2018 IEEE 16th Int. Conf. on Dependable, Autonomic & Secure Comp., 16th Int. Conf. on Pervasive Intelligence & Comp., 4th Int. Conf. on Big Data Intelligence & Comp., and 3rd Cyber Sci. & Tech. Cong.

Smart Crib Control System Based on Sentiment Analysis

Ying Liu
Software College
Northeastern University
ShenYang, China
liuy@swc.neu.edu.cn

Dequan Zheng
Software College
Northeastern University
ShenYang, China
ZDeeQ@outlook.com

Tongmao Lin
Software College
Northeastern University
ShenYang, China
lnrdlin@outlook.com

Xianqi Liu
Software College
Northeastern University
ShenYang, China
liuxq@swc.neu.edu.cn

Deshuai Wang
Xi Kang
Neusoft Company
ShenYang, China
wangdeshuai@neusoft.com

Frank Hopfgartner
Information School
University of Sheffield
Sheffield,UK
f.hopfgartner@sheffield.ac.uk

***Abstract*—One of the key selling points of smart home devices is that they provide solutions tailored to our needs. Identifying this need, however, is not always trivial, especially when dealing with infants who are not yet able to express their wishes using clear words. In this paper, we present preliminary work on identifying infants' needs based on categorizing their crying behavior. Our solution is embedded in a smart crib system which is designed to support parents in better understanding their babies' sentiment. The high accuracy of our experimental results are promising.**

Keywords—smart crib, sentiment analysis, support vector machine, system design, crying process.

I. INTRODUCTION

Triggered by the rapid development of digital technologies since the early 1990s and the accumulation and analysis of large amounts of data, we are currently witnessing the start of a new revolution in which artificial intelligence (AI) techniques are employed to take over manifold types of tasks for our convenience [1]. One of the most popular domains of AI is theso-called smart home, i.e., the combination of AI and the Internet of Things (IoT) in a connected household [2]. In fact, in recent years, a large variety of products for smart homes have entered the market such as smart refrigerators, smart air conditioners and similar devices, all of which promise us a more convenient and comfortable lifestyle. For example, smart refrigerators can automatically identify the type and amount of food in the refrigerator, and intelligently adjust the refrigerator temperature mode to keep the food fresh and even create recipes for users. Furthermore, as shown in [3], accompanying Apps or computer programs can allow users to easily check on the content of their refridgerator from anywhere in the world.

As Wilson et al. [9] discuss, no major study has been published on the actual users of smart home technologies. This is mainly due to the fact that advances in this field are mainly driven by industry who do no communicate their findings to the public. An analysis of their marketing strategies, however, suggest that they have identified two main target groups as potential users of their products: (1) Elderly people who can rely on assisted living technologies such as automatic fall detection systems, and (2) technology-savvy individuals who like to "play" with new technologies and gadgets. The largest group of technology-savvy consumers are the so-called Millenials, i.e., young people who are born between the 1980s and late 1990s who grew up experiencing digital technologies as "digital natives".

Many Millennials have now reached an age in which they are starting their own family. Therefore, it does not come as a surprise that various smart home applications are now entering the market that specifically address the needs of young parents. Examples include methods to measure babies' saliva, predictive health trackers, breast feeding assistants, or nutrition monitors that can help parents to keep track of their baby's development [14]. Other applications directly interact with the baby. For example, [4] offer a smart baby bed that automatically detects when the baby starts crying and tries to soothe him/her to sleep by playing white noise and by moving the baby mattress. Although both movements and white noise are well known factors that can help infants to calm down, this application ignores the fact that crying is a baby's basic mean of communication. In fact, as many parents will confirm, a baby cries for many reasons, including hunger, need of physical contact, tiredness, need for diaper change, and many other reasons [5].

In this paper, we address this limitation by proposing a smart baby crib that aims to identify the baby's sentiment to support parents in better understanding their baby's needs. As illustrated in Figure 1, the crib uses a set of sensors to measure the baby's weight, bed wetting, and, most importantly, analyses the infant's sentiment. While most sentiment analysis methods rely on natural language processing, text analysis, computational linguistics, or bio statistics to systematically identify, extract, quantify, and study emotional states and subjective information [6], we suggest to analyse babies' sentiment by analysing their crying patterns. To the best of our knowledge, this is the first work that aims to understand infants' sentiment based on their crying. Moreover, our smart grip is able to trigger various actions that parents would perform such as playing comforting

978-1-5386-7518-2/18/$31.00 ©2018 IEEE
DOI 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00047

IEEE computer society

sound, or shaking the crib, and additionally can inform the parents about their infant's needs.

