Computers and Electrical Engineering 93 (2021) 107181

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Sleep pattern inference using IoT sonar monitoring and machine learning with Kennard-stone balance algorithm

Tengyue Li [a,*], Yaoyang Wu [a], Feng Wu [b], Sabah Mohammed [c], Raymond K. Wong [d], Kok-Leong Ong [e]

[a] Zhuhai Institute of Advanced Technology, Chinese Acadmy of Science, China
[b] Medical Device R&D Department, Zhuhai Institute of Advanced Technology China
[c] Department of Computer Science, Lakehead University, Thunder Bay, Canada
[d] School of Engineering, University of New South Wales, Australia
[e] School of Business, La Trobe University, Australia

ARTICLE INFO

Editor: Dr. M. Malek

Keyword:
Sleep pattern analysis
Data stream mining
Sonar sensors
IoT monitoring
Data sampling

ABSTRACT

A new paradigm of IoT monitoring using sonar sensors and microphones is studied as a contactless alternative to the traditional devices for sleep disorders in this paper. A pair of sonar sensors are used to measure the distance from the bottom and side of the body to the bedside respectively, and the basic posture of the human body during sleep can be inferred by using machine learning. When a person sleeps still, two streams of sonar signals from two adjacent sides remain unchanged. Any movement would disrupt the stationary sonar streams. Hence it could be detected as change of posture from lying still. Different sonar patterns could be recognized as specific sleeping postures by some non-linear machine learning model. This simple and novel solution could potentially be used as an alternative or supplementary to video analytics which can feedback to the user about their sleeping pattern. A new data transformation method namely Kennard-stone Balance (KSB) algorithm is also proposed for simplifying the data streams and enhancing the accuracy of the machine learning model. Simulation results show the feasibility of this sonar method, and KSB is able to improve the pattern recognition performance.

1. Introduction

Sleep disorder which is one of the prominent health problems nowadays comes in various forms. One of the most serious cases is sudden infant death syndrome (SIDS). SIDS is defined as [1]: "The sudden death of an infant under 1 year of age, which remains unexplained after a thorough case investigation, including performance of a complete autopsy, examination of the death scene, and review of the clinical history." This disease has following risk factors, which can increase the possibility of SIDS happening: infant prone sleep position, sleeping on a soft surface, maternal smoking during pregnancy, overheating, late or no prenatal care, young maternal age, prematurity and/or low birth weight [2]. Among these factors, relatively controllable ones are infant prone sleep position, sleeping on a soft surface and overheating to avoid SIDS. According to statistics, the incidence of SIDS has sharply declined in the

This paper is for special section VSI-aiia. Reviews processed and recommended for publication by Guest Editor Prof. Shiqian Wu.
* Corresponding author.
E-mail addresses: gloria.l@ziat.ac.cn (T. Li), wuyaoyang@ziat.ac.cn (Y. Wu), wufeng@ziat.ac.cn (F. Wu), mohammed@lakeheadu.ca (S. Mohammed), wong@unsw.edu.au (R.K. Wong), Kok-leong.Ong@latrobe.edu.au (K.-L. Ong).

https://doi.org/10.1016/j.compeleceng.2021.107181
Received 7 July 2020; Received in revised form 14 April 2021; Accepted 23 April 2021
Available online 5 July 2021

United States, since the US promote to reduce infant prone sleep [3]. As long as adult could keep eyes on infant to avoid prone sleep, it would help to prevent SIDS happening. However, during the night time or parent busy time, it is difficult to detect anomalies. So that, a 24 h infant sleep position detection system is necessary for every family with new born baby.

Nowadays, there are many healthcare products to monitor sleep quality. Usually, the methods are divided into two groups, contact-based and noncontact. For the former group, there is a standard diagnostic tool in sleep medicine, called Polysomnography (PSG). It measures a wide range of bio-signals, including blood oxygen, airflow, electroencephalography (EEG), electrocardiography (ECG), electromyography (EMG) and electrooculography during sleep time [4]. It will unavoidably use many sensor contact human body, not only disturbing human natural sleep, but also costly. Another group is noncontact methods, using image sensor to capture sleep posture, such as microwave, thermal imaging and near-infrared camera.

However, the existing methods are not suitable for monitoring the sleeping position of infants. Firstly, contact-based sensor will cause strong curiosity or fear for infants, which will directly infect infants sleep. Secondly, time cost is precious. Since, asphyxia caused by infant prone sleep can be rescued immediately when found in time. In this scenario, image-based detection methods may lose its advantages.

In this paper, we proposed a simple, quickly responsive and cost-sensitive method based on IoT sonar sensor. Just by determining two points of proximity data streams, and learning the current changing distance of each point, a machine learning model can determine whether the baby is lying flat, lying on its side or lying prone. This in-home monitoring device can extend its application to other sleep disorder cases as well. In general, proximity sensors inform users as a post-sleep feedback about the quality of sleep over the night. However, it can be programmed to serve as a watch-dog – if any abnormal pattern about the sleep is detected, an alerting alarm goes off. It will be useful detection hence prevention in critical situations, such as a baby is about to fall off from the edge of the bed, being in some near-suffocation position, or lying in the same position all night which probably will lead to stiff joints.

There are a large variety of health monitoring devices on the market. Traditionally, these devices read the vital signs such as cardiology signals, SpO2 levels, blood pressures, and sometimes brain waves directly from the skin contacts, by wearing a wrist band and/or sleeping over an electronic pad. A lot of research works have advanced in the literature in this aspect. Contactless technologies, besides video or depth camera analytics, are relatively less explored.

Therefore, in this paper, we explore the potential effectiveness of using IoT sonar sensor (IoTSS) as a means to measure proximity information from sleep behaviour patterns. Such move could potentially be used as feedback or detection mechanism towards assessing sleep disorder, in the hope of further study about SIDS or similar health concern related to sleep.

The reminder of this paper is structured as follow. Related work is in Section II. The solution for monitoring sleep patterns using sonar sensors and KSB algorithm is proposed in Section III. That includes the hardware and software designs, pre-processing method and the program design. Experiment is presented in Section IV. The experimental results are discussed in Section V. Section VI concludes the paper with future work.

2. Related work

2.1. The relation between sleep posture and sleep disorder

Many studies show sleep posture has strong relation with sleep disorder [4–6]. Normally, human would choose supine, lateral and prone positions during sleep time. Prone position is rarely to find among adults but commonly in teenager and infant. Since many countries like the US and European countries proposed to avoid prone position and launched a "suppressive sleep" education campaign, most of the adults have awareness of avoiding prone sleep. However, teenager and infant are the big group of prone position members and inability of self-correct which cause a series of sleep problems. One of the key reasons to cause SIDS is prone position sleep. In 1992, the American Academy of Pediatrics (AAP) has warned parents and babysitters to let their baby sleep on their backs to reduce SIDS risk [7]. The incidence of SIDS in the United States has fallen by more than half. Experts attribute this outstanding result to recommending supine sleep. However, keeping a watchful eye on a baby over 24 h and 7days a week is not possible by any human.

Not only the SIDS is related with sleep behavior, there are also Obstructive Sleep Apnea Syndrome (OSAS), Autism Spectrum Disorder (ASD) and Attention Deficit and Hyperactivity Disorder (ADHD). Some studies [8,9] found that OSAS affecting 500,000 children per year in the United States, long time supine sleep will worsen OSAS, compared with supine and prone position, lateral position is the most suitable posture for children with OSAS. For children with ASD, they usually company with sleep problems. Although sleep problems are not part of the diagnostic criteria for autism, they are considered to be a characteristic of the ASD phenotype. Early morning waking, unusual bedtime routines, and night walking are common among children with ASD. Home sleep monitoring is important to each family, especially for home with infant or children with sleep problem. Through scientific analysis [10], the result could be helpful to avoid negative situation and to find and cure sleep problem.

2.2. Monitoring sleep based on contact sensor

There is a standard diagnostic tool [11] in sleep medicine, called Polysomnography (PSG). It measures a wide range of bio-signals, including blood oxygen, airflow, electroencephalography (EEG), electrocardiography (ECG), electromyography (EMG) and electrooculography during sleep time [12]. Although contact sensor could get bio-signals which cannot be collected from contactless sensor, these methods unavoidably affect user natural sleep routine. Moreover the younger user, like infant and teenager, they are more sensitive to anything new than adults.

Pressure mattress is a better way to collect sleep data with tiny and sensitive pressure sensors which cannot be felt. Through analysis movement frequency and body pressure point, it could show the sleep state and the posture of sleep. However, this method requires matrix-embedding a lot of pressure sensor to form a network structure, which makes the product manufacturing costly, not suitable for normal family use. In addition, from the perspective of accuracy, the shallower the sensor, the more accurate the pressure data. However, this will cause another problem. Sensors close to the body can make people feel uncomfortable. Monitoring sleep based on contactless sensor

Monitoring through contactless approach seems the most ideal solution. Image sensors are commonly applied in contactless solution, including microwave, thermal imaging and near infrared camera. These advanced sensors could collect data no matter user with cover or not and could record by video during sleep. To enhance the accuracy, some studies proposed to design a pajama with special marks on the human joints to facilitate the tracking and identification of image sensors. However, each above mentioned sensor is not only highly cost, but also may violate user's privacy once the data has security risks. So that, using image sensor is not the perfect solution for smart home. Except image sensor for sleep detection contactless solution, it is hard to find a way to monitor sleep using other sensors. In this situation, we innovatively proposed this method based on sonar sensor.

2.3. IoT sleep pattern sensors in industrial applications

Internet of Things (IoT) technology plays an important role in the modern industrial applications for solving sleep disorder problems [13]. From the economical perspective of industrial application, IoT offers the preambles and possibilities to build cost-effective sleep monitoring systems [14]. Off-the-shelf commercial sensors upgraded with Internet communication abilities to be IoT sensors are used as relatively inexpensive components to measure the sleep quality in general use. The sensors in those low-cost systems monitor bio signals such as bodily movement, heartrate, blood oxygen saturation level, audio patterns of snoring and moving in bed. With the Internet transmission capabilities, measured data are transmitted from sensors in real-time to a central system, for analytics over the patterns of data. Often AI is used for inferring the sleep behaviours and for evaluating the quality of sleep.

