

Dealing With Inaccurate Sensor Data in the Context of Mobile Crowdsensing and mHealth

Robin Kraft , Fabian Hofmann , Manfred Reichert , and Rüdiger Pryss 

Abstract—The technological capabilities and ubiquity of smart mobile devices favor the combined utilization of Ecological Momentary Assessments (EMA) and Mobile Crowdsensing (MCS). In the healthcare domain, this combination particularly enables the collection of ecologically valid and longitudinal data. Furthermore, the context in which these data are collected can be captured through the use of smartphone sensors as well as externally connected sensors. The TrackYourTinnitus (TYT) mobile platform uses these concepts to collect the user's individual subjective perception of tinnitus as well as an objective environmental sound level. However, the sound level data in the TYT database are subject to several possible sensor errors and therefore do not allow a meaningful interpretation in terms of correlation with tinnitus symptoms. To this end, a data-centric approach based on Principal Component Analysis (PCA) is proposed in this paper to cleanse MCS mHealth data sets from erroneous sensor data. To further improve the approach, additional information (i.e., responses to the EMA questionnaire) is considered in the PCA and a prior check for constant values is performed. To demonstrate the practical feasibility of the approach, in addition to TYT data, where it is generally unknown which sensor measurements are actually erroneous, a simulation with generated data was designed and performed to evaluate the performance of the approach with different parameters based on different quality metrics. The results obtained show that the approach is able to detect an average of 29.02% of the errors, with an average false-positive rate of 14.11%, yielding an overall error reduction of 22.74%.

Index Terms—Crowdsensing, environmental sound, mHealth, noise measurement, sensor anomaly detection, tinnitus.

I. INTRODUCTION

SMART mobile devices, especially smartphones, are becoming increasingly ubiquitous. This ubiquity, as well as

Manuscript received 22 April 2022; revised 30 June 2022; accepted 9 August 2022. Date of publication 11 August 2022; date of current version 7 November 2022. (Corresponding author: Robin Kraft.)

Robin Kraft is with the Institute of Databases and Information Systems, Ulm University, 89081 Ulm, Germany, and also with the Department of Clinical Psychology and Psychotherapy, Ulm University, 89081 Ulm, Germany (e-mail: robin.kraft@uni-ulm.de).

Fabian Hofmann and Manfred Reichert are with the Institute of Databases and Information Systems, Ulm University, 89081 Ulm, Germany (e-mail: fabian-1.hofmann@uni-ulm.de; manfred.reichert@uni-ulm.de).

Rüdiger Pryss is with the Institute of Clinical Epidemiology and Biometry, University of Würzburg, 97080 Würzburg, Germany (e-mail: ruedi.pryss@uni-wuerzburg.de).

Digital Object Identifier 10.1109/JBHI.2022.3198254

the technical capabilities of these devices, predestines them for two concepts: Ecological Momentary Assessments (EMA) and Mobile Crowdsensing (MCS). In health care, a combination of these concepts can be used to collect ecologically valid and qualitative longitudinal data while simultaneously capturing the context in which these data are collected by using the sensors of smartphones and connected devices (e.g., wearables) [1]. For example, environmental data (e.g., noise [2], [3]) could be measured while a user is completing an EMA questionnaire. In this way, questionnaire data can be correlated with environmental data to provide new insights about users (e.g., patients). However, to provide meaningful information, these sensor measurements must meet several aspects of data quality, such as *accuracy*, *objectivity*, *believability*, and *reputation* [4]. For non-standardized smartphone sensors such as the microphone (i.e., different manufacturers, different mobile operating systems, different scales), achieving these properties can be particularly challenging.

The mobile platform *TrackYourTinnitus* (TYT) uses EMA and MCS to track a user's individual tinnitus. Tinnitus is the perception of a sound in the ears in the absence of a corresponding external acoustic stimulus. Tinnitus symptoms are subjective and vary over time. Therefore, TYT was created to monitor and assess the variability of these symptoms in the daily lives of tinnitus patients or interested users [5]. The platform has been running since 2014 and consists of a website for general information and user registration (<https://www.trackyourtinnitus.org/>), a mobile app available for both iOS and Android, and a central backend for data storage. Users of the mobile app are asked to complete EMA questionnaires at different times of the day to assess their individual tinnitus perception (e.g., tinnitus loudness and distress) [6]. In addition, the mobile app records environmental sound levels while completing the daily questionnaire [6]. The process of TYT mobile app [1] as well as the underlying data set, i.e., the structure and findings about the collected data [7], have been described in detail in previous work.

The overall goal of this paper is to investigate the relationships between environmental sound levels and reported tinnitus symptoms. More specifically, we want to investigate whether environmental sound levels have an effect on tinnitus. If the objective sound levels can be related to the subjective data collected in the EMA questionnaires, new insights could emerge, allowing, for example, predictions of tinnitus loudness based on the environmental sound data. In this context, it should also be noted that for tinnitus and many other diseases and disorders, longitudinal studies capable of collecting ecologically valid data over longer

periods of time (more than half a decade for TYT) are still very rare. However, the sound data available in the TYT database [7] are not suitable for direct comparison with each other and thus for meaningful interpretation in terms of correlation with tinnitus symptoms [8]. This is because the environmental sound levels were retrieved from many different device models from different manufacturers and thus potentially many different microphones and other audio components. In addition, the sound level is stored as a relative amplitude value (Android) or uncalibrated decibel value (iOS) as retrieved from the mobile system APIs, as opposed to a comparable *Sound Pressure Level* (SPL) or weighted decibel value (e.g., dB(A)). In addition, sound level values are stored as-is, and no error detection or plausibility checks are performed during or after measurements, which can result in many erroneous values (e.g., zero values) being stored in the database. Furthermore, the mobile app does not indicate on the user interface that the sound measurement is running. Therefore, the user could interfere with the measurement (e.g., if the microphone is covered or the smartphone collides with an object) and generate erroneous values again. These erroneous sensor readings are a common and recognized problem in sensor network applications [9]. To address these problems, we take several approaches. On the one hand, we focus on smartphone calibration, app adaptations to retrieve more meaningful values, and in-app error detection mechanisms [8]. On the other hand, we are working on approaches to cleanse the existing data in the TYT database from a *data perspective*. This work proposes such a *data-centric* approach that aims to detect sensor errors in mHealth mobile crowdsensing data sets and subsequently exclude erroneous sensor values from all further analyses.

Based on the characteristics of the given environment, the following requirements for the approach were defined:

- 1) The approach should be able to detect and identify sensor faults in mHealth data sets.
- 2) The approach should be able to handle data sets for which it is not known a priori which values are affected by sensor faults.
- 3) The approach should be computationally efficient, so that it is possible to apply it to very large data sets.

The main challenge for the algorithm in the proposed approach is that it is generally not known which sensor measurements are erroneous and which are not. For this reason, a simulation using generated data with similar characteristics to the TYT data set was designed and run to assess the performance of the algorithm with different parameters based on different quality metrics. Furthermore, since little information is available about the sound measurement itself that would allow detection of sensor errors, the algorithm takes into account additional information (i.e., responses to the EMA questionnaire). The core of the proposed approach is a *Principal Component Analysis* (PCA) complemented by a constant value check. In addition, since conventional PCAs are not able to handle missing values, an upstream procedure is used to impute data.

The remainder of this paper is organized as follows. Section II considers related work. Section III presents in detail the materials and methods used for the proposed approach. Section IV presents the results of the simulation and the application of the

approach to the TYT data set. A discussion of the results in terms of the practical relevance and limitations of the approach is given in Section V, while Section VI concludes the paper with a summary and an outlook on future work.