The paper is structured as follows. In Section II, we provide an overview of the system architecture. Section III outlines the signal processing algorithm to analyze crying patterns. In Section IV, we summarize the implementation of our system. The experiment is introduced in Section V. Section VI concludes this work.

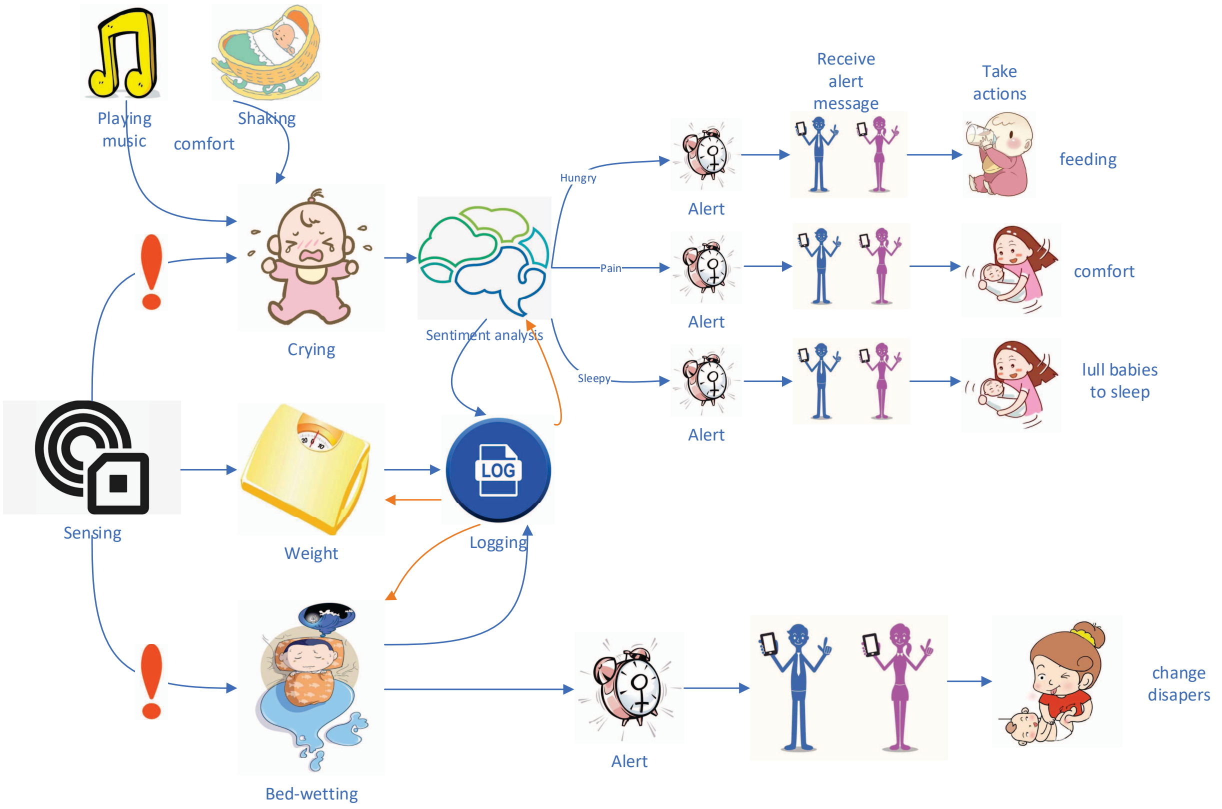


Figure 1. Functions of Smart Crib

II. System Design

A. System Architecture

As shown in Fig. 2, the system is divided into three components: hardware (mainly Raspberry Pi and sensors), server and mobile application.

