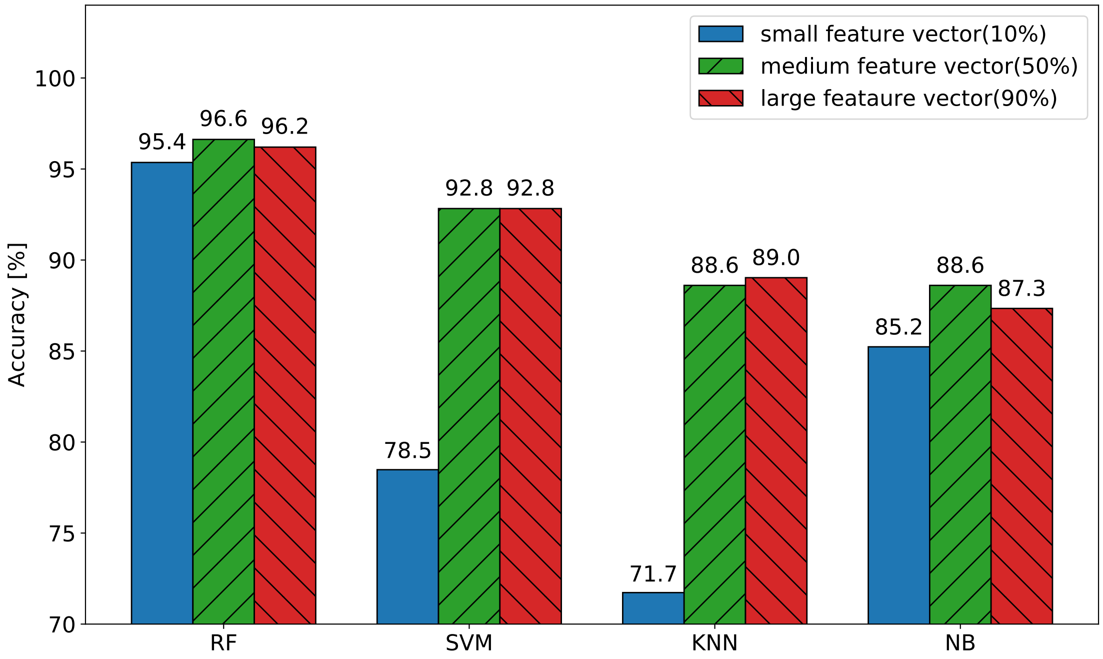
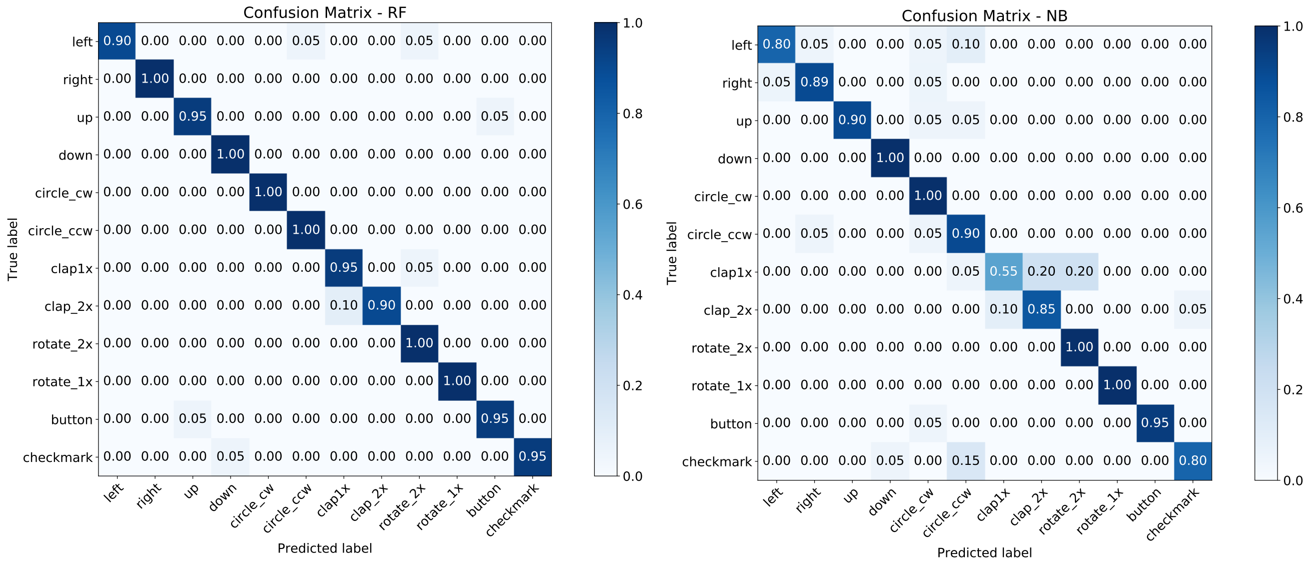