II. RELATED WORK

Much of the previous work on anomaly detection for sensor signals has been done in the field of *Wireless Sensor Networks* (WSNs). Teh et al. [10] identified two main types of solutions in this context: statistical methods (e.g., *Principal Component Analysis* (PCA) [11], [12]) and classification-based methods (e.g., *Artificial Neural Networks* (ANNs) [13]). For WSNs, *Statistical Process Control* (SPC) [14] approaches are mostly used for the surveillance of chemical processes [10], for which scenarios with highly correlated data from distributed sensors are prevalent. PCA analyses, in turn, are based on multivariate detection mechanisms, such as *Squared Prediction Errors* (SPEs). *Body Sensor Networks* (BSNs) [15], as derivatives of WSNs, make use of sensors applied to patients in form of implantable sensors or wearable devices. Note that for both WSNs and BSN, all of the above-mentioned analyses techniques are commonly applied as they generate multivariate data. In contrast to this, univariate data can be also analyzed, for example, by the use of *Contribution Plots* (CPs) [14] instead of SPEs.

Classification-based techniques rely on a set of pre-labeled training data that allow to build a prediction model for further detection. Safaei et al. [15] stated that ANNs are one of the most popular methods to efficiently identify outliers. During a training phase, a classifier uses pre-labeled data to learn separating outliers from *normal* data. This approach of combining neurons in a dynamic network in order to describe logical problems [13] allows the development and use of very accurate and fast detection algorithms. However, [16] and [17] name the difficulties in defining the most suitable ANN parameters that enable correct error detection. These parameters include, for example, the type of neural network, the number of hidden layers, and their number of neurons. In addition, training data sets have to be available and precise to create a useful detection mechanism. The authors developed an approach that is suitable for automated fault detection, but due to challenges mentioned above, it is only able to detect 50% of the defined error types. The authors also pointed out a wide gap in the explainability of given input and the retrieved results. In the mHealth context, the large variety of devices with unknown technical peculiarities, creates an additional challenge for this classification. Therefore, apart of the very dynamic possibilities of this approach, we did not consider it suitable for our purposes.

Error detection approaches that need no prior knowledge and can be applied to data of arbitrary form can be found in the field of statistics. PCA is recognized as one of the most popular [10] tools for sensor error detection in the WSN context. [18] proposes the usage of robust PCA in combination with SPE to detect multivariate anomalies within a set of physiological measurements. For this purpose, a central computational device gathers measurements of distributed sensors and normalizes data during pre-processing. In a first step, the dimensional size of the

data is reduced. This is followed by SPE to detect multivariate anomalies by comparing it with an adaptive threshold [19]. The authors apply CPs to each physiological value afterwards in order to determine the contribution of each value to the detected anomaly. Finally, the PCA model is updated if the parameters change.

To validate their approach, [18] compared different (online and offline) PCA-implementations in a case study with regard to their performance. The authors found different robust offline PCA-implementations to perform much better than the classic implementation. They investigated the degree of variability that retains after dimension reduction for a number of l principal components: $l = 3$ describes 98.2%, compared to $l = 4$ for 94.9% or $l = 5$ for 99.8%. This reflects in smaller proportions of unexplained variability and gives better dimension reductions.

Miller et al. [14] developed a quality metric that addresses the question of the origin of an observed fault in SPC. In this context, a variable that can be computed with values from the PCA model is given by the *Hotelling's T^2 score*, which we will use as a control variable in the approach of the work at hand. In contrast to classification-based approaches, PCA-based approaches in combination with the mechanisms mentioned above seem to fit better into our application context. This is not only due to certain similarities of WSNs and mHealth applications, but also due to the very sound methodology and the independence from prior knowledge about the correctness of the data.

Other efficient outlier detection approaches were created, for example, by Saneja and Rani [20]. The authors propose three different phases that are scalable for Big Data applications to identify anomalies in highly correlated parameters. Initially, a correlation coefficient matrix is constructed. Then, correlating sensors are used for dynamic predictions using linear *support vector machines*, *sliding windows* and *dynamic thresholds*. The last stage is constituted by a parallel computing phase. Aderibigbe and Chi [21] propose a mechanism to identify temporal outliers, using *median absolute deviation*. After exceeding a static threshold, the *majority voting* allows the detection of false-positives. Due to dynamically changing conditions, dynamic thresholds in particular have a big validity for sound measurements. A Big Data approach may be out of scope for this initial evaluation of methods, but might nevertheless find application in future developments of our approach.

[20] and other statistical-based approaches in the eHealth or mHealth context rely on physiological databases such as PhysioNet¹ to evaluate their performance. PhysioNet (formerly Physiobank²) or MHEALTH³ consist of measurements that were gathered within various studies and contains arbitrary anomalies. To better control performance, synthetic data generation is required. Although there exist several different simulators [22], most of them are strongly tied to WSNs and tend to focus on validating their protocols. Approaches like [23] therefore manually select a number of 10,000 data points they consider error-free.

¹[Online]. Available: <https://physionet.org> (Accessed: 2022-03-14).

²[Online]. Available: <https://archive.physionet.org/physiobank/> (Accessed: 2022-03-14).

³[Online]. Available: <http://archive.ics.uci.edu/ml/datasets/mhealth+dataset> (Accessed: 2022-03-14).

Afterwards, the authors inject errors at random, specified by their probability. In our approach, we also apply random errors of defined form to the data set. However, due to the large effort of manual data selection, the approach in the work at hand instead relies entirely on synthetic data.

In summary, none of the aforementioned approaches is able to handle a univariate data set with possibly missing data points, which is the case for TrackYourTinnitus. The contribution of the work at hand is therefore to provide such an outlier detection approach and, in addition, to provide a simulation tool to evaluate the performance of this approach with different parameters suitable for our TYT mHealth data purposes. This is in line with [24], as the authors have stated that quality of imputed data is essential and can therefore influence medical conclusions drawn from it. Our proposed approach is validated against a generated data set that contains realistic anomalies, allowing these anomalies to be tracked and thus fine-granular performance measurements to be made. The aim of this synthesis is to provide the possibility of dealing with verified error-free data points and errors of defined shape. We therefore hope to be able to define algorithm parameters in a more accurate way than other approaches.

III. MATERIALS AND METHODS

In the following, the materials and methods used in the present work are described. In this context, the data set to which the approach is to be applied is outlined in more detail. Furthermore, the chosen approach, including the different algorithm phases, is explained in detail.

A. Materials

The analyzed data set was extracted from the TYT database on the 23 rd of December, 2021. It contains a total of 107,231 entries. The structure of the TYT database was described in [7]. The mobile apps capture the individual tinnitus perception by asking users to fill out tinnitus assessment EMA questionnaires at random times of the day. The process was described in detail in [1]. Each entry in the database contains the answers to the eight questions of the TYT EMA questionnaire [25], along with a timestamp and the user agent that contributed the answers. In addition, a sound level value is stored in each entry that is retrieved by measuring the environmental sound level during the first 15 seconds of the user completing the EMA questionnaire by utilizing the microphone of the mobile device [6]. This value is denoted as *sensor value* in the following, while the user agents that contributed the EMA answers and sensor measurements are denoted as *sensors*.

B. Methods

As depicted in **Table I**, sensors are considered at a certain level of abstraction. Each of the sensors s is treated as a black box and provides measurement data in nominal values. These values are identified by $[0, 1] \cup \{\perp\}$, where \perp describes an erroneous value. The measurements of the sensors are further referred to as measurements m .