3. Proposed solution

In the industrial applications of sleep quality evaluation, more sophisticated sensing equipment than those off-the-shelf sensors are manufactured, often for specialized uses. Just to name few: acoustical respiratory monitoring devices, that consists of electronic stethoscopes and pulse oximetry, is used for precise acoustic respiratory rate monitoring for clinician use. The sensors are based on piezoelectric film technology which are generally delicate and expensive [15]. Another example of how specialized device is used to monitor electrocardiographically derived respiration, or electrocardiogram based EDR in short, is using conductive textiles in bed [16]. The patient is lying on bed in supine position. The textiles are placed over both sides of his body. Analogous electrical signals are produced, from the textiles over his chest as bed sheets and his head as a pillow. In this case, special Ag coated textile which is sensitively conductive to the moisture of the skin is used. While the hardware is tuned to its optimum, AI is applied in better analyzing the signals, and also in suppressing the noise level between the skin and electrode in signal processing [17]. There are other sophisticated systems in the industrial applications where IoT is used mainly to rely the measured data remotely to a central system. Another extreme example is sleep apnea machine which has high connectivity over multiple sensors including low-pressure airflow sensors, board mount pressure sensors, humidity sensors, atmospheric sensors and magnetic sensors working as a single system. AI again is applied for modelling the sleep patterns from the collected data, via heuristic approaches, machine learning and statistical approaches and even deep learning and transfer learning for enabling the predictive ability. Overall, these technologies are mainly categorized into medical and consumer groups where sophistical and expensive versus economical but low precision production costs are associated with respectively [18]. The medical group has the main objective of serving as clinical support systems, and for intervention and research purposes. The most common sleep related problems are obstructive sleep apnea and sleep disorder where these specialized monitoring systems are able to help [19]. The consumer group offers general healthcare services as recommendation system, and consumer-graded gadgets as lifestyle products. In both cases, IoT technology and AI play their roles in connectivity and communication, and analysis and recommendation making respectively.

The monitoring strategy that we proposed here consists of two parts: the data collection framework using sonar sensors for capturing sleeping patterns; and the machine learning model which interprets from the collected sensing data about what the sleeping behaviors are.

3.1. The sonar sensing strategy

IoT sonar sensors are defined as distance-sonar sensors with an IoT enabled gateway which can relay continuous streams of measured data to a remote analysis center, while knowing the identifier of each individual sensor by a unique IP address. In our proposed solution, sonar sensors are installed in L-shaped at the rim of a bed or mattress which scan from two sides of the bed to the subject which is lying at the center of the bed. Assuming that the bed is of rectangular shape, the scanning directions of the sensors cover the side view (body) and bottom view (feet) respectively. Therefore, the intensity and frequency level of bodily movements and feet movements can be captured and measured in terms of displacement from the sensor. It could be imagined that the two rows of sensors forming a L shape, produce a pair of constant data flows which are perpendicular to each other. If the subject that lies at the intersection which is normally the center of the bed remains perfectly still during sleep. The displacement values from the two rows of sensor remain unchanged. As such, any slight movement of the subject would disrupt the sensor data streams, causing fluctuations at the data stream values. The changes in values in the data streams which are constantly measured and analyzed in real-time, serves as

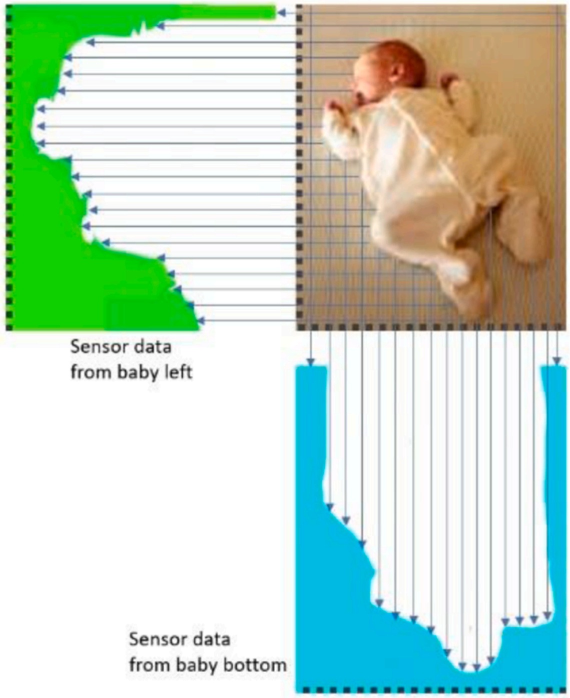


Fig. 1. Illustration of sonar sensors are being deployed to monitor the posture of a sleeping baby.

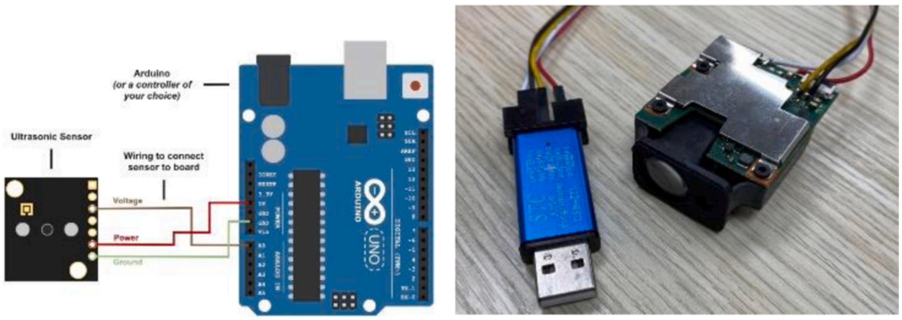


Fig. 2. The custom-made sonar sensor used in our experiments; left: sensor and the controller, right: the actual sensor.

indication about the movement of the subject if any. The resolution of detection of the subject, about how subtle the changes of the body, would be proportional to the density of the available sensors installed along the two sides of the bed. As an illustration, Fig 1 shows a crib in which a baby is sleeping and its left and bottom edges each has a row of sensors installed. The two arrays of sonar sensors would be sensing the distance to the target, giving an outline of the target on the left part of the baby, and an outline at the bottom side of the baby. This way, any slight movement by the baby, the sensed distances would change in values. It is known that humans do exhibit certain postures during sleep as a pattern signature; by using machine learning which maps the information about the two arrays of sensed data to specific sleep postures, a computer or software program could learn about these sleep postures in supervised learning. Once they have learnt sufficiently well, the sonar sensors with the ML model could be capable of recognizing the postures in a field application.

Intuitively and ideally, the more sensors the high resolution could be achieved, but the cost increases. It is therefore sensible to try find the minimum number of sensors that could offer just sufficient information for the application, where medical diagnosis requires higher resolution, general health monitoring could take moderate resolution. For example, a 4×1 configuration of sensors, having four sensors at the longer side of a bed and one sensor at the foot position, could give movement data on five positions: the head-and-shoulder, arms-and-chest, waist-and-hips, and shin positions on the side, as well as feet positions at the bottom. On one extreme, if only two sensors to be deployed, probably for cost-saving purposes, the movements then on the body side and feet positions would be generalized to be the overall displacement from the sensors which are stationary at the boundary of the bed.

There is a spectrum of sleep related information that can potentially be measured by some kind of smart bed technology, that is

Table 1
Table of sleep-related symptoms that could be measured by different sensing technologies.

Level of complexity / precision	Information	Example	Devices
5	Body vital signals	Emotions, EEG, ECG, health stats	Medical wearable devices
4	Sleep dynamic behaviors	Movements, activities, actions	Hi-resolution video cameras, wearable sensors
3	Sleep static postures	How the body curl up, limbs positions, sleeping positions, joints locations	Video cameras, IR imaging, wearable sensors
2	Body locations	whereabouts in bed (near center, near edge), overall body postures	Video cameras, proximity sensors
1	Sleep status	Idle state, active state, intensity and frequency of movements, calmness vs chaos	Pressure sensors, sound analytics, video analytics and proximity analytics

depicted in Table 1. Among this spectrum of measurements, IoT sonar sensors are capable of covering the detection of body movement and postures, which could be used in lieu of video analytics. The main advantage is privacy protection because video captures usually the full features of a person including the face; sonar sensors takes only the relative displacement between the sleeper and the perimeter of the bed. Different postures and movement could be recognized by streams of non-interrupted reflected signals, provided that appropriate machine learning models are available.

3.2. The hardware component

Sonar sensors which are also known as proximity sensors, ultrasonic sensors etc. are a safe, reliable, cost-effective solution for sensing distance sensing, proximity, and obstacle detection. The principle of sonar sensor is based on an ultrasound wave at an ultra-high frequency beyond human audibility. The sensor has an electronic transducer like a microphone that emits out and receives back the ultrasonic pulses that bounced back from an object for calculating the object's proximity.

The distance between the target and the sensor is measured from the time lapses between the sending the ultrasonic pulse typically at 40 kHz and relaying of its echo. The formula for calculating the proximity distance is simply a division of the round-trip travel time which is known as time of flight, by the speed of sound in air at 58 uS/cm. For example, if the target object is placed 45 cm from the sensor, the measured time of flight is 2619 uS by the transducer, the range (in cm) = time of flight ÷ speed of sound; range = 2610 uS ÷ 58 uS/cm = 45 cm.

Pegging the sensors stationary at a fixed position along the edge of a bed, it is therefore possible to measure the time of flight as displacement from the target sleeper. In general, sonar sensors are cost-effectively used in contactless detection of: presence, position, distance and intensity of movements of the target. The operations of the sensors using time of flight are unaffected by the target's materials and color, light, dust, and smoke. Sonar sensors are more reliable than infrared sensors in-home monitoring context because infrared imaging is sensitive to thermal blanket, netting materials, insulated jackets, cold air flow from air-conditioner etc. which disrupt heat signatures.

The configuration of the sonar sensors used in this experiment is rather simple. An ultrasonic transducer built onto an Arduino controller which sends the proximity data via an USB cable and into a computer. After the controller has the Arduino uploaded, a coding software called Arduino Sketch can be installed. The following code can be compiled and activated for starting to feed the proximity data to the computer from the sensor. The code instructs the sensor using assembly directives from Arduino board to acquire real-time distance measurements in cm, that is detected from whatever closest object in front of the sonar sensor. More details about the hardware setup can be found at: https://gist.github.com/Maxbotix/b417753919ce736d4ef31f46aa73f313#file-read-ultrasonic-an-distance-in-cm

3.3. The software component

There are mainly three modules in the software architecture: the data acquirer, the data preprocessor and the pattern recognizer. When the sensors operate, the data acquirer is acting as a middleware interface with the controller that controls the data flow. Data sampling rate is set here. It maintains the constant speed flow of the data stream with little latency and performing error checking at data level. The data is collected in real time which loads into the preprocessing module for "compression" by some specific algorithm called Kennard-stone which is further elaborated in the following section. The final part in the software system is the machine learning model - it is a dual step which uses supervised learning to learn and recognize the data stream patterns from different sensor positions. Once the model has enough representative sleeping postures and movements, it remembers so. Subsequently it is able to recognize unseen patterns in the future by using the induced model from the seen training data. The supervised learning works by mapping the time series values into the target labels which are defined as different types of sleeping behaviors as well as what deemed as healthy and unhealthy. The software architecture that we proposed here is generic, that should be able to work with different learning algorithm. Therefore, our following experiment is geared towards testing the efficacy of different machine learning algorithms coupling with different preprocessing methods, for the objective of achieving the highest possible recognition rate.