Figure 15. Accuracy scores of all 16 models trained by the ring data set of the watch session. Each model type was trained with three different feature vector sizes and tested against the held-back test data set.

The RF and the NB classifiers seem to be able to handle different feature vector sizes well compared to the SVM and KNN classifiers. The difference between the worst model and the best model for different feature vector sizes is 1.2% for the RF classifier and 3.5% for the NB classifier. In contrast, the difference for the KNN and SVM classifiers is 17.3% and 14.3% respectively. However, these differences originate from the low score for the small feature vector while the scores for larger vectors are almost identical. This suggests, that these classifiers require at least a certain number of features in order to perform good on the data set. The KNN model representative is the only model that gains an advantage by using the large feature vector over the medium vector.

The confusion matrix in Figure 16a shows the individual classifications for the best performing model. Most of the gestures are perfectly classified, hence, the high accuracy of the model. The worst performing gestures are the *Clap Twice* and the *Flick Left* gestures with a true positive rate of 90%. The *Clap Twice* gesture is confused with the *Clap Once* gesture with a false positive rate of 10% which is reconcilable due to the similarity between them.

Figure 16b shows the confusion matrix for the worst performing model representative. Similar to the RF model, the worst performing gesture for the NB model is the *Clap Once* gesture with a true positive rate of 55%. It is frequently confused with the *Clap Twice* and the *Rotate Twice* gesture. Due to the low overall accuracy, several gestures are misclassified with true positive rates ranging from 80% to 95%. The distribution in the matrix suggests that the NB model has problems differentiating directional gestures from circle gestures.



(**a**) (**b**)

Figure 16. Confusion matrices illustrating results. (**a**) Confusion matrix for the best performing model trained by the ring data set of the watch session. The random forest (RF) model achieved an accuracy of 96.6%, while utilizing the medium sized feature vector. (**b**) Confusion matrix for the worst performing model representative trained by the ring data set of the watch session. The naive Bayes (NB) model achieved an accuracy of 88.6%, while utilizing the medium sized feature vector.

4.2. Ring Session Results

The data for the ring session was recorded only by the smart ring, while utilizing the force sensitive resistor as the activation method. The advantage of this method is that the ring can be activated with just one hand by using a finger that is adjacent to the ring. Another usability difference is that the button is pressed for the duration of the gesture and it can be released to finish the gesture at an earlier time for convenience reasons.

Smart Ring Observations

The data from the ring session are evaluated the same way as the data from the watch session. Figure 17 shows the 12 models that were cross-validated with different feature vector sizes and tested against the held-back test set. For each model type, a model representative is selected, with the highest accuracy score as the criterion. Since two of the SVM models and two of the KNN models share the same score, the model based on the smaller feature vector is preferred. The best and worst performing representatives are compared in further detail utilizing their confusion matrix.

The RF representative performed the best with 93.2% accuracy, while the NB representative performed the worst with 85.9% accuracy. The two model representatives differ in 7.3%, which is consistent with the representatives from the watch session. The representative for the SVM classifier is close to the best model with 92.2% and both satisfy the defined minimum requirement of 92% accuracy.