TABLE I

VARIABLES THAT ARE USED THROUGHOUT THE DEVELOPMENT AND IMPLEMENTATION OF THE ANOMALY DETECTION ALGORITHM

Variable	Description
s	number of sensors
p	scalar EMA answers (properties)
m	number of sensor measurements (samples)
k	kNN imputation number of neighbors
l	number of components for PCA
ϵ	threshold to detect constant values
δ	T^2 control limit

Mapping a sensor value x into the range of $[0, 1]$ is done by applying the mapping function $T(x) = \frac{x-a}{b-a}$, with $[a, b]$ being $[0, 1]$. As mentioned above, the TYT data set contains not only sensor data, but also EMA answers. Since these answers are either of binary form or retrieved from visual analogue scales (VAS), they can easily be mapped to the above range and are therefore completely uniform. Each sensor s has m measurements as well as m properties p . Each device is characterized by $s \geq 1$ sensors and $p \geq 0$ properties that are associated with it. In addition, it was assumed that the clocks of the devices are synchronized and thus the time distances between different measurements are approximately the same. Although TYT data was collected at arbitrary points of time, the algorithm outcome should not be influenced by this fact.

PCA was chosen as core of the proposed approach because it is a well-established statistical method for detecting sensor errors in the context of WSNs (as outlined in Section II), which have similar conditions to (mHealth) mobile crowdsensing applications. Since constant value readings, as a severe form of sensor fault that is quite common in the TYT data set (see Section III-B3), could not be reliably eliminated by the PCA in initial tests, the PCA step was complemented by a constant check based on the approach proposed by [9].

1) **Algorithm Overview:** The overall process for the developed algorithm is illustrated in Fig. 1. In a first step, sensor measurements and EMA properties are treated separately. The algorithm takes ④ sensors and ③ properties as an input and returns ⑩ a set of marked sensors. The actual implementation handles devices, as properties are treated alongside with sensor measurements, due to their uniform character.

Due to prerequisites of the algorithm main stage, the elimination stage, ① pre-processing was defined as a necessary intermediate stage. It includes ⑤ data selection as many data points were considered unusable, e.g., due to too few sensor measurements. As the PCA-based algorithm is not able to handle missing values, a ⑥ data imputation step was added. This is done by applying a *k-Nearest Neighbor (kNN)* interpolation for missing data points.

Since constant values are a complete failure form of sensor faults [26], a ⑦ constant check forms the first step in the ② elimination stage. Afterwards, the proposed ⑧ PCA-based approach is performed, using Hotelling's T^2 score and *Contribution Plots (CPs)*. The algorithm finishes with an ⑩ output of all erroneous and correct sensors. In the following, the different steps are described in detail.

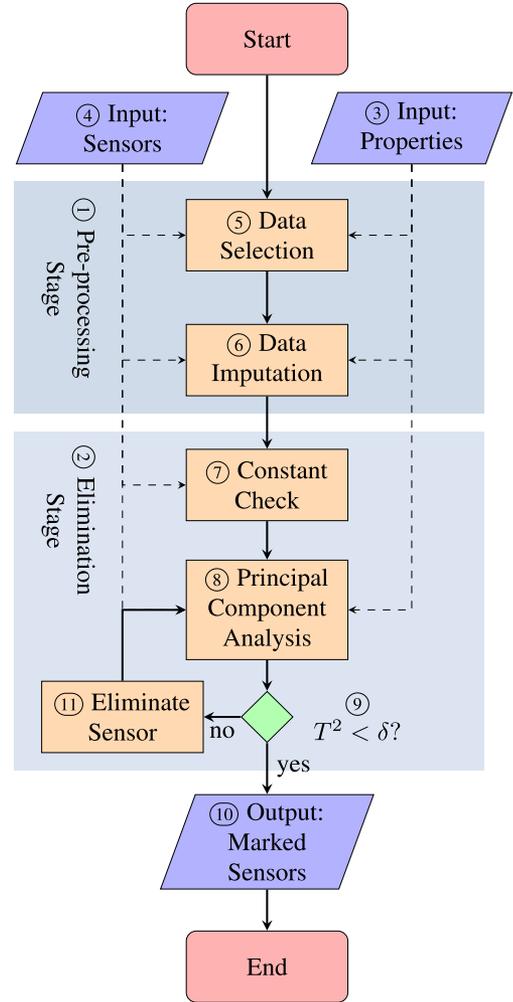


Fig. 1. Sensor anomaly elimination process that includes a pre-processing stage ① and an elimination stage ②. The algorithm output ⑩ includes all sensors, where faulty ones are marked.

2) **Algorithm Stages:** As a common language for the domains of scientific computing and data science, the implementation of the proposed architecture from Fig. 1 was done in *Python*. As a general convention, in- and output files as well as configurations use the *JSON* file format. Libraries *SCIKIT-LEARN*⁴ and *NUMPY*⁵ provided some helpful routines, for instance *SCIKIT*'s PCA, or different numeric algorithms from the *NUMPY* library.

The following describes the development of Algorithm 1 as proposed in Section III-B1.

⑤ **Data Selection:** Prior to the actual analysis of data points, a set of appropriate devices has to be chosen. Real-world data sometimes poses too few measurements per device. This had to be considered during algorithm development and led to an exclusion of devices with $m < 30$ measurements. This filtering, taking place prior to Algorithm 1, will not be specified in detail here.

⁴[Online]. Available: <https://scikit-learn.org/stable/> (Accessed: 2022-03-01).

⁵[Online]. Available: <https://numpy.org> (Accessed: 2022-03-01).

Algorithm 1: Pseudo Code of the Elimination Process of Fig. 1. A Description of the Used Variables can be Found in Table I.

```

1: function IDENTIFYFAULTS(devices, m, k, l,  $\epsilon$ ,  $\delta$ )
2:   IMPUTE(devices, k)
3:   for device in devices do
4:     if SD(SMOOTHEN(device.sensor.data)) <  $\epsilon$  then
5:       device.sensor.filtered  $\leftarrow$  true
6:     end if
7:   end for
8:   t = 1
9:   while t  $\leq$  m do
10:    X  $\leftarrow$  BUILDDATAMATRIX(devices)
11:    (P, T,  $\lambda_1, \dots, \lambda_l$ )  $\leftarrow$  PCA(X, l)
12:    D  $\leftarrow$  diag{ $\lambda_1, \dots, \lambda_l$ }
13:     $T^2 \leftarrow x_t P D^{-1} P^T x_t^T$ 
14:    if  $T^2 < \delta$  then
15:      break
16:    end if
17:    c  $\leftarrow$   $\frac{P}{T^2}$ 
18:    i  $\leftarrow$  argmaxj |cj|
19:    deviceIndex  $\leftarrow$  DEVICEINDEX(devices, i)
20:    devices[deviceIndex].sensors.filtered  $\leftarrow$  true
21:    t = t + 1
22:  end while
23: end function

```

⑥ *Missing Data Imputation:* Handling erroneous sensor measurements by completely removing them and applying interpolation methods could not bring the desired success. It either was not clear if the remaining data points could still be analyzed in a significant way, or interpolation polynomials were of too high order, unnecessarily increasing complexity. As the main error detection stage ⑧ is not able to handle missing data, the imputation method of choice is stated by a *k-Nearest Neighbor (KNN)* [27] interpolation. The implementation, as seen in Line 2 of Algorithm 1, replaces missing values with a weighted average, calculated from *k* nearest neighbors in respect of their distance to the missing data point. It is calculated using the Euclidean Norm of two points $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_n)$, which is defined as:

$$D(A, B) = (\sum_{i=1}^n |x_i - y_i|^2)^{\frac{1}{2}}. \quad (1)$$

In order to handle missing values in the neighbor data set, a method was chosen as proposed by Dixon [28]: Throughout the calculation of Euclidean Distance, empty values are summed up as zero values. Afterwards, the result is weighted, using the number of missing values. This step concludes the ① pre-processing stage, resulting in data that fits further analyses.