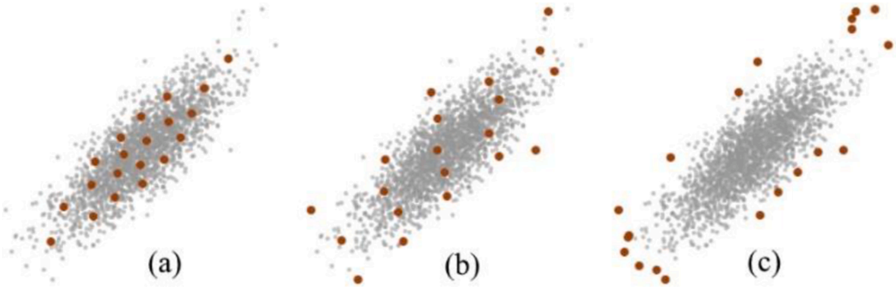


Fig. 3. Data points selected by (a) K-means, (b) Kennard-Stone, and (c) Farthest-first.

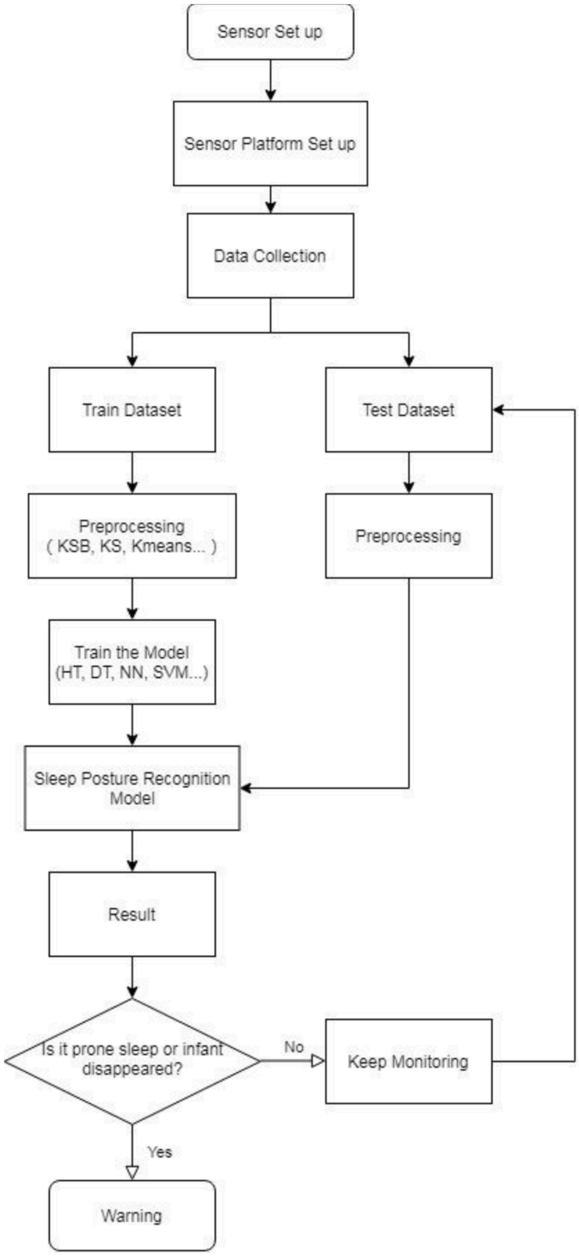


Fig. 4. Flow-chart of sleep monitoring using sonar sensors.

3.4. The preprocessing methods

In this monitoring strategy, a pre-processing method is proposed about applying a variation of Kennard-Stone (KS) algorithm for downsizing a data stream for improving the performance of the ML model. The KS algorithm [20] was originally designed for generating calibration and testing sets from an original dataset when standard experimental design was unavailable. It picks a subset of samples from the original full set which provide uniform coverage (by following a statistical uniform distribution) over the dataset plus samples at the boundary of the dataset. One can see that the result of the data subset marks the outline or shape of the whole dataset very well because it picks samples starting from the edge, then subsequently from the centre region.

An example of how a data sampling algorithm can be used to optimize the power of a predictive model is given here. Assume a number of points are artificially generated and distributed over x- and y-axis in the search space. One thousand simulated points are drawn from a 2D normal distribution at random. A data sampling algorithm is then applied to choose twenty points, in such a way that the twenty chosen points will be representing the original population of the points. In other words, the training dataset that would be used to induce a predictive model would come from a smaller subset of data in lieu of all the data points. This would boost the efficiency of the model induction, at least shortening the model training time in proportion to the ratio of the subset selected out of the full set. In our case, the ratio is 20/1000. This approach would work well for tackling big data problem with respect to model induction. However, the ultimate accuracy level of the induced predictive model would depend very much on how representative or generalized the twenty points that are chosen pertaining to the full set of points. The simplest approach is random which we would expect the accuracy of the model to be around the baseline performance at 50%. Three popular approaches are used here for illustrating the effects of data sampling: K-means, Kennard-Stone (KS) and Farthest-first (FF) [21].

As it could be seen in above illustration, the twenty selected points are clustered near the center in Fig. 3(a) when K-means was used. The distances between the selected points are more or less even spaced, though they tend to stay close to the center of the whole set. In Fig. 3(c), an opposite distribution of the selected data points can be seen; they are lying at the farthest edge of the data set, as a result of applying FD. In Fig. 3(b), KS selected points from a random mix of the positions that are at the most outer edge and that are evenly distributed over the whole dataset. By visual inspection, using the selected points in Fig. 3(b) by KS, a relatively more reliable predictive can be built, since KS gathers the points that embrace the outline of the data group and those that represent the majority of the points near the center. Furthermore, it is already proven by Rodionova and Pomerantsev [22] that (1) the predictive model constructed with the help of KS subset can predict all other samples with accuracy that is not worse than the error of selected samples evaluated on the whole data set. (2) KS subset is indeed significantly smaller than the whole training set.

Intuitively, KS should work well if and only if the points at the outer edge have more significance than those in the middle or other regions. K-means selects mainly the medoids within the whole set, which are the results of data points being averaged out, holding only the mediocre values. FD selects only the extreme points, more often they are outliers. Depending on the applications, they should have their own strengths and weaknesses. For an example of application, in earthquake prediction when only the rare-events of seismic occurrence with very high Richter magnitude scale are of concern, FD is useful in sampling out merely the rare events for building a strong predictive model. KS however would be suitable for everyday use prediction which concerns about predicting both the rare events and normal events, by building a general-purpose prediction model.

Supposing KS is a useful sampling method to be deployed for streamlining big data while keeping the distribution of the selected samples even, with a proper mix of outliers and normal values, the efficacy of the resultant predictive model depends on one more factor - the class distribution. In data mining, class balancing is an important topic and factor that determines the final performance of the trained predictive model. Assume there are two classes of data in the whole sample - a majority class and minority class. Minority class is often a small group of samples that come in small quantity (rare) but they have a very strong interestingness or significance in the prediction, e.g. forecasting a mega disaster, or discovering a rare and new disease. The majority class samples are the ordinary ones that have less or little meaning, but there are many of them in the population. It is a well-known problem called "class-imbalance" in data mining. It happens when the training samples for building prediction model are taken mostly from the majority class samples. During the training process, the predictive model was trained with very little or none at all of the minority samples. Consequently, the model that was trained with mostly majority samples, suffers from very poor accuracy rate upon testing with rare minority samples. Unfortunately, this class-imbalance problem usually may go undetected, because during testing there may be a high chance that the testing samples come also from the majority class. The model may appear as quite accurate where the success rate of prediction comes mainly from testing with majority samples. This sort of accuracy is pseudo and the model will have low generality. When a completely new dataset is used in testing which may contain certain amount of rare sample, the accuracy drops. This effect of the class-imbalance problem obviously depends on how the training samples were selected, by using which data sampling algorithm prior to model construction. Therefore, in this study, it is wary that using a naive version of KS is insufficient to cover the class-imbalance problem which is not uncommon in data mining. Moreover, KS is meant to be used in big data environment, given its capability in abstracting an appropriate subset of samples from a huge population. The current design of KS requires a full set of data to be available in order to run the algorithm. For big data, the data size may be unbounded so it would be natural and efficient if KS could be extended to suit data stream mining. Therefore, the contribution of the proposed algorithm, called Kennard-Stone Balance Algorithm (KSB) is twofold: KS is modified so that it can operate in data stream mining environment using a sliding window for incremental data sampling and incremental learning. The incremental version of KS is modified into KSB which takes into account of the class-imbalance problem by using reservoir sampling and round-robin in selecting an approximately evenly distributed data samples over different class.

By the original design of Kennard–Stone [23] principle, this technique picks a set of objects over a space that is which are 'uniformly' distributed into a subset of candidates. In the simplest means, it could be done by pure random selection. However, from a given dataset, KS is supposed to pick only the most representative samples. How 'representative' is defined then? In the context of selecting a representative data subset from a very large training dataset, the selected subset should cover most if not all the data objects which would be useful for model induction – so-called machine learning. Assuming the objects in the space are evenly distribution, representative objects should be those that resemble the original structure of the data group in the original dataset which should follow a similar selection coverage as in Fig. 3(b).

In order to achieve such selection pattern, it is expected that KS would include some extreme objects located at the boundary of the data space for preserving approximately the outline of the data group, then the selection would go on towards the centre of the space for portraying the rough shape of the original data cluster. In general, this kind of selection algorithms are referred as uniform designs (e.g. sphere exclusion, maximum dissimilarity) that commonly have the following working steps a stepwise procedure:

1 Compute the mean of the original data group, and compute the distance between each pair of the data objects;
2 Start putting a pair of objects which are farthest apart into the candidate subset;
3 Compute the dissimilarity between the remaining objects in the original dataset, as well as the selected objects already in the candidate subset;
4 By considering the dissimilarity, pick the object from the original dataset, which is has the maximum dissimilarity to the already selected objects in the candidate subset;
5 Repeat the picking by returning to step 3 until no more objects are available in the original dataset or the required number of objects in the candidate subset is fulfilled.