Only the RF classifier is able to handle the small feature vector without a significant loss in accuracy. The difference between the worst and best performing models of the RF classifier is only 1.5%, while the KNN classifier shows a difference of 9.9%. This makes the RF the most flexible classifier in regard to the feature size when compared with the results of the watch session, since it is the only classifier that consistently shows this size resistant behavior. The SVM and KNN classifiers show the same scores for medium and large vectors and could be further refined by adding new feature vectors with additional sizes between 50% and 90%.

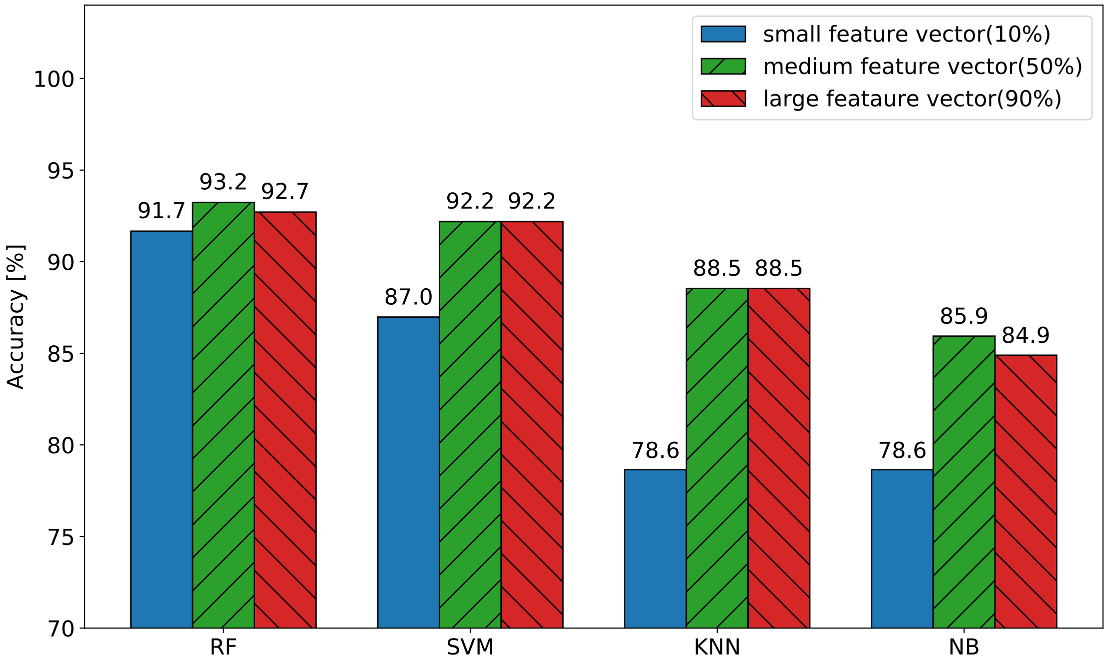
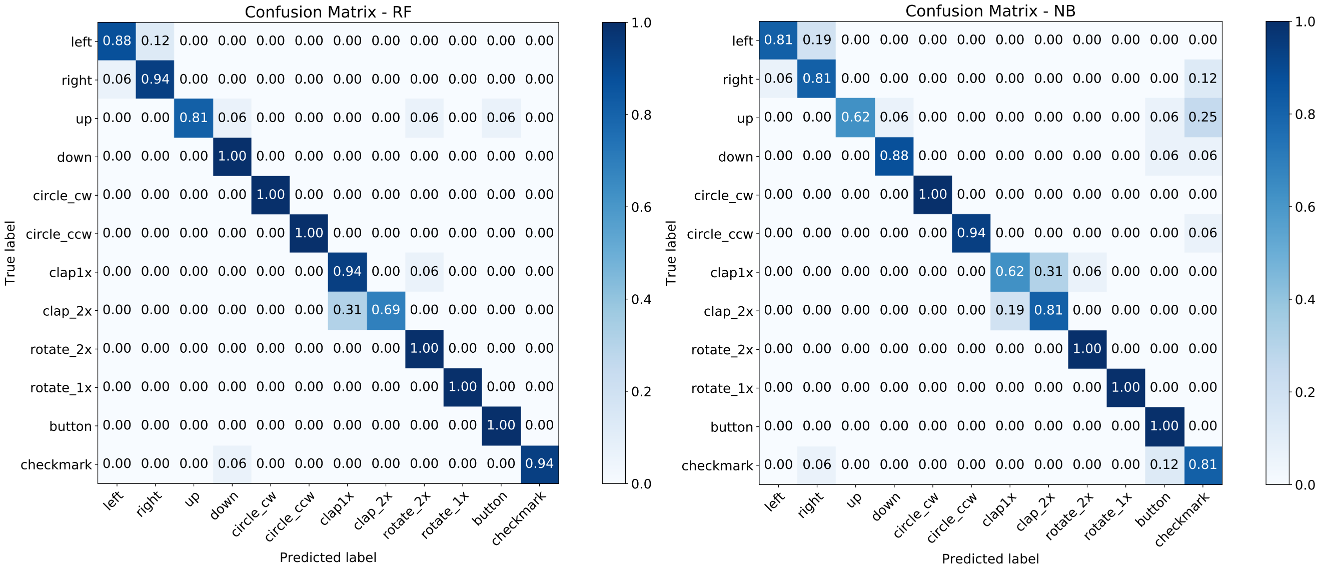


