WEARABLE SENSING ANALYSIS – IDENTIFYING ALCOHOL DRINKING FROM DAILY PHYSIOLOGICAL DATA

A Thesis

Presented to

The Faculty of the Graduate School

At the University of Missouri-Columbia

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Chen Zhang

Dr. Yi Shang, Thesis Supervisor

July 2016

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

WEARABLE SENSING ANALYSIS – IDENTIFYING ALCOHOL DRINKING FROM DAILY PHYSIOLOGICAL DATA

Presented by Chen Zhang

A candidate for the degree of

Master of Science

And hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Yi Shang

Dr. Timothy Trull

Dr. Yunxin Zhao

# **ACKNOWLEDGEMENTS**

I would like to take this opportunity to thank all the people that had made efforts and helped with this thesis work.

First of all, I would like to express my sincere appreciation to my academic advisor, Dr. Yi Shang, for his continuous support and consideration. It is him who leads me to do great research and always gives valuable advices and motives me with encouragement, depending on which I could produce solid works. It is really my great honor to learn with him along the whole graduate career.

I would also like to thank Dr. Tim Trull. We work together for almost the same length of time as that with my academic advisor. The helpful insights and professional knowledge he brought play the key role to my research studies. Without him and his help I couldn’t finish this thesis work.

Finally I would like to thank Dr. Yunxin Zhao for reviewing my thesis and providing valuable comments and suggestions. There are many parts of my thesis work are inspired by courses I learnt with her during my graduate studies. Thank her so much for taking time to be my commit member.

And thank all my colleagues working in the lab. They have always been helpful and I really enjoyed the time spending with them along the way.

**TABLE OF CONTENTS**

[**ACKNOWLEDGEMENTS** ii](#_Toc456267892)

[**LIST OF FIGURES** v](#_Toc456267893)

[**LIST OF TABLES** vi](#_Toc456267894)

[**ABSTRACT** vii](#_Toc456267895)

[**1.** **INTRODUCTION** 1](#_Toc456267896)

[1.1 Problems 3](#_Toc456267897)

[1.2 Contributions 5](#_Toc456267898)

[1.3 Thesis Organization 7](#_Toc456267899)

[**2.** **BACKGROUND AND RELATED WORK** 8](#_Toc456267900)

[2.1 Reviews of mAAS 8](#_Toc456267901)

[2.1.1 System Components 9](#_Toc456267902)

[2.1.2 In-Residence Study Protocols 13](#_Toc456267903)

[2.1.3 A 7-Layer Model for System Overview 15](#_Toc456267904)

[2.2 Related work 16](#_Toc456267905)

[**3.** **SYSTEM IMPROVEMENTS** 17](#_Toc456267906)

[3.1 Random Survey Scheduling 17](#_Toc456267907)

[3.2 Object Oriented Programming 18](#_Toc456267908)

[3.3 Digital Envelope Implementation 20](#_Toc456267909)

[**4.** **A NEW DATA ANALYSIS PIPELINE** 26](#_Toc456267910)

[4.1 Pipeline Overview 26](#_Toc456267911)

[4.1.1 Statistics on Raw Data 26](#_Toc456267912)

[4.1.2 Pipeline 29](#_Toc456267913)

[4.2 Data Cleaning 30](#_Toc456267914)

[4.2.1 Step-by-Step Procedures 30](#_Toc456267915)

[4.2.2 Results for Cleaned Data 36](#_Toc456267916)

[4.3 Feature Extraction 37](#_Toc456267917)

[4.3.1 Feature Definition 37](#_Toc456267918)

[4.3.2 Visualization 41](#_Toc456267919)

[4.3.3 Feature Analysis 43](#_Toc456267920)

[4.4 Classification 48](#_Toc456267921)

[4.4.1 Data Labeling 48](#_Toc456267922)

[4.4.2 Classifiers 50](#_Toc456267923)

[4.4.3 Performance Measurements 50](#_Toc456267924)

[**5.** **EXPERIMENTAL RESULTS** 52](#_Toc456267925)

[5.1 Experimental Design 52](#_Toc456267926)

[5.2 Classification Results 53](#_Toc456267927)

[5.2.1 ECG Features Only 54](#_Toc456267928)

[5.2.2 ECG and RIP Features 56](#_Toc456267929)

[**6.** **CONCLUSIONS** 60](#_Toc456267930)

[**7.** **FUTURE WORK** 61](#_Toc456267931)

[**8.** **REFERENCES** 62](#_Toc456267932)

# **LIST OF FIGURES**

[Figure 1. Examples of Android Application Pages 9](#_Toc456267872)

[Figure 2. Sensor Suites: SEM (left) and Hexoskin (right) 11](#_Toc456267873)

[Figure 3. A 7-Layer Model for Whole System 15](#_Toc456267874)

[Figure 4. Illustration of Random Survey Scheduler 17](#_Toc456267875)

[Figure 5. UML of OOP Design 19](#_Toc456267876)

[Figure 6. Principle of Symmetric and Asymmetric Cryptography 23](#_Toc456267877)

[Figure 7. Security Implementation - Digital Envelope 23](#_Toc456267878)

[Figure 8. Weeks of Study for Each Participant 26](#_Toc456267879)

[Figure 9. Valid RR Interval Days for Each Sensor Suite 27](#_Toc456267880)

[Figure 10. Pipeline of Data Analysis 29](#_Toc456267881)

[Figure 11. Illustration of Outliers Removal 32](#_Toc456267882)

[Figure 12. Illustration of Discriminating Activity-Affected Signals 33](#_Toc456267883)

[Figure 13. Illustration of Gaps and Insufficient Data 34](#_Toc456267884)

[Figure 14. Valid-Minute Rate Screening 35](#_Toc456267885)

[Figure 15. Illustration of Other ECG features (left) and RSA (right) 41](#_Toc456267886)

[Figure 16. Distributions of All 23 Features, Hexoskin Suite 42](#_Toc456267887)

[Figure 17. HRV Features for One User Whole Day 43](#_Toc456267888)

[Figure 18. Histogram Represented as Heat-map for Heart Rate, SEM Suite 44](#_Toc456267889)

[Figure 19. Pairwise Correlations of All 23 Features, Hexoskin Suite 45](#_Toc456267890)

[Figure 20. Illustration of Drinking Timestamp Expansion, to One Hour 49](#_Toc456267891)

# **LIST OF TABLES**

[Table 1. Statistics of RR Interval Data for the Two Sensor Suites 36](#_Toc456267933)

[Table 2. HRV Features 38](#_Toc456267934)

[Table 3. Respiration Features 40](#_Toc456267935)

[Table 4. Ranks of Features in Each Category, Hexoskin Suite 46](#_Toc456267936)

[Table 5. Principal Component of 25 Raw Features 47](#_Toc456267937)

[Table 6. Quick Statistics of Data to Be Classified 53](#_Toc456267938)

[Table 7. Classification Results on ECG Features, SEM Suite 54](#_Toc456267939)

[Table 8. Results Comparison on ECG Features, SEM Suite 55](#_Toc456267940)

[Table 9. Classification Results on ECG Features, Hexoskin Suite 55](#_Toc456267941)

[Table 10. Classification Results on ECG and RIP Features, Hexoskin Suite 57](#_Toc456267942)

[Table 11. Classification Results for Case Study, Hexoskin Suite 58](#_Toc456267943)

# **ABSTRACT**

In practical psychology research, questionnaires and interviews with examinees are commonly used. Besides, to study about certain human behaviors such as our subject – alcohol drinking craving, it usually lasts for several days which could be costing and time consuming, to both researchers and examinees. Previous work in our lab provided a smartphone-client and server based distributed system which does not need participants to go to the clinic doing any paperwork nor any invasive blood or medical-electronic based tests, so that saving time and money for better researching purposes. The system called mAAS and started to collect precious data from our recruits. Improvements that I made to the system are presented in this thesis. Also, a new data analysis pipeline is built and evaluated. Modern wearable technology is integrated without extra burdensome equipment in our system. Simple, they would be just smartphone and a vest suite with sensor in it, those are already our daily-carry personal stuffs. Smartphone application will be in charge of communicating with sensor suite and server wirelessly and safely. We then can get large numbers of detailed physiological information about participants in their real life. Data preprocessing is applied to get cleaned data. Features about ECG and RIP are extracted afterwards. As a result, involving machine learning methods, we could apply classification on these features and give out pretty good performance on understanding and analyzing tasks. In this thesis work, alcohol drinking activities are identified from daily body physiological data through the pipeline as presented.