⑦ *Constant Check:* This first step in the elimination stage was designed to overcome constant value errors by utilizing a simple constant check [9]. Realized in Lines 3–7 of Algorithm 1, the standard deviation for a set of sensor measurements is calculated and compared to a threshold ϵ . For standard deviations below this threshold, values are assumed to be constant and their belonging

sensors are marked with a flag. Before standard deviations can be calculated, a linear Kalman filter [29] is applied. This noise reduction technique allows the estimation of proper sensor values in real time, and serves for the reduction of falsely detected constant values.

⑧ *PCA-stage:* The underlying formal description was provided by Teh et al. [10]: Let $X \in \mathbb{R}^{m \times n}$ be a matrix, consisting of *m* rows and *n* columns. Columns *n* include all measurements of a sensor at a given time stamp. Rows *m* contain the measurements of all sensors at a time frame. A loadings matrix *P* contains the *l* first eigenvectors $[\lambda_1, \dots, \lambda_l]$ of the covariance matrix $X^T X$. Prior to this, one needs to choose the number of principal components to keep: $l \leq \min\{m, n\}$. For $l < n$, the dimension of the data set can be reduced. Principal components can be obtained in a score matrix *T* with the linear transformation $T = X P \in \mathbb{R}^{m \times n}$, where *P* is defined as:

$$P = \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_l \\ | & | & & | \end{pmatrix}, \quad (2)$$

consisting of the above-mentioned eigenvectors. This formal description is implemented in Lines 10–13 from Algorithm 1. Samples, containing measured sensor values and the corresponding properties, are used in chronological order to build the data matrix *X* (Line 10). Line 11 takes matrix *X* as well as the algorithm parameter *l*, which describes the number of PCA components. Next, it is described how various variables can be derived from a calculated PCA model, which are later combined in the T^2 score. Different from the original BESI algorithm, only a single PCA model has to be calculated per iteration, compared to $n - 1$ for *n* sensors.

As the described complexity reduction does not proceed lossless, Hotelling's T^2 score (Line 13) acts as a control variable [14]. Originally, the BESI-algorithm includes Q-statistics for this purpose. However, as [30] describes, this approach is far too sensitive for our purpose of an mHealth application. Compared to delicate chemical processes, which were the original field of application for this algorithm, T^2 score states a sufficient method for error detection. Let $x \in \mathbb{R}^{1 \times n}$ be a vector representing a sample. Hotelling's T^2 score is defined as:

$$T^2(x) = x P D^{-1} P^T x^T, \quad (3)$$

with $D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_l)$, as diagonal matrix of the sorted eigenvalues of the covariance matrix.

The underneath error detection mechanism relies on the idea of iteratively ① eliminating faulty sensors if ⑨ T^2 exceeds *delta*, until all faulty sensors and their belonging properties are eliminated. As described in Lines 14–16, Hotelling's T^2 score is used as a measure for abnormal behavior. For $T^2 < \delta$, a sensor is considered working correctly, no further processing is necessary and the next sensor value is analyzed.

If the limit was exceeded, Lines 17 and 18 describe the calculation of sensor contribution to the overall fault. Detecting the highest contribution is done by solving the linear equation $P c = T_t^T$ and finding the maximum argument. Afterwards, the sensor is marked as faulty in Lines 19 and 20.

TABLE II
CONCRETE VALUES THAT WERE CHOSEN FOR THE VARIABLES FROM
TABLE I, FOR THE SIMULATION

Variable	Value
s	1
p	8
m	30
k	3
l	7
ϵ	0.12
δ	8.5

Algorithm 1 takes an array of potentially erroneous devices, containing their sensors and measurements as well as properties p . For statistical reasons, non-analyzable devices are removed from the data set prior to the application of the algorithm. Sensors that are detected faulty throughout the detection process are only marked erroneous, not eliminated. Therefore, the $\textcircled{4}$ output allows a distinction between 1) erroneous sensors and 2) those considered being correct. Additionally, 3) sensors that were excluded from the analysis can be obtained. According to different use cases, these three groups might be combined for analyses.

3) Simulation: As mentioned in Section II, a common source of data to validate the error detection approach is stated by physiological databases in combination with manual selection. To overcome limitations of not being able to generate physiological data synthetically as well as injecting errors of defined form, the following paragraph describes our proposed simulation approach.

A first requirement of the developed simulation is to generate a data set which can be used as input for Algorithm 1. This includes the generation of a defined number of devices, their sensors, measurements and properties. Devices should be identical concerning their number of generated values. As second vital requirement, errors should be generated and injected according to a configuration. For validation purposes, meta data of erroneous sensors should be available throughout the whole simulation and validation process. In addition, the simulation should be portable, running on most major operating systems, and in order to create reproducible results, it should be deterministic. As one main assumption of Algorithm 1 is that there exists a correlation between sensor measurements and their properties, these values have to be generated under this assumption.

The input parameters taken by the algorithm were used to fine-tune the accuracy of the developed approach. The variables are shown in Table II and were chosen to be optimal with respect to the selected quality metrics. To this end, 1) the parameters k , l , ϵ , and δ were varied based on a binary search process, 2) 50 simulation runs were performed with each set of parameters, 3) the results were recorded, and 4) the quality metrics (see below) were calculated. Finally, the parameters with the optimal quality metrics and the corresponding results were used for the further analyses.

The implementation of the developed simulation was created in *Python*, utilizing *SCIKIT-LEARN*⁴ and *NUMPY*⁵. The whole simulator is configurable and can be used to generate a defined number of devices, their sensors and belonging data sets. For convenience reasons, the generated data is saved in *JSON* file

format and has a form that suits the algorithm input format. Via configurations, an occurrence percentage can be defined for each of the following error types:

- 1) **Missing data:** The absence of data points may have several reasons. Due to unstable network connections, there might for example occur package loss [31]. In addition, an empty device battery, for example, stops the recording and transmission of sensor data.
- 2) **Outliers:** The authors of [32] define outliers as “an observation or subset of observations that appear to be inconsistent with the rest of the data set”. In the context of the present work, the device microphone may produce outliers, for example, if the smartphone collides with a surface or object during measurement [8].
- 3) **Bias:** Described as a constant shift over time in comparison to the normal sensor readings. According to [33], this error might be caused by overheating of system components, which manifests itself, for example, in the microphone output although there is no perceivable sound source.
- 4) **Drift:** This error is similar to a bias, with the difference that a shift is increasing over time and does not depend on the sensor’s true value [33]. For example, microvibrations during smartphone usage throughout the day may cause gyroscope drifts [34].
- 5) **Noise:** As another additive fault, noise is described as “small variations in the data set” [10]. According to [35], a common model for this fault is white noise, a phenomenon that, e.g., microphones suffer from due to environmental influences.
- 6) **Gain:** Unlike previous types of sensor faults, gain is non-additive. The original sensor reading is multiplied by a constant value and therefore might deviate substantially from the original value [26]. In the context of the present work, gain might occur, for example, if the hardware sensor (i.e., the device’s microphone) is corrupted or if device software system components, such as the system APIs or the processing algorithms in the TYT app, are not operating correctly.
- 7) **Constant values:** As one of the most severe error types, sensors that only provide constant or *stuck-at-zero* values are unusable for almost all purposes [26]. For example, missing permissions to access the device’s microphone may lead to a constant value of zero being recorded by the TYT app.