Figure 17. Accuracy scores of all 16 models trained by the ring data set of the ring session. Each model type was trained with three different feature vector sizes and tested against the held-back test data set.

The confusion matrix for the best RF model in Figure 18a shows the *Clap Twice* gesture as the top misclassified gesture with a true positive rate of 69%. However, it is confused only with the *Clap Once* gesture, resulting in a false-positive rate of 31%. Half of the gestures are perfectly classified. The remaining gestures range between true positive rates of 81% and 95%. The distribution of the matrix suggests that the directional gestures *Flick Left* and *Flick Right* as well as the *Flick Up* and *Flick Down* gestures are frequently confused with each other.



(**a**) (**b**)

Figure 18. Confusion matrices illustrating results. (**a**) Confusion matrix for the best performing model trained by the ring data set of the ring session. The RF model achieved an accuracy of 93.2%, while utilizing the medium sized feature vector. (**b**) Confusion matrix for the worst performing model representative trained by the ring data set of the ring session. The NB model achieved an accuracy of 85.9%, while utilizing the medium sized feature vector.

Figure 18b shows the confusion matrix for the model representative of the NB classifier. More than half of all gestures do not reach a true positive rate higher than 88%, hence, the low overall score. One of the worst performing gestures is the *Clap Once* gesture with a true positive rate of 62%, which is mostly confused with the *Clap Twice* gesture. It shares the same true positive rate with the *Flick Up* gesture, which is mostly confused with the *Checkmark* gesture, which in turn is commonly confused with other directional gestures as well. The *Flick Left* and *Flick Right* gestures are also confused with each other.

4.3. Evaluation

In this section, results are evaluated by comparing the observations from two different data sets. The activation mechanism–independent evaluation targets the data sets of the watch session and the activation mechanism–dependent evaluation compares both ring data sets originating from different sessions.

4.3.1. Activation Mechanism–Independent

Both data sets of the watch session achieve a relatively high accuracy with 98.8% for the watch data set and 96.6% for the ring data set. Overall, the watch data set produces models with higher accuracies. This suggests that finger movements produce data with more variance, which is reasonable, because finger movements are more flexible than hand movements. The finger can be moved relative to the wrist when performing a gesture, adding varying information to the data. Since the same gestures are used, the varying information of the finger data is distributed inside the gesture class, ultimately leading to more variance inside the class. Additionally, during the observation of the data acquisition, it was evident that the finger position changed slightly over time when participants focused solely on the display of the watch. This changes the reference frame of the gestures, which adds even more variance to the data. On the contrary, the position of the watch never changed, because it was always corrected implicitly by pressing the button on the display.