# **INTRODUCTION**

There are more and more interdisciplinary sciences in recent years. Among them, computer science or data science is likely to be the most widely being integrated that plays a significant role working together with specialized knowledge in each of other disciplines. With the help of developing technologies, data around us has been grown dramatically faster nowadays. Tons of data is produced and stored in every field. Information highway makes it fast collected and easily accessible by us. Facebook users like over 4 million posts in a minute. YouTube users upload over 300 hours of new videos to the platform every minute [1]. Data science is a blossoming industry that becomes the engine of the next information revolution. We could live in a Data Age in a very short time. Every discipline moves on to big data epoch and changes accordingly. During my career as a graduate student, our lab had a close cooperation with psychology sciences department, which gave me a great opportunity to conduct an inter-science research with a data driven fashion.

Psychology is the scientific study of the mind and behavior. Clinical psychology is a broad branch of psychology that focuses on diagnosing and treating mental, emotional, and behavioral disorders. Our psychology research studies about human emotional dysregulations and behaviors about alcohol drinking craving. The study would enroll several examinees for about two weeks long and gather three types of information about their daily lives: questionnaires, physiological datasets and geographical locations. The questionnaires and location diary will together tell what and where was the participant doing. Later, data alignments with physiological measurements, such as heart rate, breathing rate, skin temperature, etc., will be conducted, helping to interpret their physiological signals changing. In old time, these were done by asking the examinees to come to a clinic every day, trying to recall their memories about the day and filling up with forms and questionnaires, then putting on some medical-electronic based equipment and getting their physiological measurement with just a very short duration time of a day. The procedure was costing and time consuming for both the researchers and examinees. And the research usually cannot get both accurate and enough data even though the study lasted longer.

Things had changed with the development of information technology. Many clinical psychological research is using mobile ambulatory assessment. Unlike it in the traditional way, a mobile ambulatory assessment gathers information about the examinees through the use of electronic diaries to collect self-reported surveys, behaviors, or physiological data in real time. For the first type of information, surveys, we now create an application running on the smartphone carried by the examinee, containing the user triggered reports and system launched questionnaires popup at randomly scheduled time in a day. Both surveys may ask about their daily activities, current circumstances, what’s the emotional feelings and other similar questions in common. Particular survey responses about specific activities, such as drinking, smoking, mood changing and medicine in taking, would be caught up and then followed -up surveys would be scheduled in a next dense period of time, focusing on the subjects of different studies, recording and later analyzing the behavioral and emotional characters and disorders. Furthermore, the smartphone now provides GPS tracking records which will show the daily movement routine clearly and precisely, without relying on the user’s memory, which might be blurred and mistaken. Both of these two signals are marked by timestamps and make it easy to be synchronized together with physiological data. Physiological data is also retrieved with the help of new wearable technology. In order to influence as few as possible on participant’s normal life and provide accurate data recording, we use remote sensing instead of traditional invasive medical procedure such as blood draw or breath-alcohol test. Recent technology allows many physiological sensors equipped with Bluetooth, and we could use that to communicate with our smartphone application. In this way, we have ability to integrate different types of sensors.

* 1. Problems

As is shown in references [16 and 18], we had a mobile ambulatory assessment system fundamentally worked for our study. It is a distributed system in which each client can be deployed to different users, and all the users’ data can be transmitted and collected through a server end. The smartphones hold the abilities of cellular connection. Surveys and daily digital diaries were provided and presented through the smartphone screen. These user interfaces were designed and implemented. Integration with different types of sensor suites is another important module of the system, interfaces were implemented as well. In the end, the smartphone and the server integrated all these modules and needed to provide stable and robust services.

The system called Mobile Ambulatory Assessment System (mAAS) [16] and is a data collecting system links wearable sensors with smartphones and servers. More general system overview would be given in chapter talking about background. The system can dispatch questionnaires, collect survey answers and all kinds of physiological sensor measurements. As the data always contain personal information and private responses, to keep the data secure, it must be encrypted during the transmission through public network. An efficient encryption way is needed thereafter. The system focuses on data collection and seeks for a way to collect as much as involved data in an efficient and affordable way. However, it didn’t work perfectly in every aspect, there are three main issues that need to be addressed during the time I helped with implementing it.

The first problem is extendable issue. We didn’t want the application to be a one-time implementation. Our application should be ready for other similar research studies with a prompt system migrating procedure. Usually the studies are all similar except for the survey categories, questions being asked and types of answers. There needs a way for our application to be extendable and changeable across different research objects.

The second is security issue. As we stated that many personal information and measurements of real participants are collected and transmitted through public network, there needs a way to guarantee the data is secure during the transmission.

The above two are about implementing of the application. The third issue is, only preliminary analysis had been applied since we had collected a large numbers of precious data. Although there are many psychology research studies worked on humans and human behaviors, little of them related to investigating the relationship between alcohol drinking and its influences on physiology reactions. For example, we all know from our life experiences that drinking would latently affect our heart rate and breathing rate, but lack of quantitative analysis or demonstration on that.

The research aims at better understanding about human alcohol drinking activities and its relationship with inner physiological reactions. Drinking habits are also with our concern, as we know that drinking may affect people’s mood and vice versa. We do this in order to help people who have alcohol abuse problems and later may offer in-time intervention through our system.

* 1. Contributions

The system was implemented collaboratively by several persons. And I have made contributions basically from the two phases:

* Phase one: during implementing, made improvements on the existing system

1. A truly randomized survey scheduler

Implemented the random survey scheduling protocol for the system prompted surveys, which is an initial requirement by system design.

1. A redesign with OOP for survey component

Applied a better Object-Oriented Programming (OOP) design on the structure of survey provider and its control logic, which made it easy to configure or reapply to other similar studies, facilitating the procedure of system migrating. At least 3/4 of the code are saved, which made it more readable and maintainable.

1. A novel approach for a secure data-transfer channel

Implemented an efficient and secure way of encrypting the data transmitted through smartphone client to our PHP server. The method has a vivid name called Digital Envelop, which combined the AES and RSA and will be discussed detailed in later chapter.

* Phase two: after data collected, built up a new data analysis pipeline

1. Data preprocessing protocol

An automated run-through program can process our available physiological data from various sources. Sequences and rules are first time tried and defined in our research study, and verified with real subjects.

1. Feature extraction on physiological data

Referred to related works and helpful professional advices, 25 of physiological features in total are defined and calculated from cleaned data from last procedure. Some of them are heart rate variability related while others are respiration related features.

1. Classification models on drinking identifying

Involved machine learning methods, classifiers of drinking episode from daily physiological data is built and tested on the real data with nearly 80% accuracies, showing the applications of this study.

* 1. Thesis Organization

The rest of this thesis is organized as follows:

Chapter 2 is about the background of our existing data collecting system, called mAAS. System is reviewed in general, main components are briefly introduced. And some related work from other researches are also discussed. Most of them have a similar design of the experiments but study about different human behaviors, such as Stress and Cocaine usage.

Chapter 3 describes implementation of three specific modules that I had improved for mAAS.