The generator-script accepts the configuration described above as well as a random number generator seed. The latter parameter allows the simulation to work deterministic by making random choices using the seed. In order to generate error-free random, but correlated numbers at first, a random matrix $C = (c_{ij})$ is created for each device. If i is a property and j is a sensor measurement, c_{ij} is uniformly chosen from $[\frac{1}{2}, 1] \cup [-1, -\frac{1}{2}]$. Otherwise, it is uniformly chosen from $[-1, 1]$. Matrix C was designed to resemble a covariance matrix with dimensions equal to the sum of the number of sensor measurements and properties. Therefore, it is symmetric, but not necessarily positive semi-definite. For this reason, it is converted to the nearest covariance matrix, by converting it to a correlation

TABLE III
ERROR PROBABILITIES DEFINED FOR THE SIMULATION

Error Type	Probability
noise	0.15
bias	0.05
drift	0.005
outliers	0.03
gain	0.03
constant_value	0.005
stuck_at_zero	0.005

matrix, clipping the eigenvalues and then converting it back to the covariance matrix. *NUMPY*'s `multivariate_normal`-routine takes the resulting matrix to generate random but correlated numbers. As the outcome suffers from oscillations, a real periodic smooth random function [36] is applied, C giving function coefficients. Afterwards, the result is mapped to the interval of $[-1, 1]$.

The different types of errors (as described above) are then applied to the set of data points that were correct up to that point in time. Within the provided simulation configuration, probabilities of occurrence are specified for each type and generation is controlled accordingly. The probabilities used for the configuration in the scope of the present work are shown in Table III. Meta information about the injected error are stored in place with the data points. This meta information is not visible to the detection algorithm, but serves for analysis and validation purposes.

As an example, outliers are created according to their definition from [37]: Let p be a sensor value and let p_1, \dots, p_q, \dots be a sorted list of the remaining data points sorted ascending by their distance to p . The point p is called an $O(d, q)$ outlier if $|p_q - p| \geq d$ holds. Respectively, we try to mimic an $O(d, 3)$ outlier by calculating the mean μ of a sensor reading and its two direct siblings as well as using the specified standard deviation σ .

In addition to the actual simulation, a set of quality metrics was defined that allows a performance validation of the approach. These metrics are calculated by comparing the generated sensors with injected errors to the cleansed sensors after applying the proposed approach. The following metrics based on the *true positives* (TP), *false positives* (FP), *true negatives* (TN), and *false negatives* (FN) as defined by [38] are used:

- 1) Positive Predictive Value (also referred to as *precision* or *confidence*) [38]

$$PPV = \frac{TP}{TP + FP} \quad (4)$$

- 2) Fault Detection Rate (also referred to as *recall* or *true-positive rate*) [38]

$$FDR = \frac{TP}{TP + FN} \quad (5)$$

- 3) False-positive Rate [38]

$$FPR = \frac{FP}{FP + TN} \quad (6)$$

TABLE IV
CONCRETE VALUES THAT WERE CHOSEN FOR THE VARIABLES FROM TABLE I, FOR THE APPLICATION TO THE TYT DATA SET

Variable	Value
s	1
p	8
m	30
k	5
l	7
ϵ	0.001
δ	17

- 4) F-measure [39]

$$F\text{-measure} = \frac{2 * PPV * FDR}{PPV + FDR} \quad (7)$$

- 5) Average Error Reduction

$$\overline{ER} = 1 - \frac{\overline{Error}_{after}}{\overline{Error}_{before}} \quad (8)$$

where $\overline{Error}_{before}$ and \overline{Error}_{after} is the average error in the data set before and after the application of the proposed approach. The *Error* for a single sensor is thereby defined as the Euclidean Distance of the initially correct sensor values to the faulty sensor values after the error-injection.

4) *Data Application*: A real-world scenario to validate the proposed approach was given by its application to the TYT data set. Due to a diverging data structure compared to the accepted algorithm input, it was first converted into a utilizable form. This includes removing unnecessary information and selecting user data with a number of $m \geq 30$ sensor samples. As described in Section III-B2, a lower number of measurements and properties was considered non-analyzable. Sound level measurements were scaled into the target domain $[0, 1]$ in order to adjust the weighting of the scalar questionnaire answer data. Missing answers were afterwards imputed by kNN-imputation.

The parameters chosen for the application of the approach on the TYT data set are shown in Table IV. Since the data did not seem to suffer from noise, which was supported by a manual review of the data set, the optional Kalman filter was skipped. In addition, $\epsilon = 0.001$ was considered sufficient for an adequate constant value detection, because constant value readings in the TYT data set are not much affected by noise (i.e., they are mainly stuck-at-zero values). Parameters k , l and δ were chosen based on the results of several iterations of the algorithm with different parameters obtained through a binary search process (similar to the parameter selection during the simulation process in Section III-B3).

IV. RESULTS

The results from the simulation as well as the application of the approach to the TYT data set are described in the following.

A. Simulation and Performance

The characteristics of the simulation are shown in Table V. 50 runs of the simulation were performed resulting in a total of

TABLE V
RESULTS OF THE SIMULATION AFTER 50 RUNS

Characteristic	Value
Total simulated sensors	25,000
Total faulty sensors	6,465
Total sensors eliminated	4,491
Faulty sensors eliminated (TP)	1,877
Correct sensors eliminated (FP)	2,614
Average positive predictive value (\overline{PPV})	41.79%
Average fault detection rate (\overline{FDR})	29.02%
Average false-positive rate (\overline{FPR})	14.11%
F-measure	34.26%
Average error reduction (\overline{ER})	22.74%

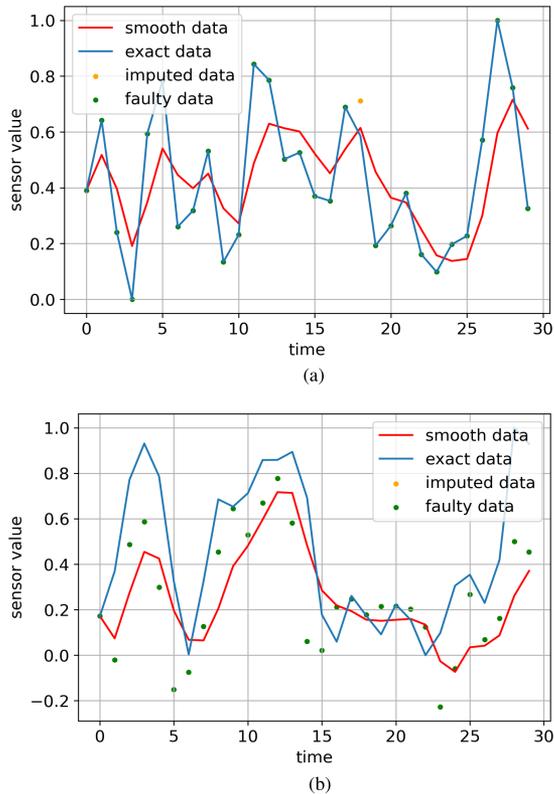


Fig. 2. Plots of the sensor values for simulated devices including different types of injected errors. (a) missing data. (b) missing data, noise, gain.

25,000 simulated sensors with different types of sensor faults according to the pre-defined error probabilities (see Table III). Examples for the values of these sensors with different injected error types are shown in Fig. 2.

In total, 6,465 (25.86%) of these sensors were considered faulty. After the application of the approach, a total of 4,491 (17.96%) sensors were eliminated, with 1,877 ($\overline{PPV} = 41.79\%$) of these sensors actually being faulty. Thus, the average fault detection rate (\overline{FDR}) over all runs was 29.02%. The fault detection rate of each individual run is shown in Fig. 3. In total, 2,614 correct sensors were eliminated by the approach, yielding to an average false-positive rate (\overline{FPR}) of 14.11%. Fig. 4 shows the false-positive rate for each individual run. Finally, the average error reduction (\overline{ER}) across all runs was

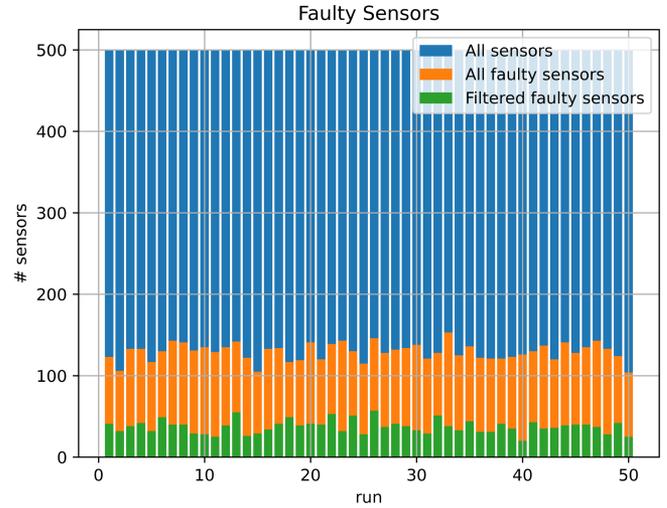


Fig. 3. Detection rate of faulty sensors for each run in the simulation.