Another explanation could be that the data originating from the Apple watch are preprocessed internally, which is likely, because the separation of the gravity and the linear acceleration vectors needed to be reversed in order to mirror the data of the ring. To separate the data in the first place, Apple's algorithm needs to preprocess the data, which could affect the data set. However, since the data are preprocessed again by a coarse smoothing filter and most of the features are windowed, it is unlikely that Apple's preprocessing algorithm has a significant impact on the quality of the data set.

When comparing the confusion matrices of the RF models for both data sets, the only similarity is the confusion of the *Clap Once* and *Clap Twice* gestures. However, labeling errors can be ruled out completely, because the gestures were recorded and labeled simultaneously. Generally, it is evident that the clap gestures are more difficult to distinguish across all models. The explanation for this observation could be that the time between the claps is variable, however, this does not affect the rotation gestures, which are also based on a varying time interval. A more reasonable explanation emerges when examining the preprocessed data manually. The moving average smoothing filter distorts the data because the claps create high frequency signals with high amplitudes, which the primitive filter cannot handle very well.

4.3.2. Activation Mechanism–Dependent

The data set of the watch session achieves an accuracy of 96.6%, while the ring session results show a maximum accuracy of 93.2%. Generally, all models of the ring session are slightly weaker compared to the watch session results. This suggests that the activation mechanism introduces additional variance to the data set.

Since the participants were allowed to choose an ending point for the directional gestures, some gestures of the same type might contain a different amount of information. The ring session results clearly show confusion between the directional gestures *Flick Left* and *Flick Right* as well as for the gesture *Flick Up*. These gestures implicitly contain gestures in the opposite direction when the ending point of the gesture is equal to the starting point. This is the case because the hand is moved back to the origin after the gesture is performed. Depending on the chosen ending point, this might introduce ambiguity between the directional gestures.

However, the most significant difference between the sessions is the confusion between the *Clap Once* and *Clap Twice* clap gestures. The reason for this confusion can be found in the user feedback. Most participants reported that the mechanism is not very well suited for these two gestures, because every clap could move the thumb slightly, which in turn could lead to deactivation of the sensors by releasing the activation mechanism unwillingly. An early deactivation for the *Clap Twice* gesture results in a gesture that is very similar to the *Clap Once* gesture. The deactivation cannot be observed visually due to the location of the sensor and therefore, cannot be corrected. Thus, a *Clap Once* gesture might be labeled unintentionally as a *Clap Twice* gesture, which explains the high false positive rate of 31%.

4.4. Summary

All selected models exceed the defined minimum accuracy of 92%. Therefore, the overall result is satisfactory in a sense that both devices and activation mechanisms are capable of sensing gestures. Figure 19 shows the results for the best model representatives of each session. It is evident that the RF classifier is the best classifier for all three data sets with an accuracy ranging from 98.8% to 93.2%. The second-best classifier is the SVM classifier, but both are relatively complex model types when compared to KNN and NB.