KSB is extended from KS which is a family member of the data sampling algorithms. KS and KSB share the same working logics of uniform design as listed above. It works with n-dimensional objects which could be thought of rows in a 2D data matrix of $m \times n$ characterized by m attributes and comprised of n rows. In data stream mining environment, since we do not know exactly the overall distribution of the data instances, the incoming data are assumed as uniformly distributed in the search space. The data group in the space is not static but increasing in quantity as time passes. The data arrive segment by segment via a sliding window that delivers fresh data objects one window at a time. In each window session, we assume there are w objects where $w<<n$ and n could potentially be unbounded, nà∞. All the w objects from the originally given data set X(w, m) are potential candidates to be selected into the candidate subset of size K, where $w \leq K<n$ in a normal circumstance. By the uniform design, the central mean of the given data group is computed. So, the closest object to the mean can be found, and it could be the first object to be recruited into the candidate subset. The object nearest to the central point that is first included in the candidate subset is regarded as the most representative one. After that, the dissimilarity distance between each pair of objects is, and the pair that has the longest distance is selected and regarded as the second most representative. Those objects pivot the rough outline of the data group marking the most outer edge.

Subsequently, the remaining objects from the original set having the maximum distance away from the already selected objects are selected and added to the end of the rank list in the candidate subset. This sequential process is repeated until the end. KBS is formulated as follow by the guidelines below:

After the start-up where the first nearest central object is chosen, the next two objects from the original group are selected by sorting and picking the dissimilarity of the farthest apart distance in terms of Euclidean distance. Other distance measures could be used in lieu depending on the user's choice. The normalized mean distance for all the objects is computed by equations:

$$\boldsymbol{d_{ij}} = \| \boldsymbol{X_i} - \boldsymbol{X_j} \| = \sqrt{\sum_{k=1}^{m} \left(\boldsymbol{X_{ik}} - \boldsymbol{X_{jk}}\right)^2}$$

$$\overline{\boldsymbol{d_i}} = \frac{\sum_{j=1}^{n} \boldsymbol{d_{ij}}}{\boldsymbol{n} - 1}$$

$$\widetilde{\boldsymbol{MeanDistance}}^{\boldsymbol{Normalized}} = \frac{\| \boldsymbol{MeanDistance} - \boldsymbol{Min}\{\boldsymbol{MeanDistance}\} \|}{\| \boldsymbol{Max}\{\boldsymbol{MeanDistance}\} - \boldsymbol{Min}\{\boldsymbol{MeanDistance}\} \|}$$

where dij is the distance score between two compounds, and $\overline{d_i}$ is the mean distance.

Once the distances are sorted, the object which has the maximum dissimilarity which may be interpreted as having the greatest minimum distance on top of the sorted list, from each of the recently chosen objects in the previous round is selected and placed in the candidate list. This is a general step for standard KS operation.

However, for KSB balancing the class distribution over the selected objects must be taken into consideration. Instead of selecting the object right from the whole original data group according to the sorted rank list of dissimilarity, class distribution acts as primary factor.

Mathematically, the logics of KSB is formulated as follow.

In KS, the consecutive objects are selected sequentially, according to the squared dissimilarity to the objects that are already chosen into the candidate subset. In both KS and KSB, the first object is the nearest central object, and the next two objects are chosen because they are farthest apart from each other. In KS, the forth and future selected objects are those farthest from the first three objects, and so on. Supposing that k objects where k⊆K have already been chosen ($k<w$), the ($k+1$)th object in the original dataset to be chosen using the criterion:

$$\max_{k<j\leq w}\left(\boldsymbol{min}\left(\boldsymbol{d}_{1\alpha},\ \boldsymbol{d}_{2\alpha}\ ,\ \ldots\ \boldsymbol{d}_{k\alpha}\right)\right)$$

Fig. 5. (Top) The experimental testbed in our semi-soundproof laboratory. (Bottom) Calibrating the sensors using a doll.

where dij, and $i = 1,\ldots,k$, are the squared dissimilarity in terms of Euclidean distances away from an available object, α, in the original set, that is not yet assigned into the candidate subset, to the k objects that are already assigned in the candidate subset. The process is very simple for KS. However, in KBS, an extra step is needed in assigning an available object from the original set into candidate subset. The whole original dataset is logically partition into C subsets called Class Distribution sub-groups, cdi, $\forall$ i⊆C, where C is the number of class labels in the original dataset. For example, $C = 2$ for binary classification where the class labels are only true with cdtrue and false with cdfalse.

During the $k + 1$ next object selection, it goes in round-robin fashion to examine for qualified available object from each of the cd sub-group, taking turn one at a time. So, in each round, a fair chance is given to each cd subgroup in selecting the available object from that particular subgroup using the same maximum dissimilarity criterion. The consideration is local pertaining only within the sub-group that is currently given its turn – this is round-robin selection mechanism, similar to the token-ring or FDDI strategy once popular in the 80′s. Details of the KSB formulation for batch supervised learning and incremental supervised learning (via data stream mining algorithms) are given in Appendix. The pseudo-code of KSB is given as follow:

Import original dataset A

Set K value for KSB class = A[*class*].value

Select the two samples with the farthest Euclidean distance to put them into the B dataset

*B*_max*_distance_sample_num* = find_max(*A*)

B_selected_sample_num.appand(*B*_max*_distance_sample_num*)

Remove the two samples with the farthest Euclidean distance found in the last step from the sample set A

A = np.delete(*A, B_selected_sample_num*, 0) for x in range(1, k):

Calculate the Euclidean distance from the remaining data in the sample set to the training set, and find the sample with the smallest distance for min*_distance_num* in range(0, *A*. shape[0]): min_distance_to_selected_samples = find_min(*A*)

The values (a set of numbers) in the remaining samples that are closest to the training set are arranged according to the highest to lowest distance to form an array min*_distance_to_selected_samples_list* = max(min*_distance_to_selected_samples*)

Take the first i ($i<k$) data of the array as the candidate data set of the data with the largest Euclidean distance min_distance_to_selected_samples_list = min_distance_to_selected_samples_list[:*i*] chosen_lable_random_num = random(0,*i*) max_distance_sample_number = find_max2(min*_distance_to_selected_samples_list, chosen_lable_random_num*)

If the class of the two selected point in max*_distance_sample_number* is the same

Then give up this result, find again

Else *B_selected_sample_num*.appand(max*_distance_sample_number*)

A = np.delete(*A, B_selected_sample_num*, 0)

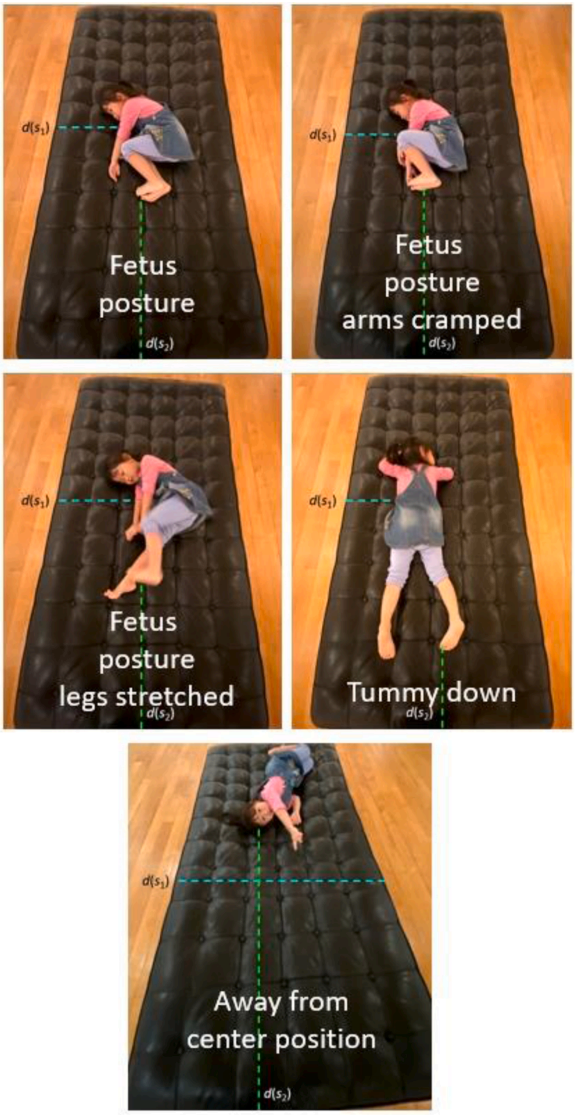


Fig. 6. A reenactment of how two sonar sensors are used to measure the proximity of a sleeper on a bed.

remaining_sample_num = np.delete(*remaining_sample_num, selected_sample_num*, 0)

4. Experiments

A collection of computer simulation experiments is conducted for verifying the practical possibilities of IoT sonar sensor bed. We would start as a preliminary motive to test sonar sensor setting with only two sensors, one placed at the side, another placed at the end of the bed. Scaling up the hardware with more sensors is a matter of investment issue trading off between cost and effectiveness. However, using only the minimum number of sensors possess a computational challenge on the software part. Since it is to be used for healthcare applications as well as general uses, the machine learning model must be both efficient and effective. As such, we need the highest possible accuracy, lowest false alarm rate, fastest learning speed, and most reliability in the machine learning model. These performance indicators much rely on the setup of the pre-processing algorithm and the use of the right machine learning (ML) algorithm. Not all ML algorithms are designed the same [24]; the input data and the underlying structure of the data patterns play a role in deciding the ML performance. This depends on how well the pre-processing methods couple with the ML algorithm, and how effective this combination would work on the incoming data. The pre-processing method is targeted at solving the big data problem in relation to the bulkiness of the data that slows down the supervised learning. It helps filter off the ordinary data for refining the machine learning ability. On the other hand, some ML algorithms were built more complex than the other; it is unknown about how well these ML algorithms would perform on streaming data that are collected from IoT sonar sensor bed. The objective of the

Table 2
Accuracy of various ML models using different pre-processing methods for IoTSS without sound recording.

13

	Decision Tree (DT)	Hoeffding Tree (HT)	Ramdom Forest (RF)	Support Vector Machine (SVM)	Neural Network (NN)	Deep Learning (CNN)
Raw	45.4369	33.7033	44.8027	45.4369	36.2403	30.1092
KS	75.7	61.9	74.6	73.5	71.8	67.4
KSB	79.4	62.5	84.4	80	71.4	69.6
K-means	69.6089	63.9359	69.2037	68.7984	65.0106	60.2185
Estimation Maximisation (EM)	45.7188	33.7033	45.1903	44.3798	37.6145	30.444
FarthestFirst	45.525	33.7033	44.9436	45.2079	36.5046	30.8844
	Naviie Bayes (NB)	K-nearest Neighbour (KNN)	JRipper	Bagging	Adaboost	
Raw	31.6068	44.327	41.8428	44.8555	35.9584	
KS	61.8	72.4	74.1	75	68	
KSB	62.5	83.3	76.7	78.9	67.4	
K-means	60.1128	68.9746	55.4087	69.3094	63.9359	
Estimation Maximisation (EM)	32.6815	44.7322	41.6667	45.2255	35.9584	
FarthestFirst	31.5187	44.5913	41.6138	45.0317	35.9584	

experimentation is hence twofold. A collection of popular and classical ML algorithms is chosen [25,26]. They are put under test of simulating a situation of several sleeping patterns.