- **Hardware:** The hardware components collects all data for processing. The main parts are various sensors to record different signals and a Raspberry Pi [7] which is used for processing. The Raspberry Pi has a small and powerful microcontroller that can handle complex data processing tasks. In addition, the Raspberry Pi also has many native sensor accessories that can be used for data processing tasks. Its built-in wireless network module enables it to interact with the server in real time.
- **Server:** The server serves as backend of the system architecture as it is the central hub used for data exchange. It contains various servlets that are employed to record infants' details such as weight and to present this information to the parents. The data is stored in a MySQL database [8].
- **Mobile Application:** The mobile Application allows parents to display the data that is stored on the server. The app runs in the background and frquently requests data from theserver. A voice command is triggered to draw parents' attention if needed.

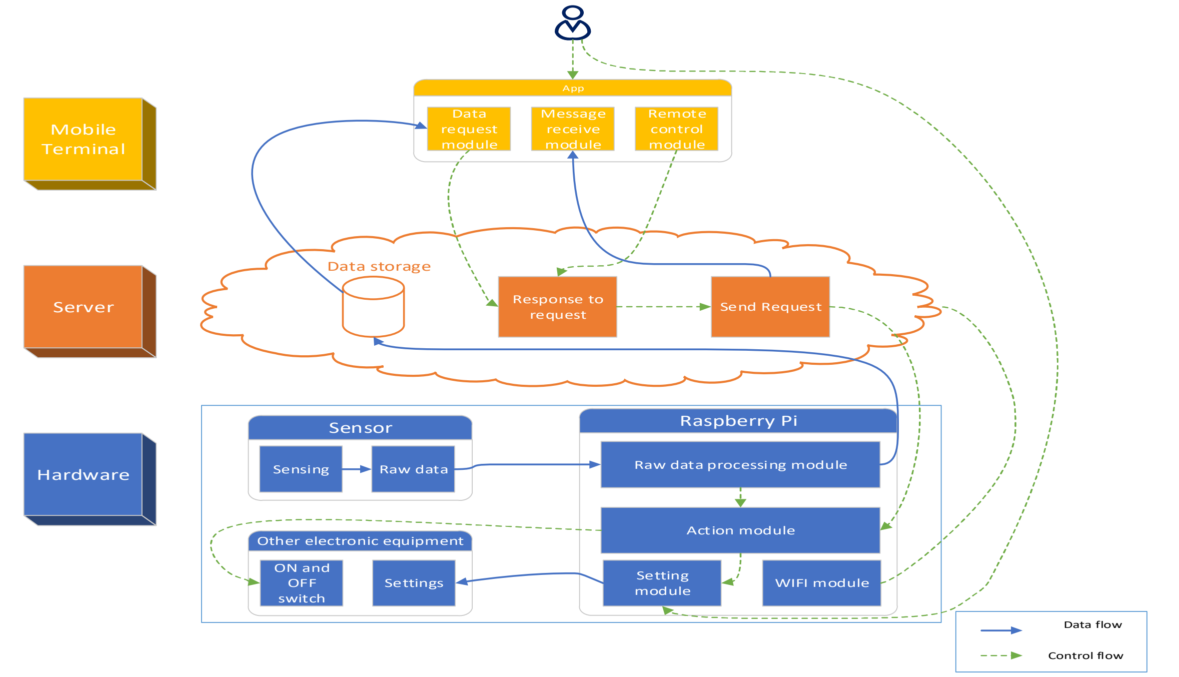


Figure 2. Architecture of Smart Crib System

B. *System Workflow*

As we can see in the workflow diagram in Fig. 3, the Raspberry Pi uses sensors to continuously collect data, including temperature, humidity, sound, and weight. This data is send to the server in frequent cycles and directly analyzed. Once the analysis is completed, the determined sentiment result is passed to the server for storage. If no action is required during one cycle, i.e., when no activity is recorded, the next cycle begins automatically. Once the Raspberry Pi determines that the baby is in an abnormal state (i.e. crying or bed wetting), it immediately starts the soothing mode to pacify the baby by playing music and by gently shaking the crib. In addition, the server sends a message to the mobile Application installed on the parents' smartphone, hence informing them what their baby might want to tell them.