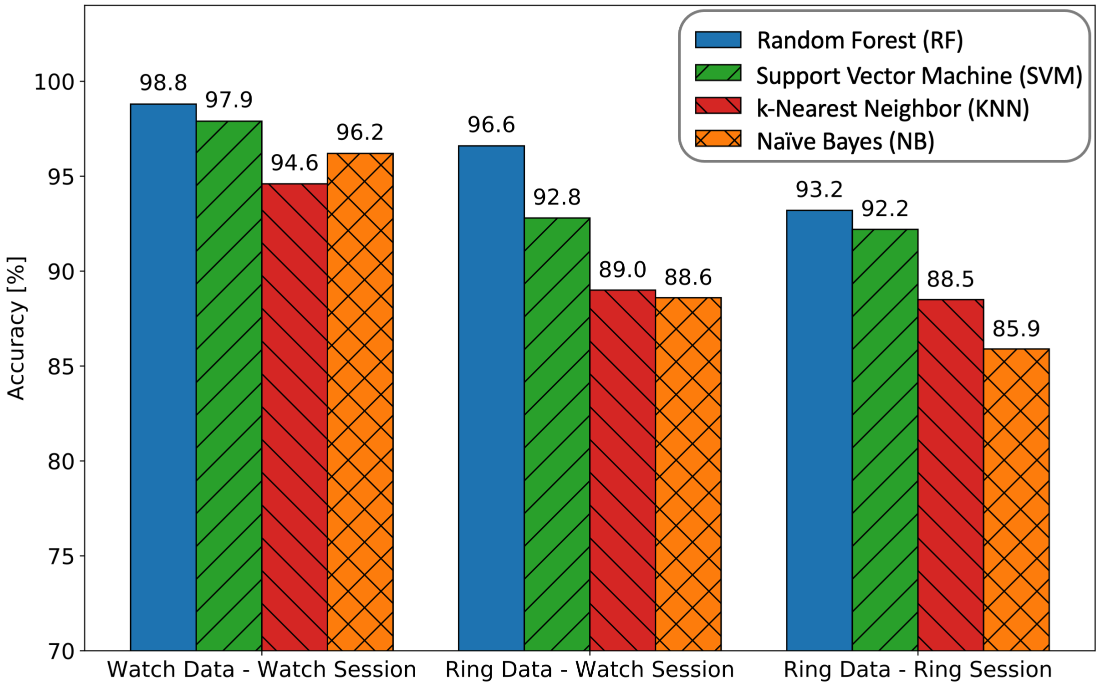


Figure 19. Best model representatives for every model type and all three data sets. The RF classifier performs best in all cases and achieves higher accuracy scores than the defined minimum of 92%.

The results suggest that all data sets contain non-linear related features, since the stronger models are known to handle non-linear relationships. The RF classifier is based on decision trees and the SVM classifier uses a radial kernel for non-linearity. Whereas, the KNN classifier is based on a linear distance measure and the NB classifier assumes independent features.

It can be observed that the data set originating from the smart ring has more variance, leading to weaker models with lower accuracy. The different activation mechanism of the ring additionally introduces ambiguities due to the varying ending points of selected gestures, resulting in even lower results. A common problem across all data sets is the clap gestures, which are confused frequently with other gestures due to the applied primitive preprocessing filter or an unintended deactivation of the sensors.

5. Summary and Outlook

5.1. Conclusions

In this article, a real-time gesture recognition system based on wearable motion sensors was built. The system is compatible with a custom-built smart ring as well as a smartwatch, both sending motion data to a smartphone, which is able to predict gestures with an accuracy of up to 98.8%. The foundations for building such a system and the associated challenges were examined. Related research covering similar challenges was organized and the relevant solutions were incorporated into the design of the system. Specifically, a minimum accuracy score of 92% was derived as a requirement for the machine learning process, various algorithms for processing the data were incorporated and the combination of accelerometer and gyroscope was selected to capture hand movements. The results show that the finger produces more information, which is consistent with the observations of Card et al. [10]. The information influences the data set by introducing more variance inside gesture classes, leading to machine models with lower accuracy. Additionally, the results clearly show that the activation mechanism affects the accuracy of the models. Since the mechanism allows for more freedom with respect to the ending point of the gesture, some gestures contain highly variant information, which decreases the model accuracy further. Additionally, the mechanism is not suited well for clap gestures, because the thumb is moved unwillingly when performing claps. This leads to an early deactivation of the sensors and thus, the *Clap Once* and *Clap Twice* gestures are confused frequently.

The system considers 12 different gestures, which are performed with the hand. They were recorded beforehand with the support of 10 participants. Each participant recorded three differently data sets during two sessions. Two data sets have been recorded simultaneously, utilizing the sensors of the smart ring and the smartwatch, while sharing the smartwatch display as an activation mechanism to start and stop the sensors. The third data set has been recorded solely with the smart ring while utilizing a force sensitive resistor as an activation mechanism.

Multiple features describing the signal were derived from each data set and sorted by importance using the Mutual Information score. Three different sized feature vectors were generated using this score and facilitated to train four different types of machine learning models. After an exhaustive parameter search, the best model for each data set was selected.

Nonetheless, every data set resulted in a model that exceeds the defined minimum score of 92% accuracy. Therefore, the proposed system is able to compete with the results of related research.