4.1. Simulation setup

The data collection experiment was conducted in a semi-sound-proof room with a table of dimension 125 cm × 80 cm where the subject lies at the center. The subject is a three years old girl, weighs 14 kg and 102 cm tall. Two sonar sensors, installed one at each side, front and bottom. The sensor that is installed at the side, facing the front of the subject is called Sensor #1, and the other sensor installed near the bottom is called Sensor #2. Each sensor is placed at approximately 30 cm from the subject, from the body and from the feet respectively. Fig. 5 shows the experimental environment.

The experiment started by requesting the subject to perform a series of basic sleeping postures, such as 1) lying face up straight and remaining still (face-up-still), 2) lying face up straight and slightly wiggling the body (face-up-moving), 3) roll the body side-way to the left and remaining still (sideway-left-still), 4) roll the body side-way to the left and wiggling the body (sideway-left-moving), 5) turn the body over face down and remaining still (facedown-still), and finally 6) sitting up (sitting-up). Each action lasts for approximately from half to one-minute duration. The data are collected in raw data mode without any modulation or demodulation. The data collected for subsequently delivered to pre-processing by KBS and others, then to inducing a ML model. The built model is then tested in 5-fold cross-validation for generating the performance results. The software that we used is customized for middleware operation, written in C++; the data mining is carried out using RapidMiner Studio version 9.4 in Windows 10 environment.

Two experimentation setups are constructed, one with sound recording added on and one with audio muted. The motivation is to test this sonar sensor setup in a mono-modal environment using the simplest setting of two adjacent sensors, and the other one in a multi-(or trio)-modal environment. Sound is known to be a useful mean to detect and measure sleep-disorder related symptoms such as snoring, sinus, breathing difficulty and excessive body movement during sleep. There are some mobile App available on the market which claim to be able to measure the quality of sleep based on recording sounds during sleep. When a body turns, certain sounds would be generated even at the slightest decibel level, from frictions between the body and the bed. It is generally known that for example, prolonged and successive body wiggling on bed means the person is not asleep or some deep sleep is not attained yet, and vice versa. In our experiments, we conducted a control experiment. One group only has sonar sensor monitoring, and the other group adds the captured sleep sounds (including all the sounds produced during sleep, such as the sound of people rubbing the bed sheet when changing sleeping positions). In this regard, we have imitated many sleep detection mobile phone app on the market. They can only base on the captured sound at night to generate the user's sleep report. The use of two sonar sensors for measuring the distances that infer the sleep postures of a sleeper is shown in Fig. 6.

The performance results with respect to using different ML algorithms and sampling techniques are charted in Figs. 9a–13a, when sound is not included in the experiment. The same are charted in Figs 9b–13b when sound is inputted as one of the data channels in the experiment. The accuracy results are tabulated in Tables 2 and 3 for clarity, for cases without and with sound recording respectively. The raw data streams which are collected from sensor #1 – side sensor, and from sensor #2 – bottom sensor, are visualized in Fig. 7. One can see that the data pattern generated by sensor #1 manifest rough shapes of sleep posture in terms of distance between the subject and the sensor; for example, when the subject turns the body and sleeps on sideway, the distance lengthened. It is more apparent when the sensor data pattern is plotted on a step-chart as in Fig. 8 where each labelled posture carries the average distance value.

Table 3
Accuracy of various ML models using different pre-processing methods for IoTSS with sound recording.

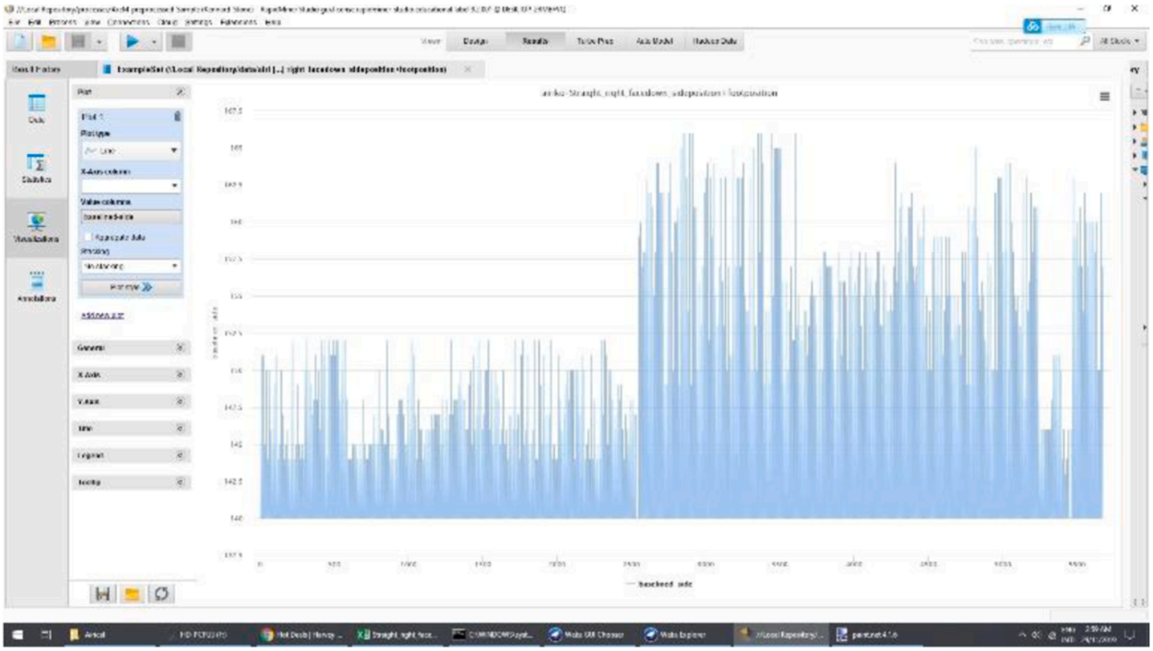
	Decision Tree (DT)	Hoeffding Tree (HT)	Ramdom Forest (RF)	Support Vector Machine (SVM)	Neural Network (NN)	Deep Learning (CNN)
Raw	75.3524	52.2375	70.0846	64.6758	66.3319	50.5462
KS	82.6	59	81.2	57.3	71.7	63.3
KSB	84.7	59.2	91.7	65.5	71.9	63.3
K-means	75.3524	44.9436	70.2079	66.2967	66.2086	50
	87.3326	82.1705	84.4433	85.8703	83.1924	74.771
	87.4912	83.351	84.549	85.8351	83.6328	73.9782
Estimation maximisation (EM)	75.4405	52.2375	70.1903	65.2925	66.4905	52.0965
FarthestFirst	77.537	52.2375	72.8682	66.9309	68.3756	52.2199
	Naviie Bayes (NB)	K-nearest Neighbour (KNN)	JRipper	Bagging	Adaboost	
Raw	44.8908	67.4595	75.0352	74.5948	51.938	
KS	58.4	75.9	76.5	81.7	52.9	
KSB	56.9	89.7	80.8	84.1	52.9	
K-means	44.8908	67.4595	75.2819	74.5948	51.938	
	74.031	83.7209	87.2445	87.544	51.938	
	74.8943	84.2142	87.914	87.4912	63.9183	
Estimation maximisation (EM)	44.8908	67.4595	75.0705	74.5772	51.938	
FarthestFirst	48.432	70.3312	76.3214	76.9732	51.938	

5. Results and discussion

The ML performance in terms of the following indicators are charted: accuracy which is simply the number of correctly classified instances over all the tested instances is graphed in Fig. 9; kappa statistics which is usually referred to how well the trained model is generalized to have good effects on different and new testing datasets - some scholars have taken kappa statistics as model reliability indicator, it is plotted in Fig. 10; False positive (FP) rate which is interpreted as false alarm rate - the number of times a testing instance which is normal but mistaken as alarmingly abnormal, is plotted in Fig. 11; ROC which stands for Receiver Operating Curve is a balanced error with respective to different proportions of testing data, it is measured by considering fully the true positive rates and false positive rates, precision, recall etc., the results are plotted in Fig. 12. The time performance which is considered to be the latency incurred as performance overhead when the monitoring is to be used in real-time. It has a significance in selecting an appropriate ML algorithm which does not take too long to induce a model, the results are plotted in Fig. 13.

From all the radar charts, both with and without sound channel added, pre-processing does have an edge over the performance comparing with the cases without any pre-processing. In other words, if the data streams of sonar without sound inclusion is taken from raw directly to inducing a ML model, the accuracy is not go much beyond the level of 45% regardless which ML algorithm is used. In contrast, the accuracy levels of a ML model rise up 84.4% and 91.7% in the case of without and with sound added respectively, by using pre-processing method KSB. It can be seen also that out of the tested pre-processing methods, KSB has its certain advantages in uplifting the performance in various scenarios. With respect to the choice of ML algorithm, in general decision tree type of algorithms such as CART decision tree, JRipper and random forest do offer pretty high levels of accuracy; so, does simple type of algorithms such as K-nearest neighbor algorithm (KNN) that is both fast and accurate. When it comes to ML model reliability, in terms of Kappa, KSB shows its superiority as in Fig. 10a and b. The same goes to false-alarm rate in Fig 11b with sound recording included. In general, with the inclusion of sound recording, overall performance improved for all the methods which shows of the complimentary effect of multi-modal relevant information inputs. With respect to speed of model induction, in general, almost all algorithms are good especially those of data stream mining type (incremental learning, such as KNN); but CNN and NN take considerable long time in model induction, yet not significant performance increase is gained. In summary, KSB that coupled with KNN offer relatively best performance. RF is good too except that RF is known to have consumed a large amount runtime memory.