Chapter 4 presents a new data analysis pipeline, which contains data cleaning, feature extraction and classification modeling.

Chapter 5 shows the experimental and analytic results.

Lastly, Chapter 6 and 7 summarize conclusions about our discoveries and discuss a little about future work.

# **BACKGROUND AND RELATED WORK**

* 1. Reviews of mAAS

The mAAS [16] in this thesis refers to the mobile ambulatory assessment system. It is an Android smartphone-server distributed system. The main job is to build a stable and secure data collection system that fulfills the requirement of our research study. The research intends to build a system that integrates remote sensing technology with the flexible programming abilities of Android smartphones. By the time we built the system, Android OS shared 84.8% of the market [2]. Another reason we choose Android devices was, the OS is open source and we had more permissions to acquire specific system information, such as battery usage, system time adjustment and GPS confidence, which are used as important parts in our system.

We had mainly tried two types of wearable body sensor suites. The first one called Equivital EQ02, we will call it SEM suite in the following. The SEM sensor suites are approved by FDA and is guaranteed to provide multi-parameter, ambulatory monitoring abilities of accurate, mobile human condition and performance data. The other type called Hexoskin Wearable Body Metrics, we will call it Hexoskin for short in the following. Our system can work with these two types of sensors, both of them are vest-like wearable suites that can measure signals as detailed as Electrocardiograph (ECG) and Respiratory inductance plethysmography (RIP). Besides, we tried other devices like wrist band or watch which can provide heart-rate, Galvanic Skin Response (GSR) and temperature measures. All these new wearable technologies provide Bluetooth connectivity and can easily be added or switched during our data collection phase in the whole study. The results will be compared in later chapters.

And this is not the only advantage of using our smartphone based sensing system, with the all-day long lasting support of the smart sensing, we can get continuous large amount of data covering the activities of the examinee for the whole day. Then we can choose what aspects of study purposes or analysis can be performed on our desires. We had tried many aspects so far, such as drinking craving, emotional dysregulation, sleeping quality and certain medicine intake effects.

* + 1. System Components

1. Android Application

Figure 1 shows what participants would use during their participating period, application pages with the following names have different functionalities.

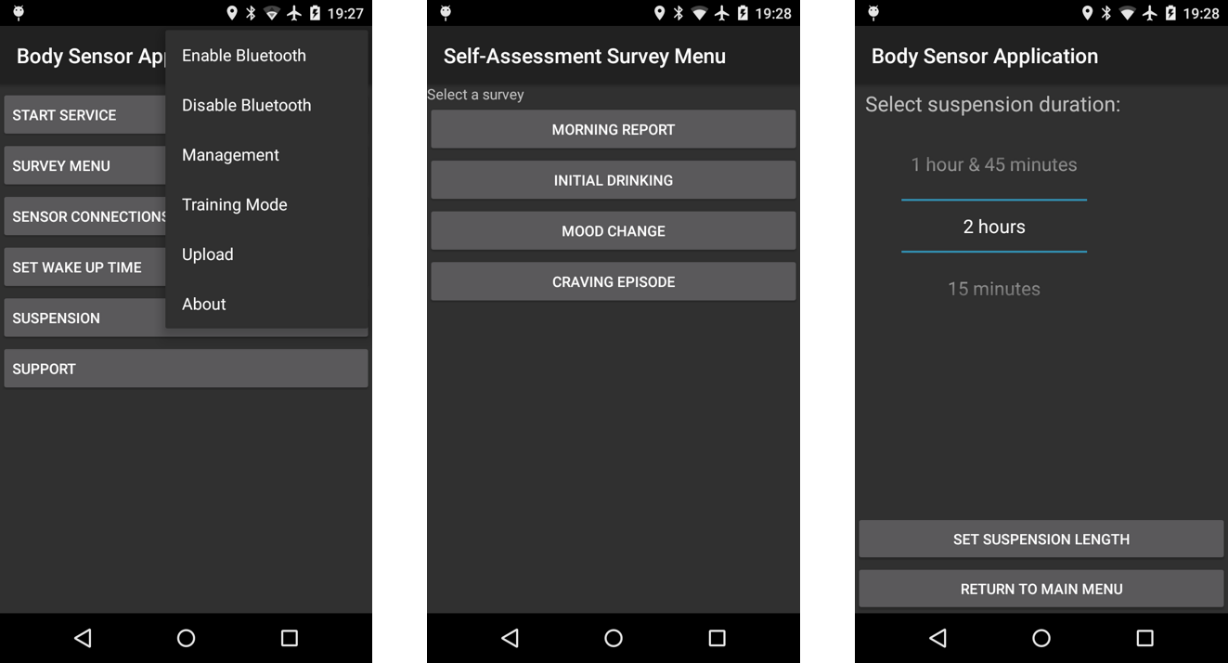


Figure 1. Examples of Android Application Pages

* Management: used by administrator for the first time only. Admin would assign a unique ID to each of the examinees and teach them how to use the application with training mode.
* Training Mode: one special mode with none of the data would be recorded. Opened and closed by admin just letting the user know how to use the application and what each of the surveys looks like.
* Upload: all the data would be submitted to our server automatically, however, under certain situations the data may lost during transmission. We also offer this uploading function which can by manually triggered after the user finished with study.
* Survey Menu: a menu that contains all types of the surveys that may be asked during the study. In this alcohol craving study, there are six types of surveys. Two of them are only scheduled and auto-triggered by the system, which are Random Survey and Drink Follow-ups. The other four can be selected by users at the time they thought certain conditions are meet, those are Morning Report, Initial Drinking, Mood Change and Craving Episode, shown in the middle picture.
* Sensor Control: to connect and manage what types of sensor suite is going to be used, status can be monitored from here as well.
* Morning Alarm: in order to remind user to wear sensor suite and carry research phone, we offer a morning alarm every day. User can set the time of morning alarm through here.
* Suspension: survey that scheduled at the specific time would triggered by system with a ringtone. We do understand under certain situations the user need to quiet the phone. Suspension function offers an up to 2-hour pause of the study, during which no surveys would be scheduled and popped up. User can set another suspension right after the previous one. The 2-hour limitation is restricted because of, sometimes the user forget to turn it back after they had done with the special situations.
* Support: when participants had encountered problems using the application, comments from the user together with system logs would be sent to us through email.

1. Sensor Suite

Beyond the smartphone application, we also used wearable sensors to collect data in a free-living environment. In this thesis, we tried two types of sensors, SEM and Hexoskin. Both of them are core devices in a small box connected to a flexible vest or shirt worn on the upper body to capture sensor data such as ECG and RIP. They also have 3-axis accelerometer to record motion activities. The battery life is about 8 hours. Characters of the two sensors are as following:



Figure 2. Sensor Suites: SEM (left) and Hexoskin (right)

* SEM

SEM suite uses a flexible one-shoulder band worn over the chest and shoulder to capture respiration data with RIP. It also contains a two-lead ECG sensor, body temperature sensor and 3-axis accelerometer. The sampling rates for the band are 26 Hz for RIP, 256 Hz for ECG, 26 Hz for each of the three axes of accelerometer readings, and 1 Hz for body temperature sensor. Other signals that derived from the above are: quality of HR, quality of BR, ECG derived Breathing, and R-R Intervals.

* Hexoskin

A bit unlike SEM, Hexoskin suite is a shirt liked sensor which states to provide more fit and comfortable wearing experience. It provides a thoracic sensor and an abdominal sensor together to measure RIP. ECG and accelerometer are also provided. The sampling rates for the shirt are 128 Hz for RIP, 256 Hz for ECG, 64 Hz for accelerometer, but there is no sensor for body temperature. Besides these, abundant of auxiliary signals are provided, such as cadence, steps, minute ventilation, tidal volume as well as qualities of HR, BR and RR Intervals.