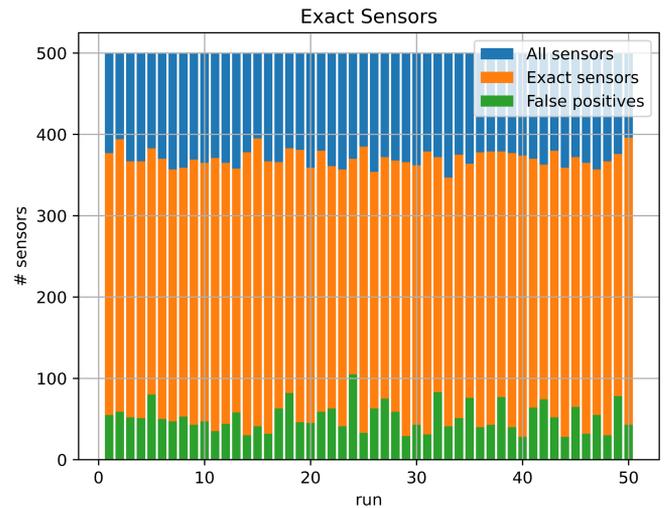


Fig. 4. False-positive rate for sensors for each run in the simulation.

22.74%. The error reduction for each individual run is shown in Fig. 5.

B. Application to the TYT Data Set

The results of the application of the elimination approach to the TYT data set are shown in Table VI. The 107,231 samples in the data set were contributed by 3,255 unique sensors (i.e., unique users with unique devices). 566 of these sensors were eligible for further processing because they held at least 30 samples. Furthermore, the KNN imputation excluded 24 sensors as they contained too many missing data points, resulting in 542 sensors that were considered in the elimination stage. The distribution of manufacturers of these sensors (i.e., devices) is shown in Fig. 6(a). During the elimination stage, the constant check eliminated 85 sensors. In the next step, another 12 sensors were eliminated by the PCA, resulting in a total of 97 (17.89%) sensors eliminated. The eliminated sensors and their contributed

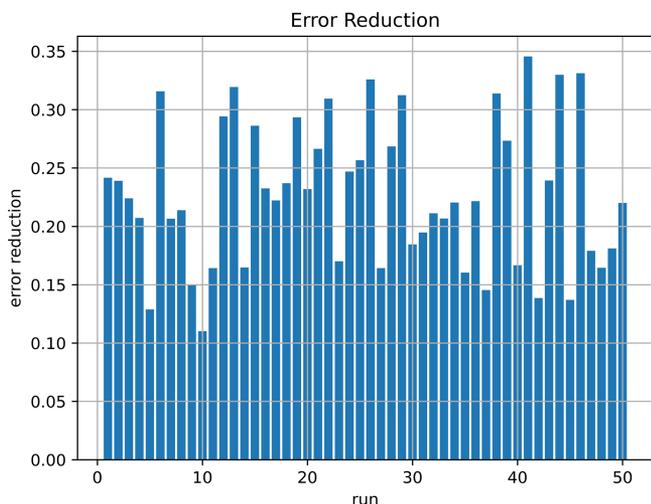


Fig. 5. Error reduction for sensors for each run in the simulation.

TABLE VI

RESULTS OF THE APPLICATION OF THE ALGORITHM TO THE TYT DATA SET

	Characteristic	Value
Baseline	Total sensors (i.e., users/devices)	3,255
	Total samples	107,231
Pre-processing	Total sensors considered	566
	Total samples considered	91,469
	Sensors after KNN imputation	542
	Samples after KNN imputation	89,111
Elimination	Sensors eliminated by constant check	85
	Sensors eliminated by PCA	12
	Total sensors eliminated	97 (17.89%)
Results	Sensors eliminated	97 (3.0%)
	Samples eliminated	14,819 (13.82%)
	Sensors remaining	3,158 (97.02%)
	Samples remaining	92,412 (86.18%)

samples were then removed from the original TYT data set, resulting in 14,819 (13.82%) samples removed. The distribution of sensor manufacturers after the elimination is shown in Fig. 6(b), while the distribution for the eliminated sensors is shown in Fig. 6(c).

V. DISCUSSION

In the following, the results of the simulation and the application to the TYT data set are discussed. Furthermore, limitations of the proposed approach are considered.

A. Interpretation of the Results

First, the results of the simulation are discussed in terms of the performance metrics. As shown in Section IV-A, the average positive predictive value (\overline{PPV}) of the proposed approach is 41.79% while the average fault detection rate (\overline{FDR}) is 29.02%. At the same time, the average false-positive rate (\overline{FPR}) is 14.11%. In other words, about every third faulty sensor is detected and eliminated by the proposed approach, while about every seventh eliminated sensor is in fact not affected by any faults other than missing data and should therefore be considered correct. The quality of the proposed approach can now be assessed by comparing these metrics. The first quality metric we can examine is the tradeoff between \overline{PPV} and \overline{FDR} ,

which is also captured by the F-measure as the harmonic mean of both values [38], [39]. Reducing the number of false-positive increases the \overline{PPV} . In reality, this increase often leads to a decrease of the \overline{FDR} , i.e., an increase in the number of false-negatives, at the same time [38]. The goal would be to maximize both values simultaneously, i.e., to maximize the F-measure. For the results of the simulation, both the \overline{PPV} and the \overline{FDR} are reasonably high, leading to an F-measure of 34.26%. In addition, we can assess the ability of the approach to discriminate between faults and non-faults by comparing \overline{FDR} and \overline{FPR} [38]. For an approach that just classifies sensors as faulty on a random basis, \overline{FDR} would be equal to \overline{FPR} , i.e., the chance of a true- or false-positive prediction would be equal [38]. However, for the proposed approach, \overline{FDR} is higher (even about twice as high) as \overline{FPR} , indicating a fair ability to discriminate between faults and non-faults. Moreover, the approach reduces the error by 22.74% on average (\overline{ER}). Overall, one can conclude a good performance of the proposed approach for the data sets generated in the simulation.

Second, the results of the application of the proposed approach to the TYT data set are considered. As shown in Section IV-B, only 542 of the 3,255 devices (16.66%) are eligible for the elimination process of the approach. This number seems low at a first glance, but it can be seen in Table VI that these devices account for 89,111 (83.10%) of the total samples in the data set. In the elimination stage, 85 devices are removed by the constant check, suggesting many constant value readings within the data set (as also found in [8]). A manual inspection of these values and a review of the code of the TYT mobile apps indicates that most of these constant values are zero values that occur because the Android app has not been granted permission to access the device's microphone. Nevertheless, this high number shows that the constant check is a valuable step in detecting sensor faults. Together with the second step of the elimination stage, the PCA, a total of 97 devices can be eliminated from the data set. As shown in Fig. 6, the manufacturer distribution before [see Fig. 6(a)] and after [see Fig. 6(b)] the elimination is similar. This is due to the fact that the manufacturer distribution of the eliminated devices [see Fig. 6(c)] roughly corresponds to that of the original data set. In other words, the devices classified as faulty by the proposed approach are distributed about evenly across device manufacturers. However, it can be observed that comparatively less miscellaneous devices are detected by the approach. This could be due to the heterogeneity of device manufacturers (e.g., LG, Motorola, HTC, OnePlus, Huawei, Xiaomi and others) in this group. The eliminated devices account for 3.0% of the devices and 13.82% of the samples in the total data set, as extracted from the TYT database. The remaining and cleansed data set could then be used as a baseline for further analyses of the environmental sound level measurements, for example, to investigate the interrelationships with tinnitus symptoms.