The research concept proposed and reported in this paper, although backed by extensive ML experiments conducted and results validated, is a first step towards the application of sonar sensors onto a monitoring bed. Conceptually and practically this experimentation demonstrates its possibility of using the simplest setup of a sensor pair for distinguishing six different sleeping postures. However, monitoring sleep patterns is a complex task. Overviewing the current approaches on the market, a common method is by sound recording which offers the most basic but essential of information about the intensity of a sleep or sleep quality, namely awake, shallow sleep and deep sleep. There is totally no information about the posture nor movements of the sleepers besides the noise that the sleeper generated. IoTSS has taken a step further in capturing those postures and movements and providing feedbacks about them to the user. Although simple postures could be recognized by only two sensors with the aid of ML algorithms. It is anticipated that array of sensors should be used for increasing the recognition resolution. The postures by different parts of body, such as head (including face and nose), chest, and abdomen have certain significance in medical domain. For example, a prolonged face down position or limbs being tightly curled up for too long are not of healthy habit. In light of this, further works shall include a knowledge module which interfaces between the ML model and the end-users. The postures and movements that are detected and recognized from sensors and ML algorithms, shall be presented in the user's sleep report simply and clearly. The end-user could read their sleep report as soon as



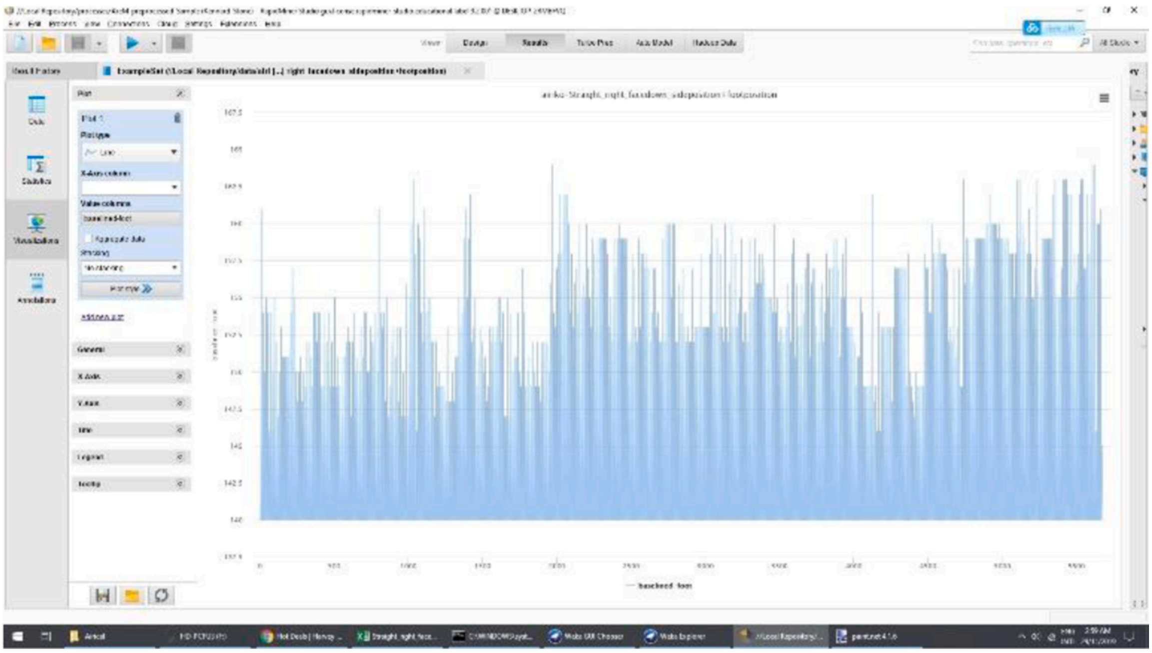


Fig. 7. Visualization of (top) sensor #1 data from the side of the bed, and (bottom) sensor #2 data from the bottom of the bed.

they wake up. The report information should include how many postures happened in their sleep and the starting time and duration of each posture. If sleep is recorded every day, this can help the user find out when the user's sleep quality is highest and lowest, and which posture is the most suitable sleep for the user. Importantly, if the user suffers from certain disease, such as asthma, it can help patients monitor the quality and posture of sleep at night, to avoid improper sleeping postures affecting breathing and threatening the life and health of patients. The work reported in this paper lays a cornerstone for applying sonar sensor to in-home health monitoring, it focused on the ML model and solving the big data stream mining issues.

6. Conclusion

Measuring sleep activities is an important approach to assessment of sleep disorder and general healthcare. In this paper, a novel

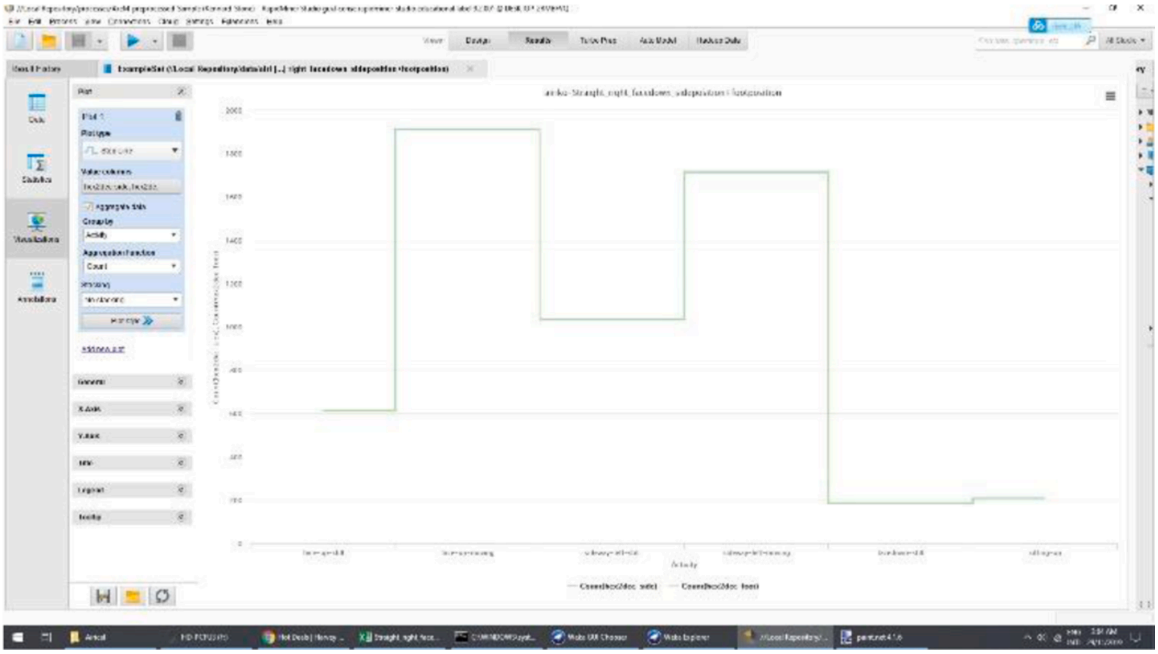


Fig. 8. Step-chart of sensor #1 data from the side of the bed.

Fig 9. (a) Accuracy of various ML models using different pre-processing methods for IoTSS without sound recording. (b) Accuracy of various ML models using different pre-processing methods for IoTSS with sound recording.

remote sensing technique is proposed, called IoT Sonar Sensor (IoTSS). It works as a motion sensor which picks up sleeper's activities on bed. Though the hardware configuration is simple, the cost-effectiveness is studied in this project where the efficacy of IoTSS is to be enhanced using ML model. Simulation results are analyzed and reported that randomforest ML algorithm produced the best results, but it is memory demanding. An alternative ML approach which is also cost-effective is incremental learning by k-nearest neighborhood. Furthermore, a new data-pre-processing method, suitable for relieving the computational requirement for the task of inducing a ML from continuous multi-modal data streams, called Kennard-stone Balance algorithm is presented. Our proposed approach is able to increase the ML accuracy up to 91.7% while the accuracy level remains at maximum about 50% by using only raw data streams for ML, based on empirical test-bed data collected. Further works should be extending from two sensors to array of multiple sensors, data fusion, attempting recognition of more variety and complexity of sleeping postures and expanding our dataset with more features.

Author statement

Tengyue Li: Writing – review & editing, data curation, software, validation, visualization
Yaoyang Wu: Roles/Writing – original draft;
Xuqi Li: Formal analysis.
Sabah Mohammad: Supervision,

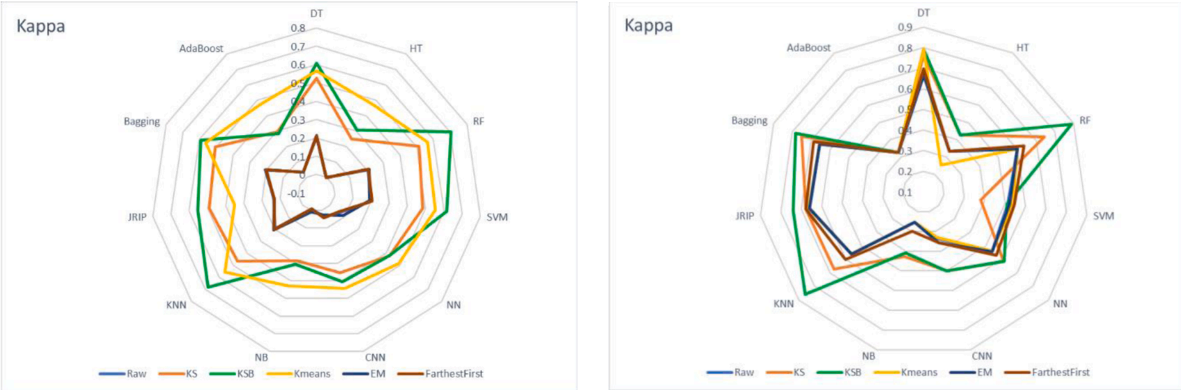


Fig 10. (a) Kappa statistics of various ML models using different pre-processing methods for IoTSS without sound recording. (b) Kappa statistics of various ML models using different pre-processing methods for IoTSS with sound recording.

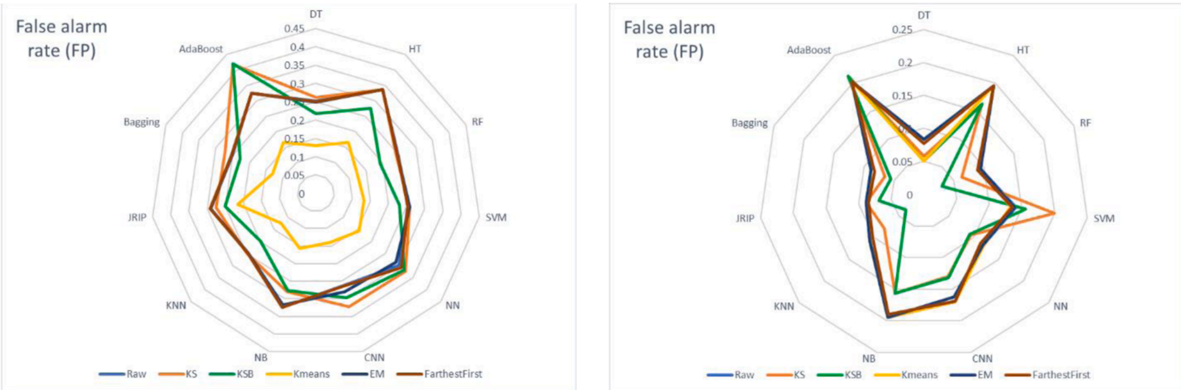


Fig 11. (a) False alarm rate of various ML models using different pre-processing methods for IoTSS without sound recording. (b) False alarm rate of various ML models using different pre-processing methods for IoTSS with sound recording.