1. Server

Server software supports data decryption, storage and visualization on the web pages. With the server, we can easily maintain the data collected from the system and get a quick look into that. The web pages provide a curve graph view for different types of physiological data that superimposed with each other according to timestamps. It also provides a good way to view all the GIS location information upon a Google Map view. It is written on PHP and implemented by other colleagues in our team.

* + 1. In-Residence Study Protocols

So far we had discussed about all the three components of the system. In our research study, we had recruited several independent examinees to participate our two-week long study. The two weeks can be consecutive or discontinuous. The examinees would live as their normal lives including their normal drinking activities. In-residence protocols are for a real user, what a normal day should be followed:

Before the participant start the study, we would invite her/him to the lab and have a tutorial with the training mode in our app. Then the server would generate a user ID and we assign it to the smartphone through the management tab. Each participant would also get a sensor suite, either SEM or Hexoskin. For the early stage of this thesis work, we mainly use SEM sensor suites, and for the later period we use Hexoskin instead because of the suite fit/comfortable issue and price concern. So we have data for these two types almost half and half.

During the study week, in the morning, the research phone would alarm and suppose the user would catch and finish a short Morning Report, recording about the morning mood states and situations for last night. Then the user would put on the sensor suite, connect and start to send physiological data to our server. If the user doesn’t catch the morning alarm, it is by noon that the user still could trigger one manually through survey menu in the application. When Morning Report finished or by the time of noon, the system would call an algorithm randomly picking 6 different time points for the system to pop up Random Surveys. These Random Surveys are supposed to catch the routine of their daily lives and hopefully any drinking activities forgot to be reported on the user’s own initiative. Normally, the user is trained to self-report an Initial Drinking at the time of drinking activity during a day. When drinking activity is reported, within the following 2 hours, the system would schedule triple Drinking Follow-up surveys for the user. The interval would be 30, 30 and 60 minutes making the total duration called a drinking episode. The duration may be extended according to the user’s drinking scenario. During the drinking episode, Random Survey would be skipped because of the similarity of the two types of survey.

At the rest of the day, the user would also trigger other types of surveys through survey menu. Mood Change is for the time when user feels mood had changed dramatically, and we would ask more additional questions about mood category. Craving Episode asks about the reason and current situation for a user who feels like to have alcohol drinks but not yet. The user could suspend the system for up to two hours each time when it is necessary. And at 9pm, the system would remind the user it’s time to set tomorrow’s morning alarm and remember to charge both the research phone and sensor suite. These make up an expected normal study day for each examinee.

After the study week, the user would return all the devices back to us. And we would remove the ID from research phone after we make sure that all the data are backup and synchronized properly to the server. Upload button can do the job if need. Data obtained from the in-residence field studies are used to develop models for detecting alcohol drinking activities and habits. The physiological data is carefully labeled with surveys, telling the current situations or events.

* + 1. A 7-Layer Model for System Overview

To summarize the whole system design, I try a new way from [7], which is finalized as a 7-layer model of alcohol drinking craving study:

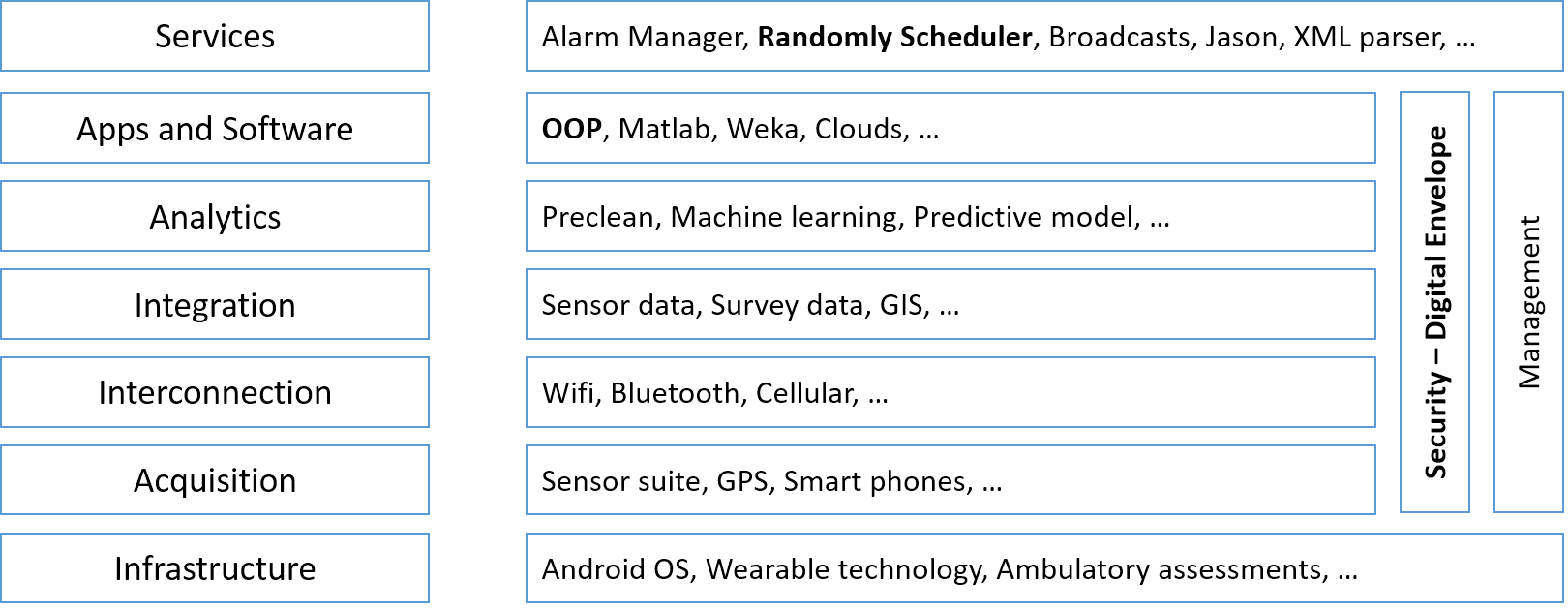


Figure 3. A 7-Layer Model for Whole System

As we can see from figure 3, the Android application consists of many different modules. And each of the modules has a specific functionality. It took us a long time to debug with these modules. Among them, I primarily made contributions to the following modules:

* Object Oriented Programming: encapsulate the survey questions and answers, making the code easily reused and better maintainable.
* Random Survey Scheduler: offer a uniform service for schedule the Random Survey in a day.
* Security: implemented a strength and secure way called digital envelope.
  1. Related work

Refer to other related studies in the field, [3] introduced a similar mobile sensing device called AutoSense, integrated with ECG, RIP and GSR Sensing. Their team built the devices by their own, which had a comparable capacity as ours. They presented an unobtrusively wearable wireless sensor suite that can collect continuous measurements for inferencing of human physiological signals.

Then [4] and [5] utilized the suite and produced models to make inferences from wearable sensors. Paper [4] is important to us because it has a similar problem with ours. It tried ways of processing data and verified on those. In work [5], several rules and standards were introduced and testified. Although both of these studied were about Stress and its assessments, I still found them very helpful to my thesis study.

Another study [6] is about physiological responses of Cocaine intake and gives us different ideas. It shows different dosage levels of stimulation may affect human body physiological readings differently. It developed a computational model for automated detection of such health related events from sensor readings such as ECG. All of these show that human physiological responses would reflect both certain Physiological and Psychological activities, providing us a variety of useful tools and confidence on making our classification and inference of human alcohol drinking craving activities. Also, short-term and long-term changing could be reviewed separately.