B. Limitations

The approach described in the scope of this work is subject to several limitations. These limitations will be discussed in the following paragraphs.

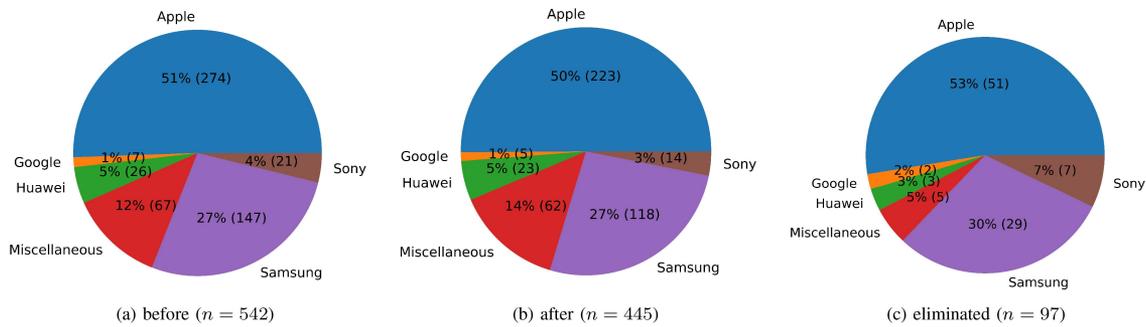


Fig. 6. Device manufacturer distribution (a) before and (b) after elimination of sensors that have been marked faulty as well as for (c) the eliminated sensors (i.e., devices).

Issues in the ecological and mHealth context: Some of the limitations of the proposed approach are given by several error factors, which occur in particular in the context of mHealth. Cho et al. [40] group these error factors into technical-related, user-associated and data-governance factors. As the TYT study is operated on data that is gathered during everyday life, there is no specified environmental setting, no delimited number of utilized devices. Hard- and software-related issues are accompanied by the “black box” behavior of smartphones concerning to their processing of collected data [41]. This lack of explainability, together with missing comparability between different operating systems and device manufacturers, states a huge quality issue. Standardization and calibration protocols [8] could provide a solution at least for the latter problem.

Assumption of over-determination: A main assumption of the algorithm in the proposed approach is an over-determination of the system. In this context, we included EMA questionnaire answers, as the devices and their sensors alone do not meet this requirement. It needs to be further investigated under which circumstance, for example, constant questionnaire answers might influence the algorithm outcome. In addition, our approach is a priori not able to handle data sets with missing values or too few samples ($m < 30$, see Section III-B2). [20] emphasize the ability of generalizing even from noisy or incomplete data that artificial intelligence-based approaches bring. It needs to be evaluated, how, for example, the generated simulation data set would be sufficient for accurately trained models and how the detection mechanism could be improved.

Lack of data distribution knowledge: Ayadi et al. [42] name a lack of data distribution knowledge prior to the analysis to be a huge challenge for parameter-based statistical outlier detection approaches. Without prior knowledge of the data form and according manual selection, the developed approach creates challenges in selecting fitting parameters and thresholds. Further research must evaluate if approaches using dynamic thresholds, such as [20], might be applied to our context.

PCA approach: The core detection mechanism of our algorithm, the PCA stage, mainly relies on an initial model training. Since this model should be trained using data mainly unaffected from errors, it is essential for the later detection mechanism. As it is still unclear how large this training set has to be, further improvements for the PCA algorithm are difficult to apply. These improvements could include, for example, the implementation

of a robust PCA, as it is less sensitive to anomalies than classical PCA [18], or of an online PCA algorithm [18], [19], which incrementally adapts an initially trained PCA model and therefore would increase performance.

False-positive rate: The proposed approach still shows a quite high false-positive rate. The application of the Boruta algorithm [43] might overcome this problem. To this end, this machine learning algorithm tries to select features that are relevant for the results. Properties are seen as features, sensors are equal to results. However, the inclusion of *BORUTA_PY*⁶ as implementation of the previously named algorithm, strongly increased computational complexity, but could not bring the desired results in a first attempt.

VI. SUMMARY AND OUTLOOK

In this work, we proposed an approach to cleanse Mobile Crowdsensing (MCS) mHealth data sets from erroneous sensor data. Specifically, the overall goal was to harness environmental sound level data provided via the TrackYourTinnitus (TYT) database to enable meaningful interpretation regarding the correlation of these objective sensor readings with subjective tinnitus symptoms. To this end, the existing data were analyzed and the problem formalized. The proposed approach includes several pre-processing steps and uses Principal Component Analysis (PCA), which considers the responses to the Ecological Momentary Assessment (EMA) questionnaire of the TYT app as additional information, and a preceding constant value check (*constant check*) to detect sensor errors. In addition, a simulation was designed and run using generated data with similar characteristics to the TYT data set to assess the performance of the approach with different parameters. The simulation was then used to evaluate the performance of the proposed approach with different quality metrics and to determine appropriate parameters for application to the TYT data. Finally, the approach was applied to the TYT data set, eliminating all devices that were found to be erroneous. The results show that the proposed approach performs well for data sets similar to the simulated data and therefore could be used for real-world scenarios.

⁶[Online]. Available: https://github.com/scikit-learn-contrib/boruta_py (Accessed: 2022-03-25).

Future work should further explore the aspects discussed in Section V. The proposed approach is subject to several limitations and uncertainties that need to be addressed. In addition, there are other approaches in the literature that could lead to even better results. For mHealth data sets such as TYT, additional information about the devices and the context of the sensor measurements would enable more accurate analyses. The TYT platform could therefore be extended to provide such information.

In conclusion, it has been shown that sensor errors in MCS mHealth data sets can be detected using a data-centric approach. However, there are many challenges and uncertainties when the distribution of sensor errors is not known a priori and when little to no additional information is available about the data quality of sensor measurements. Therefore, further work should and must be done in the future to interpret MCS mHealth data.