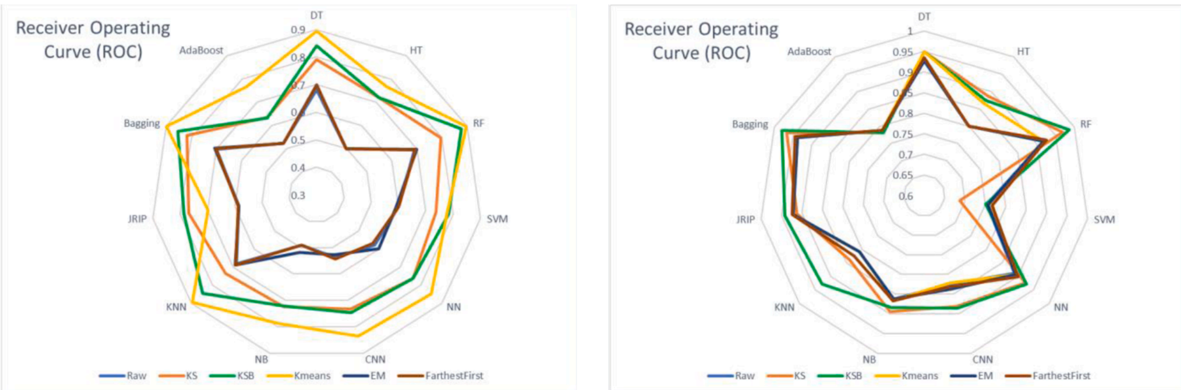


Fig 12. (a) ROC of various ML models using different pre-processing methods for IoTSS without sound recording. (b) ROC of various ML models using different pre-processing methods for IoTSS with sound recording.

Raymond K. Wong: Project administration,
Kok-loeng Ong: Conceptualization,

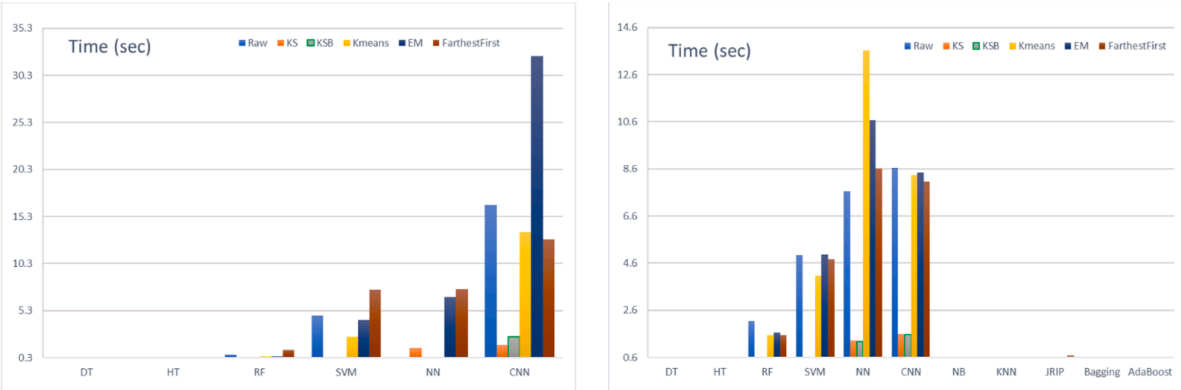


Fig 13. (a) Time performance of various ML models using different pre-processing methods for IoTSS without sound recording. (b) Time performance of various ML models using different pre-processing methods for IoTSS with sound recording.

Declaration of Competing Interest

The author(s) declare(s) that there is no conflict of interest.

Acknowledgement

The authors are thankful for the financial support from the research grants, MYRG2016-00069, entitled 'Nature-Inspired Computing and Metaheuristics Algorithms for Optimizing Data stream mining Performance', EF003/FST-FSJ/2019/GSTIC, code no. 201907010001, FDCT/126/2014/A3, entitled 'A Scalable Data Stream Mining Methodology: Stream-based Holistic Analytics and Reasoning in Parallel' offered by FDCT and RDAO/FST, the University of Macau and the Macau SAR government.

Appendix

In this Appendix the Kennard-Stone Balance Algorithm for Time-series is described in detail. Firstly it explains how the traditional algorithm, Kennard-Stone(KS) sampling algorithm, is introduced into batch learning from time series data, and then the extended version called Kennard-Stone Balance(KSB) Algorithm is introduced and applied to solve the imbalanced class problem. At the end, KS and KSB are modified into a data stream mining version.

1. Kennard-Stone (KS) sampling algorithm in batch learning

Suppose that the time series dataset is denoted as $\boldsymbol{D} = \{x_1, x_2, \ldots, x_n\}$ where$x_i$and $x_j(i, j = 1, \ldots, n)$ are data in D. Since the dataset D is a time series dataset in which all the data are order by time, this means if $i > j$, then $x_i$is generated later than x_j. Each data has m attributes, $x_j \in \boldsymbol{R}^m(j = 1, \ldots, n)$ and R is the notation of rational number.

Definition 1. Total distance

There is a specific data $x_k \in \boldsymbol{R}^m$, the total distance of x_k and T is to sum up the Euclidean distances between x_k and all the elements in dataset T, we define a function:

$$\boldsymbol{TDis} : \boldsymbol{R}^m \times \boldsymbol{T} \to \boldsymbol{R}$$

$$\boldsymbol{d} = \boldsymbol{TDis}(x_k, \boldsymbol{T}) = \sum_{x_j \in \boldsymbol{T}} ||x_k - x_j||_2$$

Definition 2. Kennard-Stone (KS) sampling algorithm

D= $\{x_1, x_2, \ldots, x_n\}$ is a time series dataset, the Kennard-stone algorithms will pick up the most representative data from the time series D to form a training set T for training the machine learning models. In KS algorithms, we denote an empty set as initial value of $\boldsymbol{T}^0$, the sampling rate r will be set as hyper-parameter to control the size of training dataset. So the high level model of Kennard-Stone is defined,

$$\boldsymbol{KS} : [0, 1] \times \boldsymbol{D} \times \boldsymbol{D} \to \boldsymbol{T}$$

$$\boldsymbol{T} = \boldsymbol{KS}\left(\boldsymbol{r}, \boldsymbol{D}, \boldsymbol{T}^0\right)$$

Henceforth, we describe the algorithms in detail.

KS starts by finding out two data, $xm_1 and xm_2$, between which has larger Euclidean distance than that of any two data among the dataset D.

$$\boldsymbol{xm}_1,\ \boldsymbol{xm}_2 = \arg\max_{x_i, x_j \in \boldsymbol{D}} \{\| \boldsymbol{x_i} - \boldsymbol{x_j} \|_2\}$$

These two data will union with $\boldsymbol{T}^0$ as $\boldsymbol{T}^1$.And the next step of KS is to use the Total distance (Definition 1) to split the dataset D and extend the subset T iteratively until the capacity of T is maximal.

$$M = \begin{cases} KS\left(r,\ D,\ \left\{\arg\max_{x_i,\ x_j \in \boldsymbol{D}} \{\| |\boldsymbol{x_i} - \boldsymbol{x_j} \|_2\}\right\}\right)\ if\ t = 0 \\ KS(r, D, T^{t+1})where\ \ T^{t+1} = \mathrm{T}^{t} \ \cup \left\{\arg\max_{x_k \ \in D \backslash T^t} TDis(x_k,\ T^t)\right\} \\ T^t if |T^t| \geq |D| * r \end{cases}$$

This means that we will iteratively search for the new data among D which has largest total distance with the elements in T and then union this new data with T to continuously find the next elements until the capacity of T reaches. Then the final training set T is what we want.

2. Kennard-Stone Balanced (KSB) sampling algorithm in batch learning

In order to settle the imbalanced class problem in data mining caused by KS algorithms, a new version named Kennard-Stone Balanced (KSB) sampling algorithm is proposed by extending KS. This algorithm will balance the class in T. The key innovation is to consider the class of the new data at each turn of searching. Then the Total distance model will be modified into an advanced version

Definition 3. Adaptive Total distance

Given the dataset T, the number of class L, adaptive total distance is to measure how dissimilar the data $\boldsymbol{x_k}$ and T under the specific the round of turns ct.

$$\boldsymbol{ADis}\ \ : \boldsymbol{R^m} \times \boldsymbol{T} \times \boldsymbol{R} \to \boldsymbol{R}$$

$$d = \ \boldsymbol{ADis}\ \ (\boldsymbol{x_k}\ ,\ \boldsymbol{T},\ \boldsymbol{ct}) = \left(\boldsymbol{\alpha} * 1_{\boldsymbol{l(x_k)=ct\%L}} + (1-\boldsymbol{\alpha})^{1-1_{l(x_k)=ct\%L}}\ \right)\left(\sum_{\boldsymbol{x_j \in T}} \left|\left|\boldsymbol{x_k} - \boldsymbol{x_j}\ \right|\right|_2\right)$$

where $\boldsymbol{\alpha}$ is the adapted weight for the data respect to current turn ct and dataset T which has an updated function g and hyper-parameter $\boldsymbol{\beta}$:

$$\boldsymbol{\alpha} = \boldsymbol{g}(\boldsymbol{T},\ \boldsymbol{x}_k) = \ \frac{\boldsymbol{\beta}}{\boldsymbol{rate}(\boldsymbol{T},\ \boldsymbol{l}(\boldsymbol{x_k})) + 0.1}$$

while $1_{\boldsymbol{l(x_k)=ct\%L}}$ is an indictor function to measure whether the class of $\boldsymbol{x_k}$ match with the corresponding class $\boldsymbol{ct}\%\boldsymbol{L}$.%

$$1_{\boldsymbol{l(x_k)=ct\%L}} = \ \begin{cases} 1 & if\ \ l(\boldsymbol{x_k}) = ct\%L \\ 0 & otherwise \end{cases}$$

where rate(X, y) is a function to calculate the rate of class y in dataset X. This function makes $\boldsymbol{\alpha}$ be dynamic with the change of current data$\boldsymbol{x_k}$and $\boldsymbol{T}$.

This modification guarantees that the algorithm will be searching for the matching-class data at fair corresponding turns. Meanwhile, because the weight $\boldsymbol{\alpha}$ will also give a selected chance to that no matching data but with huge distance and minority. Therefore, we update the KS model into**KSBmodel**.