Other similar work is done in [14, 15], which would be talked about a little in the following sections

# **SYSTEM IMPROVEMENTS**

* 1. Random Survey Scheduling

The purpose of Random Survey is to try to catch the routine of the daily life for a user. First of all, this helps us to get the mood states, understanding what kind of situation and with whom the user may has that feelings. Also, Random Survey asks about whether or not the user has had a drinking activity since last survey. There are some probabilities that the users forgot to click and report an Initial Drinking when they actually had one. We are trying to make these up. Finally, we need to find a way to evaluate the compliance of using our application during the study week, so that we could decide how much to pay for the examinees. This gives the restriction that the Random Surveys should be spread out, however, without any overlapping with each other all over a day. The random survey scheduler will make sure the system monitors the subject’s entire day.

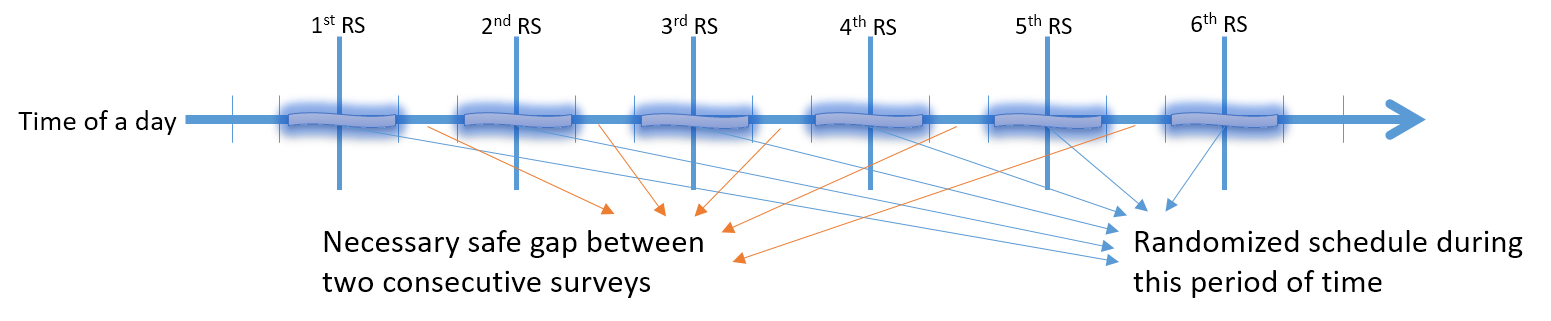


Figure 4. Illustration of Random Survey Scheduler

Figure 4 is an illustration of how we randomly schedule the Random Surveys. After the user finish the Morning Report, or it goes to the noon of day whichever comes earlier, the system will divide the rest of time, till midnight, to N+1 parts. For the craving study N is 6. So we then divide each part to three sub-portions, make there are 21 portions in total. Random number will be generated in the selected potions, indicated as the thick bars shown in the graph. This divided and randomized method will leave a necessary gap between each pair of the consecutive schedules.

I make this implementation as an API library of our Android application. Wherever or whatever it needs a service like that, we can simply call the API to get the random schedulers. Other similar project studied about medicine usage may just need three Random Surveys per day, so we can simply make it with parameter N equals 3.

* 1. Object Oriented Programming

When implementing the logic for survey categories, questions and answers, Objective Oriented Programming ideas are applied. There are numerous of advantages for implementing in this way. First of all, the structures become clear. As we can see from figure 5, there are three inherit layers of Java code. The Parents layer is abstract layer, containing Category.java, Question.java and Answer.java files. Each of the files defined certain abstract functions which are commonly used across all control flow, such as setters and getters. Then three children class inherit from their parents respectively and define the functions in detail. Those children classes are named after SurveyCategory.java, SurveyQuestion.java and SurveyAnswer.java, representing they are more specifically written. Along with the implementations of the abstract functions, basic logics of control are also defined. Survey question types are restricted to Single-choice, Multiple-choice, Number-selection and Text-input types. Then for the grandchildren layer, more specific functions are implemented to fulfill different survey requirements.

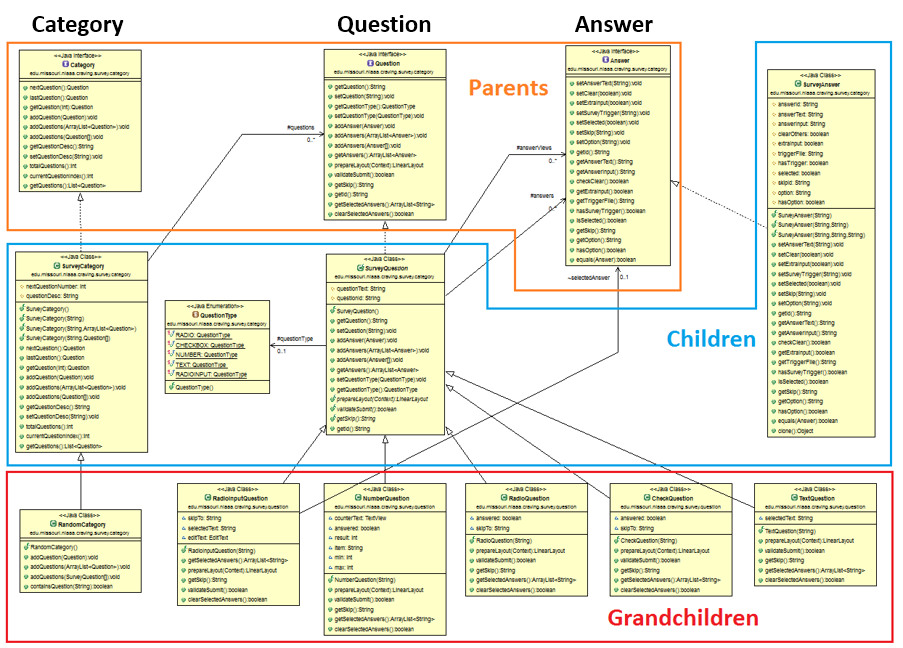


Figure 5. UML of OOP Design

The second advantage of using OOP is code reusing. As we could imagine, many of the surveys or answers are similar, either in structures or ways of presenting. I don’t need to repeatedly define and implement with similar code in this way.

The last but not least is, it makes more maintainable codes and facilitates the modifications later when we change the plan. For example, if in future we want to add a new type of question, which comes with its own set of answers. We can easily add the template into QuestionType.java and then create a new java file named after that question type. All the common functions and logically control part will be directly inherited from other predefined files.

* 1. Digital Envelope Implementation

In our Android application, collected data from all kinds of sources are transmitted around places. The transmitted data flow can be viewed as:

**Sensor** ==[Bluetooth]==> **Phone** ==[Wifi + Celullar]==> **Server**

We assume that the transmission between sensor and phone is secure guaranteed by standard Bluetooth protocols. So our primary purpose is to protect the data transmitting through the public network in second half. We will apply encryption to the data before it has been put onto the public.

1. Methods Overview

In cryptography, encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption doesn't prevent hacking but it reduces the likelihood that the hacker would be able to read the data which is encrypted [8].

Basically, all the algorithms of encryption are quite open. Instead of creating new method base on mathematics, people increase the cipher strength so that it will take a rather long time to do the cracking by known techniques. Usually the long time means much more than a couple of thousands of years. The cipher strength is represented by length of the cipher. For example, we may say an approach of AES Encryption is 128 bits. It would take a billion monkeys billions of years to break a 128-bit instance. The longer the strength of the cipher, the more secure the encryption method is, assuming that there is no new techniques been found, such as quantum computers.

Encryptions of data will happen on the phone side, so that to keep the data hidden from others in the public networks. AES encryption is a well-known and widely used cryptography method to keep the data safe from cracking. I would choose AES to be the main encryption algorithm in our implementation. However, AES belongs to the type of symmetric encryption. Symmetric method will induce a problem of how to transmit the key. We will describe this later in detail with introducing another auxiliary encryption method, who is asymmetric, RSA cryptosystem.

On the other hand, receiving data happens on the server side, data would be properly decrypted and stored in plain human readable text, put into different categories, as the sensor, the survey and the location data.