REFERENCES

- [1] R. Kraft et al., "Combining mobile crowdsensing and ecological momentary assessments in the healthcare domain," *Front. Neurosci.*, vol. 14, pp. 1–14, 2020, Art. no. 164.
- [2] R. Kraft et al., "Efficient processing of geospatial mhealth data using a scalable crowdsensing platform," *Sensors*, vol. 20, no. 12, 2020, Art. no. 3456.
- [3] R. Kraft et al., "Design and implementation of a scalable crowdsensing platform for geospatial data of tinnitus patients," in *Proc. IEEE 32nd Int. Symp. Comput.-Based Med. Syst.*, 2019, pp. 294–299.
- [4] R. Y. Wang and D. M. Strong, "Beyond accuracy: What data quality means to data consumers," *J. Manage. Inf. Syst.*, vol. 12, no. 4, pp. 5–33, 1996.
- [5] W. Schlee et al., "Measuring the moment-to-moment variability of tinnitus: The trackyourtinnitus smart phone app," *Front. Aging Neurosci.*, vol. 8, 2016, Art. no. 294.
- [6] R. Pryss, M. Reichert, J. Herrmann, B. Langguth, and W. Schlee, "Mobile crowd sensing in clinical and psychological trials—a case study," in *Proc. IEEE 28th Int. Symp. Comput.-Based Med. Syst.*, 2015, pp. 23–24.
- [7] R. Kraft et al., "Comprehensive insights into the trackyourtinnitus database," *Procedia Comput. Sci.*, vol. 175, pp. 28–35, 2020.
- [8] R. Kraft, M. Reichert, and R. Pryss, "Towards the interpretation of sound measurements from smartphones collected with mobile crowdsensing in the healthcare domain: An experiment with android devices," *Sensors*, vol. 22, no. 1, 2022, Art. no. 170.
- [9] A. Sharma, L. Golubchik, and R. Govindan, "On the prevalence of sensor faults in real-world deployments," in *Proc. 4th Annu. IEEE Commun. Soc. Conf. Sensor*, 2007, pp. 213–222.
- [10] H. Y. Teh, A. W. Kempa-Liehr, and K. I. K. Wang, "Sensor data quality: A systematic review," *J. Big Data*, vol. 7, no. 1, pp. 1–49, 2020.
- [11] R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy, "Sensor fault identification and reconstruction using principal component analysis," *Int. Federation Accountants Proc. Volumes*, vol. 29, no. 1, pp. 6578–6583, 1996.
- [12] R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy, "Use of principal component analysis for sensor fault identification," *Comput. Chem. Eng.*, vol. 20, pp. S713–S718, 1996.
- [13] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.
- [14] P. Miller, R. E. Swanson, and C. E. Heckler, "Contribution plots: A missing link in multivariate quality control," *Int. J. Appl. Math. Comput. Sci.*, vol. 8, no. 4, pp. 775–792, 1998.
- [15] M. Safaei et al., "A systematic literature review on outlier detection in wireless sensor networks," *Symmetry*, vol. 12, no. 3, 2020, Art. no. 328.
- [16] G. Jäger et al., "Assessing neural networks for sensor fault detection," in *Proc. IEEE Int. Conf. Comput. Intell. Virtual Environments Meas. Syst. Appl.*, 2014, pp. 70–75.
- [17] T. Brade, M. Schulze, S. Zug, and J. Kaiser, "Model-driven development of embedded systems," in *Proc. 12th Braz. Workshop Real-Time Embedded Syst.*, 2010, vol. 5, pp. 3–14.
- [18] L. B. Amor, I. Lahyani, and M. Jmaiel, "PCA-based multivariate anomaly detection in mobile healthcare applications," in *Proc. IEEE/ACM 21st Int. Symp. Distrib. Simul. Real Time Appl.*, 2017, pp. 1–8.
- [19] L. B. Amor, I. Lahyani, M. Jmaiel, and K. Drira, "Anomaly detection and diagnosis scheme for mobile health applications," in *Proc. IEEE 32nd Int. Conf. Adv. Inf. Netw. Appl.*, 2018, pp. 777–784.
- [20] B. Saneja and R. Rani, "An efficient approach for outlier detection in big sensor data of health care," *Int. J. Commun. Syst.*, vol. 30, no. 17, 2017, Art. no. e3352.
- [21] T. Aderibigbe and H. Chi, "A quick outlier detection in wireless body area networks," in *Proc. Pract. Experience Adv. Res. Comput. Sustainability*, 2017, pp. 1–3.
- [22] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, and P. Bhattacharya, "Wireless sensor network simulators a survey and comparisons," *Int. J. Comput. Netw.*, vol. 2, no. 5, pp. 249–265, 2011.
- [23] H. Zhang, J. Liu, and R. Li, "Fault detection for medical body sensor networks under bayesian network model," in *Proc. 11th Int. Conf. Mobile Ad-hoc Sensor Netw.*, 2015, pp. 37–42.
- [24] A. Koren, M. Jurčević, and R. Prasad, "Comparison of data-driven models for cleaning ehealth sensor data: Use case on ECG signal," *Wireless Pers. Commun.*, vol. 114, no. 2, pp. 1501–1517, 2020.
- [25] R. Pryss, W. Schlee, B. Langguth, and M. Reichert, "Mobile crowdsensing services for tinnitus assessment and patient feedback," in *Proc. IEEE Int. Conf. AI Mobile Serv.*, 2017, pp. 22–29.
- [26] J. Kullaa, "Detection, identification, and quantification of sensor fault in a sensor network," *Mech. Syst. Signal Process.*, vol. 40, no. 1, pp. 208–221, 2013.
- [27] E. Fix and J. L. Hodges, "Discriminatory analysis nonparametric discrimination: Consistency properties," *Int. Stat. Review/Revue Int. de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.
- [28] J. K. Dixon, "Pattern recognition with partly missing data," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 10, pp. 617–621, Oct. 1979.
- [29] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
- [30] L. Mujica, J. Rodellar, A. Fernandez, and A. Güemes, "Q-statistic and t2-statistic pca-based measures for damage assessment in structures," *Struct. Health Monit.*, vol. 10, no. 5, pp. 539–553, 2011.
- [31] Y. Li and L. E. Parker, "Nearest neighbor imputation using spatial-temporal correlations in wireless sensor networks," *Inf. Fusion*, vol. 15, pp. 64–79, 2014.
- [32] V. Barnett and T. Lewis, "Outliers in statistical data," *Wiley Ser. Probability Math. Statist. Appl. Probability Statist.*, 2nd ed. Chichester, U.K.: Wiley, 1984.
- [33] J. Rabatel, S. Bringay, and P. Poncelet, "Anomaly detection in monitoring sensor data for preventive maintenance," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7003–7015, 2011.
- [34] A. Lawrence and F. F. Ling, "Gyro and accelerometer errors and their consequences," in *Modern Inertial Technology*. New York, NY, USA: Springer, 1998, pp. 25–42.
- [35] A. Klöckner, F. L. van der Linden, and D. Zimmer, "Noise generation for continuous system simulation," in *Proc. 10th Int. Modelica Conf.-Lund*, 2014, pp. 837–846.
- [36] S. Filip, A. Javeed, and L. N. Trefethen, "Smooth random functions, random odes, and gaussian processes," *Soc. Ind. Appl. Math. Rev.*, vol. 61, no. 1, pp. 185–205, 2019.
- [37] B. Sheng, Q. Li, W. Mao, and W. Jin, "Outlier detection in sensor networks," in *Proc. 8th ACM Int. Symp. Mobile ad hoc Netw. Comput.*, 2007, pp. 219–228.
- [38] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Comput. Surv.*, vol. 42, no. 3, pp. 1–42, 2010.
- [39] C. J. Van Rijsbergen, "Foundation of evaluation," *J. Documentation*, vol. 30, no. 4, pp. 365–373, 1974.
- [40] S. Cho, I. Ensari, C. Weng, M. G. Kahn, and K. Natarajan, "Factors affecting the quality of person-generated wearable device data and associated challenges: Rapid systematic review," *J. Med. Internet Res. Mhealth Uhealth*, vol. 9, no. 3, Mar. 2021, Art. no. e20738.
- [41] P. Düking, F. K. Fuss, H.-C. Holmberg, and B. Sperlich, "Recommendations for assessment of the reliability, sensitivity, and validity of data provided by wearable sensors designed for monitoring physical activity," *J. Med. Internet Res. Mhealth Uhealth*, vol. 6, no. 4, Apr. 2018, Art. no. e102.
- [42] A. Ayadi, O. Ghorbel, A. M. Obeid, and M. Abid, "Outlier detection approaches for wireless sensor networks: A survey," *Comput. Netw.*, vol. 129, pp. 319–333, 2017.
- [43] M. B. Kursa, A. Jankowski, and W. R. Rudnicki, "Boruta—A system for feature selection," *Fundam. Informaticae*, vol. 101, no. 4, pp. 271–285, 2010.