5.2. Outlook

The paper shows that an energy efficient gesture recognition system can be built into market-ready wearable devices. The activation mechanism is necessary to achieve energy efficiency but does not affect the gesture recognition accuracy significantly. This inspires to improve the system by optimizing the mechanism or by enhancing the data processing stages. The usability could be further improved by recording more data for rotation independence and posture independence, so that users are able to perform gestures when they are lying on a couch or sitting at a desk. In addition, the smartwatch as a sensor device could be improved by automatically starting the sensors when the device is woken up. This would eliminate the need to press the button, resulting in a fully one-handed experience.

The system could also be trained to recognize gestures performed with both hands, utilizing a second smart ring or a combination of smart rings and a smartwatch. This could replace controllers in virtual and augmented reality applications and lead to completely new ways for ubiquitous computer interaction.

Additionally, some supplementary research aspects are worth following in future work: (i) examining whether a significant variation of inter- and intra-person gesture recognition can be determined, and (ii) evaluation of the usability of the activation methods used within this paper. Furthermore, the authors plan to release the gathered data set including gesture labels for the scientific community in the near future.

Author Contributions: Conceptualization, R.G., M.K.; software, R.G.; project administration, M.K.; supervision, M.K.; writing—original draft preparation, R.G.; writing—review and editing, M.K., E.S. 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.

Acknowledgments: This paper and all corresponding results, developments and achievements are a result of the Research Group for Secure and Intelligent Mobile Systems (SIMS).

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

References

1. Weiser, M. The computer for the 21st Century. *IEEE Pervasive Comput.* **2002**, *1*, 19–25. [CrossRef]
2. Honan, M. Apple Unveils iPhone. *Macworld*, 9 January 2007.
3. Kurz, M.; Hölzl, G.; Ferscha, A. Dynamic adaptation of opportunistic sensor configurations for continuous and accurate activity recognition. In Proceedings of the The Fourth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE2012), Nice, France, 22–27 July 2012.
4. Roggen, D.; Calatroni, A.; Rossi, M.; Holleczek, T.; Förster, K.; Tröster, G.; Lukowicz, P.; Bannach, D.; Pirkl, G.; Wagner, F.; et al. Walk-through the OPPORTUNITY dataset for activity recognition in sensor rich environments. In Proceedings of the 8th International Conference on Pervasive Computing (Pervasive 2010), Helsinki, Finland, 17–20 May 2010.
5. Roggen, D.; Calatroni, A.; Rossi, M.; Holleczek, T.; Förster, K.; Tröster, G.; Lukowicz, P.; Bannach, D.; Pirkl, G.; Ferscha, A.; et al. Collecting complex activity datasets in highly rich networked sensor environments. In Proceedings of the 2010 Seventh International Conference on Networked Sensing Systems (INSS), Kassel, Germany, 15–18 June 2010; pp. 233–240.
6. Rupprecht, F.A.; Ebert, A.; Schneider, A.; Hamann, B. Virtual Reality Meets Smartwatch: Intuitive, Natural, and Multi-Modal Interaction. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*; ACM: New York, NY, USA, 2017; pp. 2884–2890. [CrossRef]
7. Bui, T.D.; Nguyen, L.T. Recognizing postures in vietnamese sign language with MEMS accelerometers. *IEEE Sens. J.* **2007**, *7*, 707–712. [CrossRef]
8. Zhou, S.; Dong, Z.; Li, W.J.; Kwong, C.P. Hand-written character recognition using MEMS motion sensing technology. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*; AIM: Hong Kong, China, 2008; pp. 1418–1423. [CrossRef]
9. Porzi, L.; Messelodi, S.; Modena, C.M.; Ricci, E. A smart watch-based gesture recognition system for assisting people with visual impairments. In *Proceedings of the 3rd ACM International Workshop on Interactive Multimedia on Mobile & Portable Devices*; ACM: New York, NY, USA, 2013; pp. 19–24. [CrossRef]
10. Card, S.K.; Mackinlay, J.D.; Robertson, G.G. A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst.* **2002**, *9*, 99–122. [CrossRef]
11. Roshandel, M.; Munjal, A.; Moghadam, P.; Tajik, S.; Ketabdar, H. Multi-sensor based gestures recognition with a smart finger ring. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Kurosu, M., Ed.; Springer International Publishing: Cham, Switzerland, 2014; Volume 8511, pp. 316–324. [CrossRef]
12. Xie, R.; Sun, X.; Xia, X.; Cao, J. Similarity matching-based extensible hand gesture recognition. *IEEE Sens. J.* **2015**, *15*, 3475–3483. [CrossRef]
13. Jing, L.; Zhou, Y.; Cheng, Z.; Wang, J. A recognition method for one-stroke finger gestures using a MEMS 3d accelerometer. *IEICE Trans. Inf. Syst.* **2011**, *E94-D*, 1062–1072. [CrossRef]
14. Zhu, C.; Sheng, W. Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living. *IEEE Trans. Syst. Man Cybernet. Part A Syst. Hum.* **2011**, *41*, 569–573. [CrossRef]
15. Mace, D.; Gao, W.; Coskun, A. Accelerometer-based hand gesture recognition using feature weighted naïve bayesian classifiers and dynamic time warping. In *Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion*; ACM: New York, NY, USA, 2013; p. 83. [CrossRef]
16. Xu, C.; Pathak, P.H.; Mohapatra, P. Finger-writing with Smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications—HotMobile '15*; ACM: New York, NY, USA, 2015; pp. 9–14.