Definition 4. Kennard-Stone Balanced (KSB) sampling algorithm in batch learning

Given sampling rate $\boldsymbol{r}$, dataset D, last round training dataset $\boldsymbol{T^t}$***andthenumberofsearchingroundct***, KSB will search for the most suitable data in $\boldsymbol{D} \backslash \boldsymbol{T^t}$ respect to $\boldsymbol{T^t}$ using adaptive total distance(Definition 3) and the stopping rule is the size of training dataset T.

$$\mathbf{KSB} : [0,1] \times \boldsymbol{D} \times \boldsymbol{D} \times \boldsymbol{R} \to \boldsymbol{T}$$

$$\boldsymbol{KSB}(r, \boldsymbol{D}, \boldsymbol{T}^t, ct) = \begin{cases} \boldsymbol{KSB}(r, \boldsymbol{D}, \boldsymbol{T}^{t+1}, ct+1)where\ \ \boldsymbol{T}^{t+1} = \mathbf{T}^{t}\ \cup \left\{\arg\max_{x_k \in \boldsymbol{D} \backslash \boldsymbol{T}^t} \boldsymbol{ADis}(x_k, \boldsymbol{T}^t, ct)\right\} \\ \boldsymbol{KSB}\left(r, \boldsymbol{D}, \left\{\arg\max_{x_i, x_j \in \boldsymbol{D}} \{\| \boldsymbol{x_i} - \boldsymbol{x_j} \|_2\}\right\}, ct+1\right) if\ t = 0 \\ T^t if\ |\boldsymbol{T}^t| \geq |\boldsymbol{D}| * r \end{cases}$$

The setting of dynamic $\boldsymbol{\alpha}$ and the number of turns ct makes the selection more balanced, by giving fair chances to the minority class when iteratively searching runs in round-robin turn. Therefore, in KSB algorithm, the final Training set T is obtained where in initial parameter $\boldsymbol{T}^0$ is empty set and the number of searching round is 0 and $\boldsymbol{r}$ is the hyper-parameter:

Table 4

Table 4
Definition of the symbols used in the KS and KSB formula.

D	Dataset	$l(x_k)$	The label(class) of x_k
$x_1,x_2,\ldots,x_n$	Data	$rate(T,\ class_l)$	To measure how much percentage $class_l$ occupy in dataset T.
R	Rational number	$1_{l(x_k)=ct\%L}$	Indicator function
T	Training dataset	%	Complementation operation
$TDis(\bullet)$	Function of total distance	$KSB(r, D, T^t, ct)$	Kennard-Stone Balanced algorithm which has four inputs r, D, T^t, ct
$\|\| \bullet \|\|_2$	Euclidean distance	$ADis(\bullet)$	Adaptive Total distance
KS($\bullet$)	Kennard-Stone algorithm which has three inputs	α	the adapted weight
r	Sampling rate	$g(\bullet)$	updated function for α
$T^0,T^1,\ldots,T^t$	The training set obtained by 0th, 1st and tth round searching	ct	Counting the round of searching
xm_1,xm_2	First two data obtained by KS or KSB algorithms	β	Hyper-parameter to control the initial weight
$\| \bullet \|$	The size of dataset		

$$T = KSB\left(r,\ D,\ T^0, ct\right)$$

Kennard-Stone (KS-W) and Kennard-Stone Balanced (KSB-W) are in the incremental version of sampling algorithms in data steam learning. When the machine learning goes to data steam mining mode, the system will never get the whole dataset for training. Therefore, the time series data will be continuously collected in a buffer. Once it reaches the capacity, the data will be fed into the KS or KSB to generate the sampling subset for training. Subsequently, once the new buffer space arrives, in the form of a sliding window, the data steam algorithms will incrementally learn the parameters of the models which will not be discussed in the paper. However, even though KS and KSB could sample those most representative data from the buffer but some new class data may not exist in the current buffer but contained in the testing data. Therefore, in KS-W and KSB-W sampling, if the machine learning algorithm suffers from a significant drop in accuracy, the system will label it as new label and send it to the training buffer for training an updated model.

References

[1] Willinger M, James LS, Catz C. Defining the sudden infant death syndrome (SIDS): deliberations of an expert panel convened by the National Institute of Child Health and Human Development. Pediatr Pathol 1991;11:677–84.

[2] Changing concepts of sudden infant death syndrome: implications for infant sleeping environment and sleep position, task force on infant sleep position and sudden infant death syndrome. Pediatrics 2000;105:650. https://doi.org/10.1542/peds.105.3.650.

[3] Willinger M, Hoffman HJ, Wu K-T, et al. Factors associated with the transition to nonprone sleep positions of infants in the United States: the National Infant Sleep Position Study. JAMA 1998;280:329–35.

[4] Chokroverty S. Overview of sleep & sleep disorders. Indian J Med Res 2010;131:126–40.

[5] Yolba S, Yildirim A, Duzenci D, Gundogdu B, Özgen M, Serdar Koca S. Sleep quality, sleeping postures, and sleeping equipment in patients with ankylosing spondylitis. Turk J Med Sci 2017;47:1198–205.

[6] Kapsimalis F, Varouchakis G, Manousaki A, Daskas S, Nikita D, Kryger M, et al. Cytokines and pathological sleep. Sleep Med 2008;9:603–14.

[7] Reduce the Risk of SIDS & Suffocation, https://www.healthychildren.org/English/ages-stages/baby/sleep/Pages/Preventing-SIDS.aspx [Last accessed on 15 Dec 2019].

[8] Pereira KD, Roebuck JC, Howell L. The effect of body position on sleep apnea in children younger than 3 years. Arch Otolaryngol Head Neck Surg 2005;131(11):1014–6.

[9] Patino M, Sadhasivam S, Mahmoud M. Obstructive sleep apnoea in children: perioperative considerations. Br J Anaesth 2013;111(S1):i83–95.

[10] Kelly JM, Strecker RE, Bianchi MT. Recent developments in home sleep-monitoring devices. ISRN Neurol 2012:768794.

[11] Ibáñez V, Silva J, Cauli O. A survey on sleep assessment methods. PeerJ 2018;6:e4849. https://doi.org/10.7717/peerj.4849. ISSN 2167-8359.

[12] Heinrich A, Aubert X, de Haan G. Body movement analysis during sleep based on video motion estimation. In: 2013 IEEE 15th international conference on e-health networking, applications and services (Healthcom 2013); 2013. p. 539–43. https://doi.org/10.1109/HealthCom.2013.6720735.

[13] Pereira PP. Efficient IoT framework for industrial applications. PhD thesis. Lulea University of Technology; 2016.

[14] Saleem K, Bajwa IS, Sarwar N, Anwar W, Ashraf A. IoT healthcare: design of smart and cost-effective sleep quality monitoring system. J Sens 2020;2020. https://doi.org/10.1155/2020/8882378. Article ID 8882378, 17 pages.

[15] Eisenberg ME, Givony D, Levin R. Acoustic respiration rate and pulse oximetry-derived respiration rate: a clinical comparison study. J Clin Monit Comput 2020;34:139–46. https://doi.org/10.1007/s10877-018-0222-4.

[16] Solà-Soler J, Giraldo BF. Comparison of ECG-derived respiration estimation methods on healthy subjects in function of recording site and subject position and gender. In: 2020 42nd annual international conference of the IEEE engineering in medicine & biology society (EMBC); 2020. p. 2650–3. https://doi.org/10.1109/EMBC44109.2020.9176438.

[17] Jan HY, Chen MF, Fu TC, et al. Evaluation of coherence between ECG and PPG derived parameters on heart rate variability and respiration in healthy volunteers with/without controlled breathing. J Med Biol Eng 2019;39:783–95. https://doi.org/10.1007/s40846-019-00468-9.

[18] Perez-Pozuelo I, Zhai B, Palotti J, et al. The future of sleep health: a data-driven revolution in sleep science and medicine. npj Digit Med 2020;3:42. https://doi.org/10.1038/s41746-020-0244-4.

[19] Cai L, et al. OSA patient monitoring system based on the Internet of Things framework. In: 2019 4th international conference on smart and sustainable technologies (SpliTech); 2019. p. 1–4. https://doi.org/10.23919/SpliTech.2019.8783103.

[20] Kennard RW, Stone LA. Computer aided design of ex-periments. Technometrics 1969;11:137–48.

[21] Miller AJ, Nguyen N-K, Fedorov A. Exchange algorithm for D-optimal design. J R Stat Soc Ser C 1994;43(4):669–77.

[22] Rodionova O, Pomerantsev A. Subset selection strategy. J Chemom 2008.

[23] Ji W, Rossel RAV, Shi Z. Accounting for the effects of water and the environment on proximally sensed vis-NIR soil spectra and their calibrations. Eur J Soil Sci 2015;66:555565. https://doi.org/10.1111/ejss.12239.
[24] Artificial Intelligence vs. Machine Learning vs. Deep Learning, https://www.datasciencecentral.com/profiles/blogs/artificial-intelligence-vs-machine-learning-vs-deep-learning, [Last accessed on 15 Dec 2019].
[25] Issad HA, Aoudjit R, Rodrigues JJPC. A comprehensive review of Data Mining techniques in smart agriculture. Eng Agric Environ Food 2019. Elsevier.
[26] Teles G, Rodrigues JJPC, Saleem K, Kozlov S, Rabêlo RAL. Machine learning and decision support system on credit scoring, neural computing and applications. Springer; 2019. 1-18.

Tengyue Li is currently a PhD student at the University of Macau. She is also the Head of Data Analytics and Collaborative Computing Laboratory, Zhuhai Institute of Advanced Technology. Ms Li is an active researcher, manager and chief-knowledge-officer in DACC laboratory at the faculty of science and technology, University of Macau.

Yaoyang Wu is currently a PhD student at the University of Macau. She is also doing research in Zhuhai Institute of Advanced Technology, Chinese Academy of Science, Zhuhai, China.

Wu Feng is currently the director of Medical Device R&D department at Zhuhai Institute of Advanced Technology Chinese Academy of Science. Dr.Wu is an expert in field of biomedical engineering, sleep medicine engineering, aviation medicine engineering, etc.

Sabah Mohammed is full Professor at the department of Computer Science at Lakehead University (Ontario Canada) since 2002 and Core Professor at the PhD BioTechnology program at Lakehead. His research interest is in intelligent systems that have to operate in large, nondeterministic, cooperative, highly connected, survivable, adaptive or partially known domains.

Raymond Wong is an Associate Professor at the School of Computer Science and Engineering, UNSW and involved in several startup companies. He founded and led the database research group at National ICT Australia (The largest information and communication technology R&D organization in Australia). He has supervised more than 50 honours thesis students in 6 years, of which 6 of them obtained the University Medals.

Kok-Leong is an Associate Professor at La Trobe University, Melbourne, Australia. He has attracted over $1.46 m in research grant related to data analytics research and has served in over 50 Program Committees of international conferences related to analytics, including the prestigious KDD (2015 and 2017) and PAKDD (2006, 2018 and 2019).