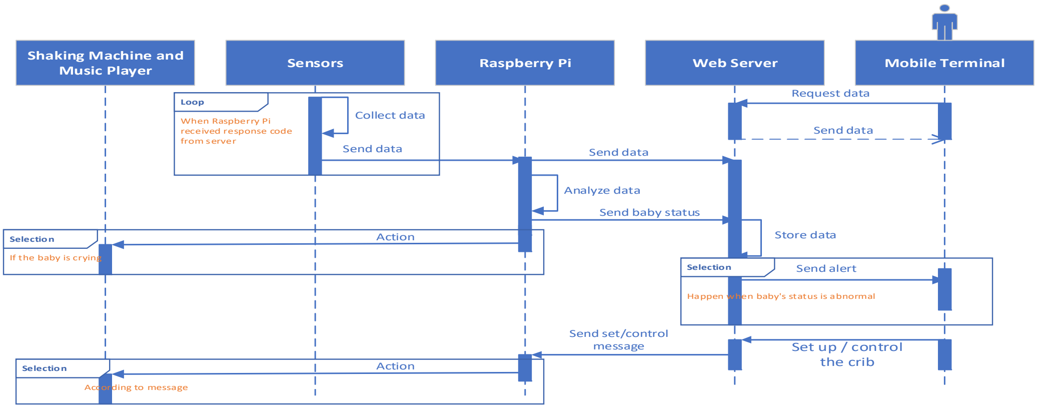


Figure 3. Workflow of Smart Crib System

III. Signal Processing Task

In this section we now describe the algorithms used to analyse the infant's cries in detail. As shown in Fig. 4, our method consists of five main steps.

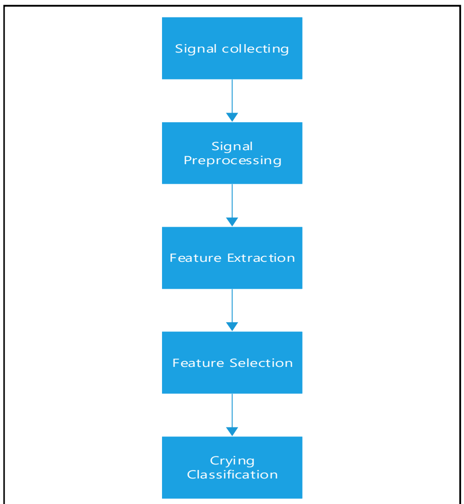


Figure 4. Algorithm workflow

A. Signal Collecting

The first step includes collecting the signal. We use a microphone to collect sound signals of the surroundings. This step does not only include collecting the raw data, but also normalizing it to reduce the difference between different crying parts. We record sound with a duration of ten seconds and normalize the collected signal by transforming it to WAV format with 16-bit resolution and a sampling rate of 8kHz.

B. Signal Preprocessing

In the signal pre-processing step, we remove unwanted noise and silence fragments from the initial signal. It includes the following steps:

1) *Framing:* In accordance to best practice to split larger signals into smaller bits, referred to as frames, we set the length of each frame to 256 sample points. In addition, we include an overlap of 50% between neighboring frames (i.e. 128 sample points) to avoid any negative effects caused by splitting up the signal into smaller bits.

2) *End-Point Detection:* In order to detect voiced segments, we also detect end-points, i.e., we remove silent pieces. There are various methods that can be employed to detect end-points, such as double-threshold detection based on short-time energy and short-time zero-crossing rate, or based on cepstrum features [10]. Here, we choose a single-threshold detection method based on intensity as it obtained better results than the double-threshold detection method. The intensity of the n-th frame is calculated as follows:

$$Intensity = \sum_{i=1}^{N} |S_n(i)| \qquad (1)$$

Where N is the number of sample point of frames, and S_i is the value of the i-th sample point.

Then we set the threshold as follow:

$$Threshold = \max_{1 \le i \le N} |I_n(i)| \times 0.1 \qquad (2)$$

Where N is the number of frames, and I_i is the intensity of the i-th frame.