1. Algorithms

In general, the system requires encryption on the Phone and decryption on the Server. There are two types of encryption: Symmetric algorithms and Asymmetric algorithms.

* Symmetric-key algorithms

It is a class of algorithms for cryptography that use the same cryptographic keys for both encrypting of plain text and decrypting of cipher text. The keys may be identical or just a simple transformation before and after. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link.

* Asymmetric algorithms

It is also known as Public-Private key cryptography, refers to a cryptographic algorithm which requires two separate keys, one of which is secret (or private) and one of which is public. Although different, the two parts of the key pair are mathematically linked. The public key is used to encrypt plaintext or to verify a digital signature; whereas the private key is used to decrypt cipher text or to create a digital signature

Both of the two algorithms have some limitations when apply to resolve a real problem. In symmetric cryptography, the same identical key is used by both encryption and decryption. Keeping the key private is critical issue making the data confidential. To address this difficult, it introduces asymmetric cryptography, which uses a public-private key pair to encrypt data. Only private key is critical to be kept confidential in this scheme. However, asymmetric algorithms are generally much slower than symmetric ones. In practice, asymmetric algorithms are used to exchange smaller secret keys which are used to initialize symmetric algorithms.

Suggested by Oracle, the Java provider, in our implementation, we choose AES as the one of the symmetric cryptography, and RSA as the asymmetric cryptography. Let the data encrypted by AES, meanwhile the unique AES key protected by RSA key-pair cryptography, so that encrypting the much majority of the data with faster algorithm and protecting the small-size key with the slower buy safer one. The overall idea is presented below in figure 6:

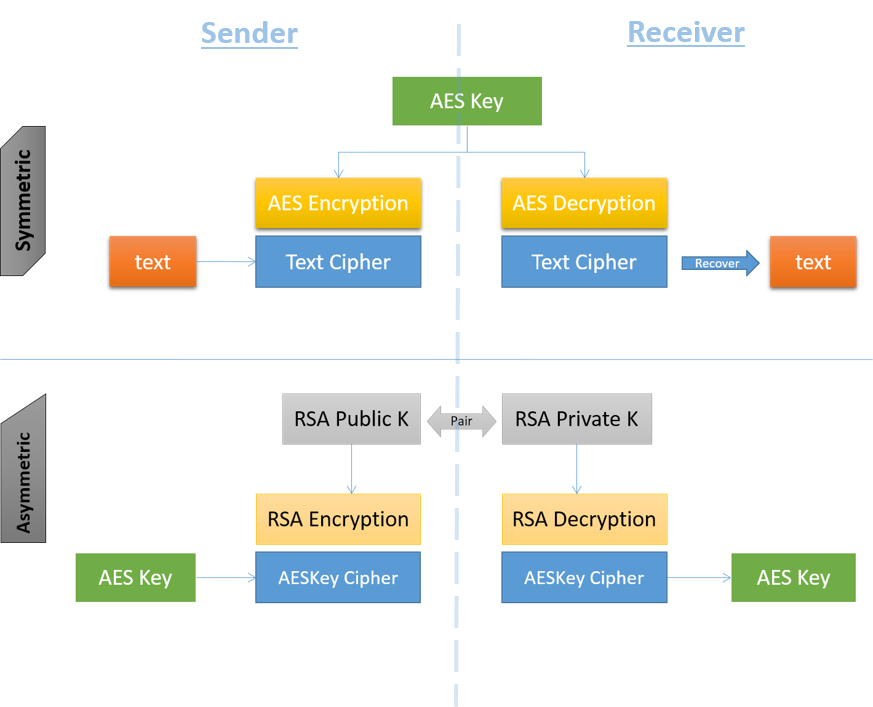


Figure 6. Principle of Symmetric and Asymmetric Cryptography

1. Implementation

From the principle architecture shown above, we can now implement the encryption and decryption in our system. Symmetric Cryptography would deal with the ciphering issue to plain text, since AES performs a higher speed on ciphering, we encrypt larger sized text files with it. For its symmetric key, which is much smaller, we then encrypt it with RSA cryptography method.

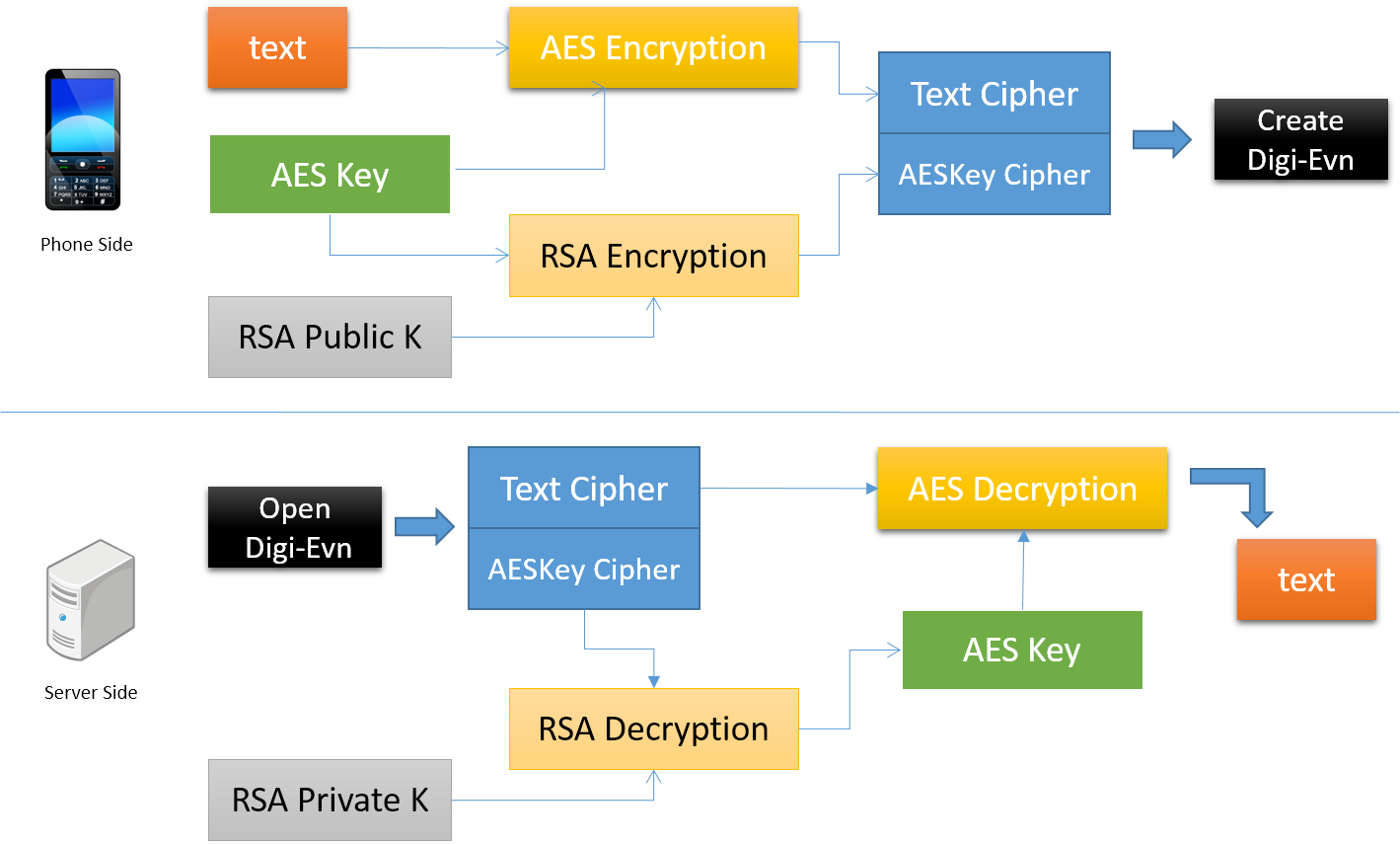


Figure 7. Security Implementation - Digital Envelope

System Architecture design is shown in figure 7. The data flow follows the following sequence:

1. Generating the RSA key pairs on Server.

Public Key assigned with the Smartphone, and kept Private Key in confidential

1. Generating the AES key on Smartphone

128-bits AES cryptography key

1. Encrypting files on the Smartphone

Encrypt file with AES and the key itself with RSA respectively, merge the two ciphers together

1. Transmitting to Server, getting decrypted files

Use RSA private key to decrypt AES key and later on get decrypting files

After debugging and adjustments to our application, the system works well and protects all the collected sensor data and survey results for participants. The system guarantees the following:

* All the files stored on the Smartphone side, which would be transmitted to Server wirelessly, are encrypted and cannot easily be cracked as long as 128-bits AES is still announced secure.
* AES key will be stored and transmitted in encrypted format, which is encrypted by 1024-bits RSA algorithm.
* Encrypted file ciphers and encrypted key cipher would be transmitted to the Server via public network together at one time, they won’t be messed up.
* Private Key to RSA will be generated and stored only on the Server and protected by administration of the Linux system, which is also strong security warranted.
* Files stored on Server will be plant human readable text files. We assume that the Server is always secure on the basis of Linux system.

# **A NEW DATA ANALYSIS PIPELINE**

* 1. Pipeline Overview
     1. Statistics on Raw Data

As it is described in the protocol of our study, we kept recruiting participants for our alcohol craving research. We only had a few sets of research smartphone and sensor suite, so we would only have a few users every week. We named the user ID with 4-digit numbers starting with 1001. In our dataset, the first user tried on our study could be as early as October 2014. Most of the examinees were college students, our recruiting would stop during the holiday breaks or vocations between semesters. By the time of March 31 2016, we had collected a large amount of raw data. Figure 8 shows the weeks of examination for each user with two different sensor suites.

Figure 8. Weeks of Study for Each Participant

During our study, we had enrolled more than thirty users, however, some of them quit halfway, the figure above is just showing users who finished and how many weeks they had participated in rough. We already know there are two sensor suites, bar shape in orange color shows the ones for SEM suite, while the blue bars represents Hexoskin. By April 2016 we have 10 users for SEM and 14 for Hexoskin. We are also noticed by the graph that some of the user might try both of the suites in different weeks.

Above are for the whole weeks the users were examined. However, due to the incorrect usage by user or the system reliability issue, we do not have the same amount of days for the particular signals as we want to use in the following procedure. The signals we use are RR Interval, Breathing Rate and Minute Ventilation. Take RR Interval as an example, all available RR interval we got from examinees are displayed as figure 9, which actually comes less than the total days of participating:

Figure 9. Valid RR Interval Days for Each Sensor Suite

Across the users, we have 101 days of available RR Interval data for SEM and 236 days of that for Hexoskin. Although they are much less than participating days, still it gives a sufficient amount for the following analyzing.

Also, since the study is about the alcohol craving, we have to draw the relevant survey answers for the users from their survey data. There are three types of surveys asking about alcohol drinking situations: Initial Drinking, Drinking Follow-up and Random Survey. In total, 403 drinking surveys are exported from the whole survey data set, no matter which types they belong to. Figure for these are skipped.

Data collected from real world is always mixed with noise. In work [1, 4, 5 and 11], they all have to screening out the data before they can make further discoveries. In our study, after learnt from and understanding the data for actual participants, I would summary the inappropriate data coming from the following sources:

* Data missing: this happened either when sensor suites transmitting data points to smartphone, or on the way from smartphone to server. Transmission protocol is not perfect, sometimes the server stops working and we have to recovery data from backups on phone manually, which are archived encrypted files and may not work all the times.
* Blank or extreme value: these situations are inevitable. Most of the blank values are when users turned on their devices but not yet wearing on, or undressed them. So most of the time, blank values happened at the beginning and ending of the dataset. Extreme value seems to be related to our sensors’ build-in settings. Every time it was unable to get correct readings, because of dry-skins, loose-suites or other similar reasons, it gave out these strange numbers. Usually it is zeros or to the largest number of its resolutions.
* Low confidence data: during the measurement of our sensor, under some scenarios, the sensor still output readings, however they may not be accurate values. The good thing is, usually there comes along with another indicator, showing confidence of the data. We considered confidence above 80 or 85 are good ones.
* Noise: again this is inevitable in everywhere an electronic devices involved. Noise may come during a normal measurement. We would apple filters to screen it out as much as we can.
  + 1. Pipeline

Basically, the job of our data analysis pipeline is to work against the above effects and situations, getting cleaned data, extracting useful features and classifying over those. The pipeline of data analysis is shown in figure 10.

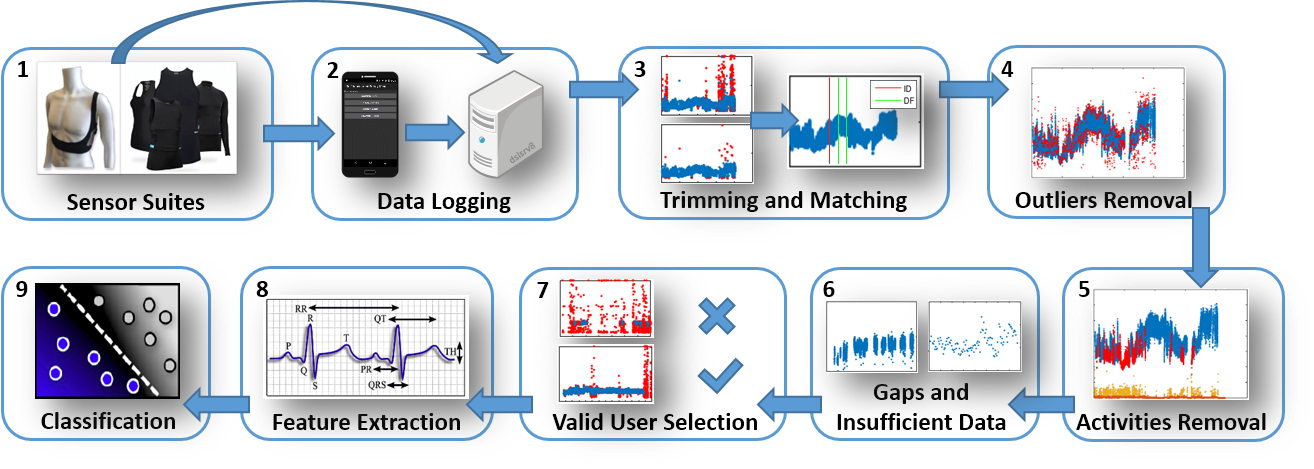


Figure 10. Pipeline of Data Analysis

The first two steps represent how the data coming into our cleaning procedure, which already been described detailed in last chapter. It shows that our server would gather the physiological measurements either directly from the sensor or received by the smartphone. Also, survey scores and geographic information about the user are transferred there. The server would save them all into csv files.

Steps from 3 to 7 are data cleaning procedure. All these sub-steps are implemented by Matlab and automatically process till the end. We can still change some parameters to control the processes. Step-by-step procedures would be discussed in next section.

The last one is the Classification step, it represents how our classifiers perform training and testing procedures, which requires that the output of our data clean procedure and data should be cleaned, labeled and all timestamp matched.

* 1. Data Cleaning
     1. Step-by-Step Procedures

1. Trimming and Matching

Trimming is to address the blank values or zero values as discussed earlier in this chapter, and it is pretty simple. The first x minutes and last y minutes will be dropped. Here x and y are parameters that can be modified. In this thesis, I set x to 5 and y equals 10. The parameter may not work for every single beginning and ending situations. We still have other steps to address the potential unremoved blank values. Also, data points that marked as LOW confidence by the internal sensor suite indicators are also removed from the data sets.