17. Wen, H.; Ramos Rojas, J.; Dey, A.K. Serendipity: Finger Gesture Recognition Using an Off-the-Shelf Smartwatch. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 7–12 May 2016; pp. 3847–3851. [CrossRef]
18. Seneviratne, S.; Hu, Y.; Nguyen, T.; Lan, G.; Khalifa, S.; Thilakarathna, K.; Hassan, M.; Seneviratne, A. A survey of wearable devices and challenges. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2573–2620. [CrossRef]
19. TinyCircuits. Product Page. Available online: https://tinycircuits.com (accessed on 24 February 2021).
20. Kopetz, H. *Real-Time Systems: Design Principles for Distributed Embedded Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
21. Shin, K.G.; Ramanathan, P. Real-time computing: a new discipline of computer science and engineering. *Proc. IEEE* **1994**, *82*, 6–24. [CrossRef]
22. Gheran, B.F.; Vanderdonckt, J.; Vatavu, R.D. Gestures for Smart Rings: Empirical Results, Insights, and Design Implications. In *Proceedings of the 2018 Designing Interactive Systems Conference*; ACM: New York, NY, USA, 2018; pp. 623–635. [CrossRef]
23. Gheran, B.F.; Vatavu, R.D.; Vanderdonckt, J. Ring x2: Designing Gestures for Smart Rings Using Temporal Calculus. In *Proceedings of the 2018 ACM Conference Companion Publication on Designing Interactive Systems*; ACM: New York, NY, USA, 2018; pp. 117–122. [CrossRef]
24. Kodratoff, Y. *Introduction to Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
25. Kozachenko, L.F.; Leonenko, N.N. Sample estimate of the entropy of a random vector. *Probl. Peredachi Inf.* **1987**, *23*, 9–16.
26. Kim, M.; Cho, J.; Lee, S.; Jung, Y. IMU sensor-based hand gesture recognition for human-machine interfaces. *Sensors* **2019**, *19*, 3827. [CrossRef] [PubMed]
27. Liaw, A.; Wiener, M. Classification and Regression by randomForest. *R News* **2002**, *2*, 18–22. [CrossRef]
28. Burges, C.J. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.* **1998**, *167*, 121–167. [CrossRef]
29. Cunningham, P.; Delany, S.J. k-Nearest Neighbour Classifiers. *Mul. Classif. Syst.* **2007**, *34*, 1–17. [CrossRef]
30. Rish, I. An empirical study of the naive Bayes classifier. In Proceedings of the International Joint Conferences on Artificial Intelligence 2001 Workshop on Empirical Methods in Artificial Intelligence, Seattle, WA, USA, 4–10 August 2001; Volume 3, pp. 41–46.