3) *Detecting frames containing crying:* The next step required is to detect the crying signals from the voiced signal. Here we use a double-threshold detection based on short-time energy and short-time zero-crossing rates as suggest in [10]. We first determine the short-time zero-crossing rate of the n-th frame and the short-time energy of n-th frame using the following equation:

$$ZCR_n = \frac{1}{2} \sum_{i=1}^{N} |sgn(S_n(i)) - sgn(S_n(i-1))| \qquad (3)$$

$$E_n = \sum_{i=1}^{N} S_n(i)^2 \qquad (4)$$

Where N is the number of sample point of frames, and $S_n(i)$ is the value of i-th sample point.

We then use these two equations to calculate the accurate value and set the thresholds accordingly. Then, we set three states: silent, voiced and uncertain. We sequentially determinate each frame and get the voiced frames. Finally we connect these frames to get the voiced signal.

C. Feature Extraction

As long series of signals alone do not yet allow us to judge and classify cries, the next step includes identifying specific characteristics that can be used to train our classifier on. Various methods have been introduced to extract features [11]. In our algorithm, we extract four time domain features and six frequency domain features.

Consider the following definitions:

- $S_n(i)$: the signal of the n-th frame in the time domain.
- $S'_n(i)$: the signal of the n-th frame in the frequency domain after framing, Hamming windowing, and FFT.
- *T*: the features extracted in time domain.
- *F*: the features extracted in the frequency domain.
- *N*: the number of sampling in a frame.

[11] supplies the detail of each feature. Due to space limitations, the calculation equations of each feature are summarized as follows:

1) *Magnitude*

$$TM_n = \sum_{i=1}^{N} |S_n(i)| \tag{5}$$

2) *Average*

$$TA_n = \frac{1}{N}\sum_{i=1}^{N} S_n(i) \tag{6}$$

3) *Root mean square*

$$\mathrm{TRMS}_n = \sqrt{\frac{\sum_{i=1}^{N} S_n(i)^2}{N}} \tag{7}$$

4) *Spectral centroid*

$$FC_n = \frac{\sum_{i=1}^{N}(|S'_n(i)|^2 \times i)}{\sum_{i=1}^{N}(|S'_n(i)|^2)} \tag{8}$$

5) *Spectral bandwidth*

$$FB_n = \sqrt{\frac{\sum_{i=1}^{N}(|S'_n(i)|^2 \times (i - FC_n)^2)}{\sum_{i=1}^{N}(|S'_n(i)|^2)}} \tag{9}$$

6) *Spectral roll-off*

$$\sum_{i=1}^{FR} |S'_n(i)|^2 = 0.85 \times \sum_{i=1}^{N} S'_n(i) \tag{10}$$

7) *Valley*

$$FValley_{n,k} = \log\left\{\frac{1}{\alpha N}\sum_{i=1}^{\alpha N} S'_{n,k}(N-i)\right\} \tag{11}$$

8) *Peak*

$$FValley_{n,k} = \log\left\{\frac{1}{\alpha N}\sum_{i=1}^{\alpha N} S'_{n,k}(i)\right\} \tag{12}$$

Where k is the number of sub-band and α is a constant. We set k and α to 7 and 0.2, respectively.

9) MFCC

MFCC is an abbreviation for Mel-Frequency Cepstral Coefficients. In the first step, we get S'_n by framing, windowing and FFT. The next step is to filter S'_n by the Mel-filter bank. Then, we use Discreate Cosine Transform (DCT) and extract dynamic difference parameters. We extract MFCC1-MFCC12 [10].

D. Feature Selection

Sequential forward floating search (SFFS) algorithms are often used to select the optimal set of feature vectors. Starting with an empty set, a subset x from the unselected features each round is selected. Then, the evaluation function is optimized after joining the subset x, and then the subset z is selected from the selected features, so that after eliminating the subset z, the evaluation function is optimal [11]. We use support vector machines for classification and k-fold cross-validation to calculate the classification accuracy. Finally, we use the SFFS algorithm to obtain feature sets. The detail of SFFS is shown as follows.

```
Input: $F$ is the set of all unselected features;
result := {∅};
E() is the evaluation function;
done := false;
while(!done) do
    /* select feature */
    for each feature $x$ in $F$ do
        /*select the best feature $x^+$ */
        if $E(result \cup x^+) = \mathrm{argmax}[E(result \cup x)]$
        then
            $result := result \cup x^+$;
            $F := F - x^+$;
        end if
    /* stop when no features can be selected */
    if no feature is selected
    then done := true;
    end if
    /* remove feature */
    if (done = false)
    then
        repeat
            for each feature $z$ in result do
            /* select the worst feature $z^-$ */
            if $E(result - z^-) = \mathrm{argmax}[E(result - z)]$
            then
                if $E(result - z^-) > E(result)$
                then
                    $result := result - z^-$;
                    $F := F + z^-$;
                end if
            end if
        until no feature is removed
    end if
end while
Output: result;
```

E. Crying Classification

Now that we have the highly abstract feature vector of crying signal, we use this feature vector to train the SVM classifier. SVM is a supervised learning model in the field of machine learning [13]. Its principle can be described from linear separability, then extended to linearly inseparable cases, and even extended to non-linear functions, with a deep theoretical background.

We use Python's existing SVC (Support Vector Classifier) for training. The training set labels used are hunger, sleepiness, pain, and non-crying. SVC uses the "one-versus-one" method to achieve multiple classifications. This algorithm adopts the voting method. Once the model is trained, we use this model to predict crying states.

We give the pseudocode as follows.

```
Input: $f$ is the feature vector extracted;
$C[]$ are the candidate classes;
$S()$ is the trained SVM classifier;
$V[]$ are the number of votes of the classes;
repeat
    /* use trained SVM to classify */
    $class := S(C[i],C[j])$;
    $V[class] := V[class] + 1$;
until all possible tuples composed of two candidate classes have been put into $S()$
/* return the class who gets the most votes */
Output: $C[max(V)]$;
```

IV. System Implementation

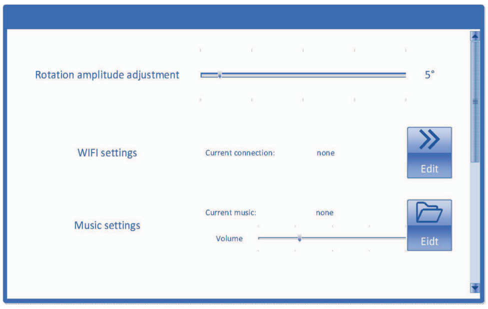
While we already introduced the system architecture in Section III, we now provide further details on the components used in our system.

- **Sensor:** The sensors we use include pressure sensors, temperature sensors, sound sensors and humidity sensors. The data collected by the pressure sensor does not need to be processed, and the data obtained by other sensors all need to be processed to determine the next action taken by the system. These sensors are all native sensors of the Raspberry Pi, which means that they are all compatible with the Raspberry Pi.

- **Raspberry Pi**: We use the Raspberry Pi as user interface to control the hardware. This interface allows the user to modify some settings, such as the frequency and amplitude of crib sway, music played, etc. The graphical user interface is displayed on an external LCD display connected to the Raspberry Pi. After connecting with the pin of the Raspberry Pi expansion board, the user can set up the device. The GUI is shown in Fig. 5. The data processing algorithms, including speech pre-processing, SVM algorithms, etc, are implemented in Python language.

- **Server**: The servlets on the server are implemented in Java programming language. There are several servlets to response particular requests. All servlets are running on Tomcat which is equipped on an ALiYun server. Tomcat is a container of servlets that can store all the servlets and enables them to run on the server. The data sending to or receiving from servlets are in JSON format, which is a uniform communication format.

- **Mobile Terminal:** So far, we have only developed an App for Android phones as Android is the most commonly used operational system for smartphones. Java programming language using the Android Software Development Kit (SDK) has been used for the development and implementation of the App. The main functions on this App is showing baby's data in graphs and charts and uploading baby's videos as well as photos. The app also allows parents to set the hardware, like playing particular music, setting buzzer's volume, etc. With the help of this App, parents can see the growth of their baby directly and know the next move of their baby so that they can make some changes to meet the baby's requirement.



Rotation amplitude adjustment 5°

WIFI settings Current connection: none

Music settings Current music: none

Volume

Figure 5. User Interface of the Raspberry Pi

V. Experimental Results

The experiment were performed on a computer with an Intel(R) Core(TM) i7-7700HQ processor and 8GB of main memory, running Microsoft Windows 10 Professional.

All crying data we used for the experiment has been extracted from videos of crying babies that have been shared on the YouTube platform. Our dataset includes five male and six female babies of different ethnicity, i.e., three Asian babies, five Caucasian babies, and three Black babies. Their sentiment is labeled according to the title of the video and assessed by a professional nurse. The non-crying data also comes from the Internet, including silence, noise, laughter, chicken roar, barking, meows, footsteps, etc.

In the remainder of this section we show the results of the algorithm described in Section III, including the result of the crying preprocess and the result of classification.

A. Results of Crying Preprocess

Fig. 6. shows the results of each step. First, we extract the raw data. Then we get the data after framing. Next, we perform the endpoint detection task. The red line indicates the beginning of the voiced signals and the green line indicates the end of the voice. Afterwards, we get the data based on the vocal fragment after cry unit detection. Each data between the red and green line is a baby splicing. crying fragment. And finally, we get the baby crying signal after cutting and splicing.

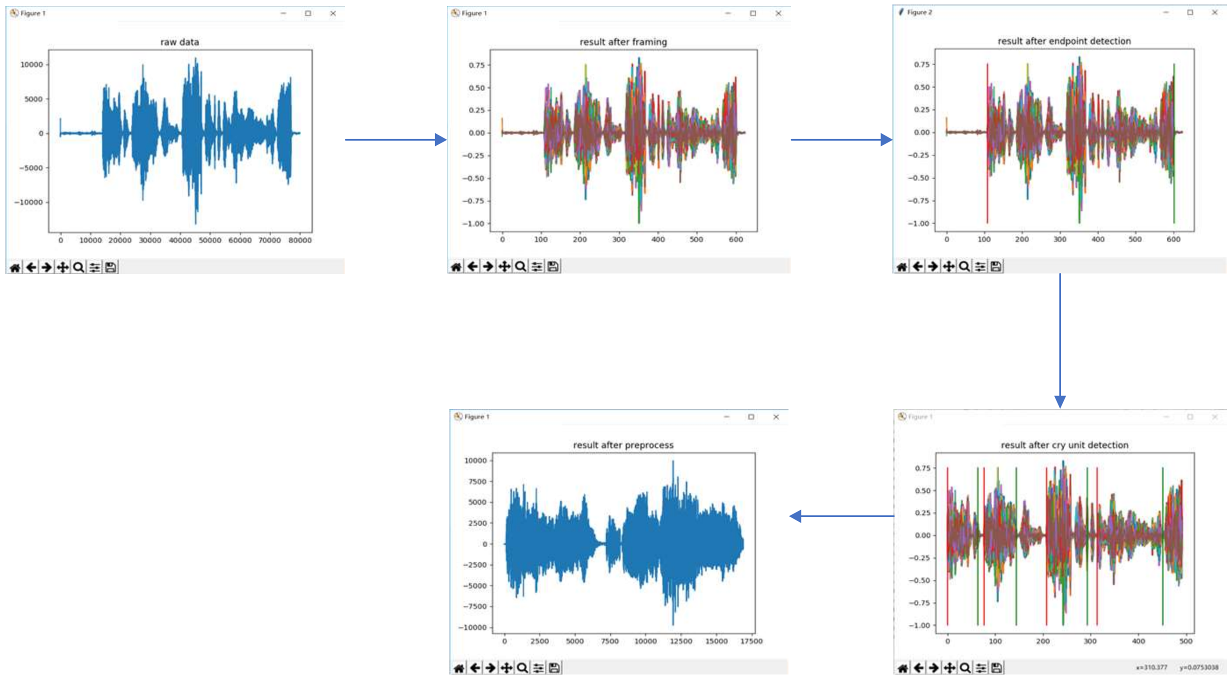


Figure 6. Results after Pre-Processing steps

B. Results of Classification

We separated crying data into three types according to the reason of crying, including hungry, pain, and sleepy. The number of the preprocessed data is shown in TABLE I.

TABLE I. NUMBER OF DATA SAMPLE

Kind of Data Sample	Training Data Number	Testing Data Number	Total
Hungry	54	54	108
Pain	47	46	93
Sleepy	47	48	95
Non-Crying	150	150	300

The results of the classifications, which include judging whether it is crying, sentiment analysis (judging of the baby is hungry, in pain or sleepy) and a comprehensive analysis in which we divide the data into hungry, in pain, sleepy and non-crying is shown in TABLE II.

TABLE II. RESULTS OF CLASSIFICATIONS

Classification	Accuracy	Prediction	Recall	F1
Crying/Non-Crying	97.33%	97.44%	97.37%	97.33%
Hungry/Pain/Sleepy	81.08%	82.27%	80.77%	80.95%
Hungry/Pain/Sleepy/ Non-Crying	90.67%	86.70%	85.58%	85.71%

We can see that when judging whether the signal depicts a crying baby, the SVM achieves a great result. The classification of different reasons of why the baby is crying are also high. In the comprehensive analysis, 90.97% accuracy is achieved.

VI. CONCLUSION

In this paper we introduced the system architecture, workflow and system implementation of a smart crib control system. The system explores the idea of approaching crying analysis as a sentiment analysis task. We use framing, endpoint detection and cry unit detection to extract data signals. Then, we extract feature vectors and use SFFS for feature selection.

Finally, we put the final feature vector into the SVM that implements o-v-o strategy to classify and predict.

At present, we have implemented a laboratory demo of the smart crib and proposed a design of the smart crib system. As future work, we will perform additional experiments using larger datasets, which would also allow the application of more sophisticated methodsq.

ACKNOWLEDGMENT

This research was supported by the National Natural Science Foundation of China (Grant No. 61202085), and the fundamental research funds for the central university (Grant No. N161704003).

REFERENCES

[1] T. G. Dietterich and E. J. Horvitz. “Rise of concerns about AI: Reflections and directions,” *Communications of the ACM*, vol. 58, pp. 38-40, 2015.

[2] B. L. R. Stojkoska and K. V. Trivodaliev. ”A review of Internet of Things for smart home: Challenges and solutions”, *Journal of Cleaner Production*, 140(3):1454-1464, 2017.

[3] A. D. Floarea and V. Sgarciu. "Smart refrigerator: A next generation refrigerator connected to the IoT," In 2016 8th International Conference on Electronics, Computers and Artificial Intelligence, Ploiesti, pp. 19-24, 2016.

[4] “SNOO-Happiest baby,” Happiest Baby, 2018. [Online]. Available: https://www.happiestbaby.com/pages/snoo. [Accessed 12 4 2018].

[5] S. Sharma and M. Tomar. *Principles Of Growth And Development*, p. 128, Isha Books, 2005.

[6] A. P. Shahnawaz, "Sentiment analysis: approaches and open issues". In 2017 International Conference on Computing, Communication and Automation (ICCCA), pp.154-158, 2017.

[7] C. Severance. "Eben Upton: Raspberry Pi," Computer, 46(10):14-16, 2013.

[8] "MySQL," [Online]. Available: https://www.mysql.com/. [Accessed 12 04 2018].

[9] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin. “Smart Homes and their users: A systematic analysis and key challenges”. *Personal and Ubiquitous Computing*, 19(2):463-476, 2015.

[10] Y. Abdulaziz, S. M. Syed Ahmad. “Infant cry recognition system: A comparison of system performance based on Mel frequency and linear prediction cepstral coefficients," In Proceedings of the 2010 international conference on information retrieval and knowledge management, pp. 260-263, 2010.

[11] A. Poornima. “Basic Characteristics of Speech Signal Analysis” *International Journal of Innovative Research & Development, 5(4):*169-173, 2016.

[12] P. Pudil, F. J. Ferri, J. Novovicova, and J. Kittler, J. “Floating search methods for feature selection with nonmonotonic criterion functions”. Pattern Recognition, 2, 279–283, 1994.

[13] W. J. Chen, H. Guo, R.A. Renaut, and K. Chen. "A new SVM model for classifying genetic data," In 2010 International Conference on Bioinformatics, Computational Biology, Genomics and Chemoinformatics (BCBGC-10), pp.54-60, 2010.

[14] K. Gaunt, J. Nacsa, and M. Penz. “Baby lucent: Pitfalls of applying quantified self to baby products”. In CHI EA’14, pp. 263-268, 2014.