Matching does the alignment job among data coming from different source: physiological data, survey data and location data. The may use different time zone tags and in respective format. This section of code convert them into timestamps using uniform format.

1. Outliers Removal

In work [4 and 6], several preprocessing procedures are taken to prepare the data for training. Commonly they both consider a one-minute window of physiological data to be the size of data concerned. Based on their references and studies, human ECG responds very quickly to stimulators, and one minute is proven to be a successful candidate.

To screen out the noise mentioned above, we want the data to go through a moving window filter. Take RR Interval as example, first of all, we need to compute the moving average value for each minute, then RR Intervals that measured to be more than two standard deviations (2SD) away from the moving average, are marked as outliers and screened out. The hypothesis behind this, as stressed in many related works, is that the RR Interval of human are considered to be normally distribution. Figure 11 illustrates the procedure, the green curve shows the moving average in each one minute window, the bottom curve plots the standard deviation. Outliers away from two of that are marked as red, those are the data points we ignored.

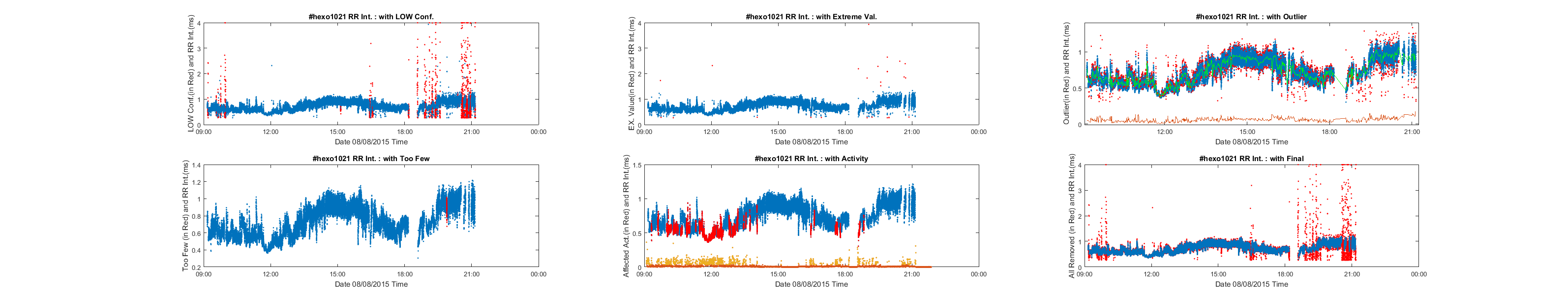


Figure 11. Illustration of Outliers Removal

Again since here we assume RR Intervals follow normal distribution, to account for between-person differences, the base features can be z-score normalized before calculating any other statistical features.

1. Activities Removal

Physical activity used to be a main confounder of our desired signal. In work [6, 12 and 13], they all applied similar procedure to remove the activity-affected data points and minimized its influence. As we all know, activities especially the intensive ones, would dramatically increase human heart rate and breathing rate. Comparing to physical activities, the effects of our study object – alcohol drinking – are too small to be distinguished from those.

We had the 3-axis on-body accelerometer equipped on the suites. With the 3 gravity readings, we could compute the activities magnitude by taking square root of summation of the square for each direction. This magnitude is independent of the orientation of the placement for accelerometer sensors. The standard deviation of magnitude, from 10-second windows, is used to discriminate physical activities based on a threshold. In the thesis, the threshold is 0.035. Figure 12 shows the determination procedure.

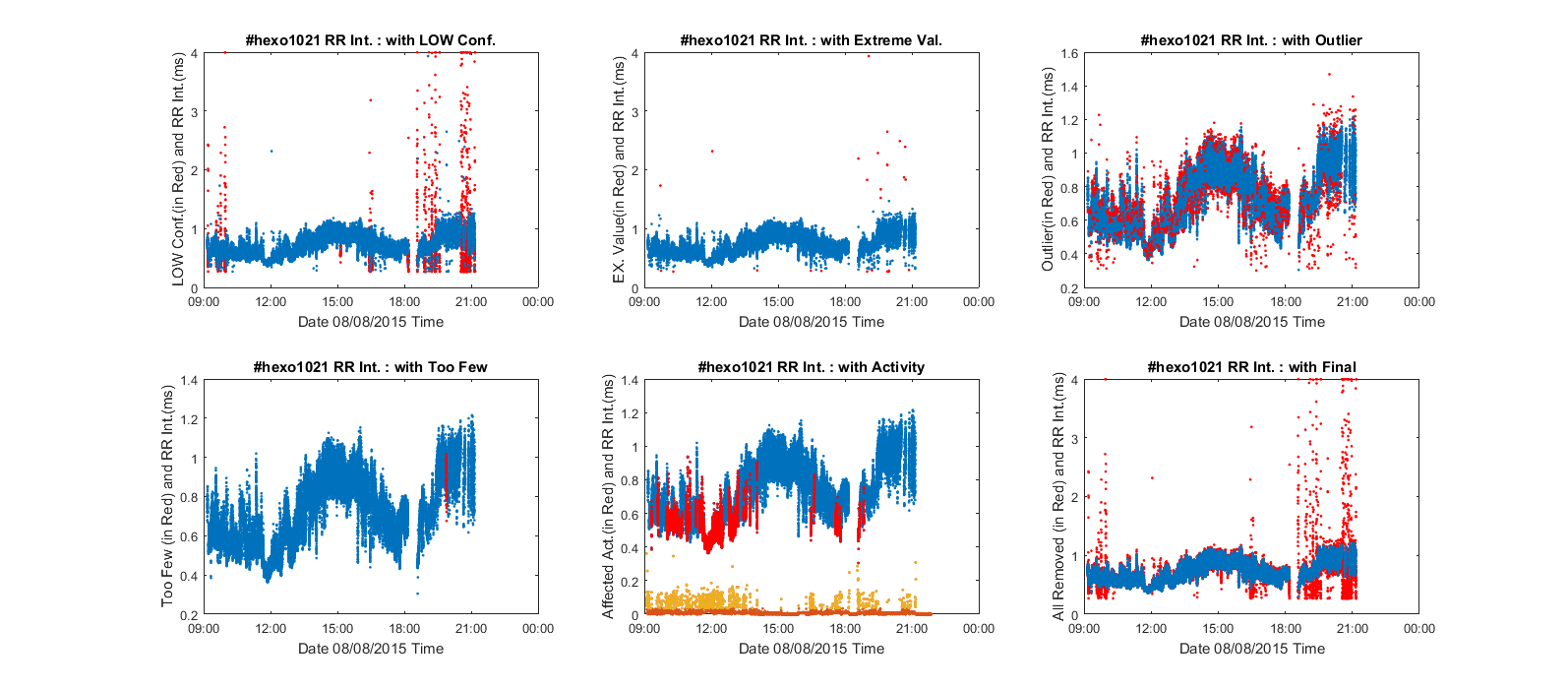


Figure 12. Illustration of Discriminating Activity-Affected Signals

Since the threshold judger is in 10-second window and I applied it to one-minute windows, each of these one-minute physiological signals would get 6 flags, the majority side of the flags would decide whether this current minute belongs to under-activity or free-activity class.

1. Gaps and Insufficient Data

Giant gaps and data that is too short to constitute a reasonable window need to be dealt with before our final step. As shown in figure 13.

Gaps in here mean that there is more than 10 minutes duration of time without or with a rather small amount of data points. Usually these period is caused by user took off the suite or something else making the sensor didn’t contact with body very well. Insufficient data straightforward means that in a one-minute window, there are less than 50 data points. Again, this is the setting I am using in this thesis work and can be changed easily in parameter settings.

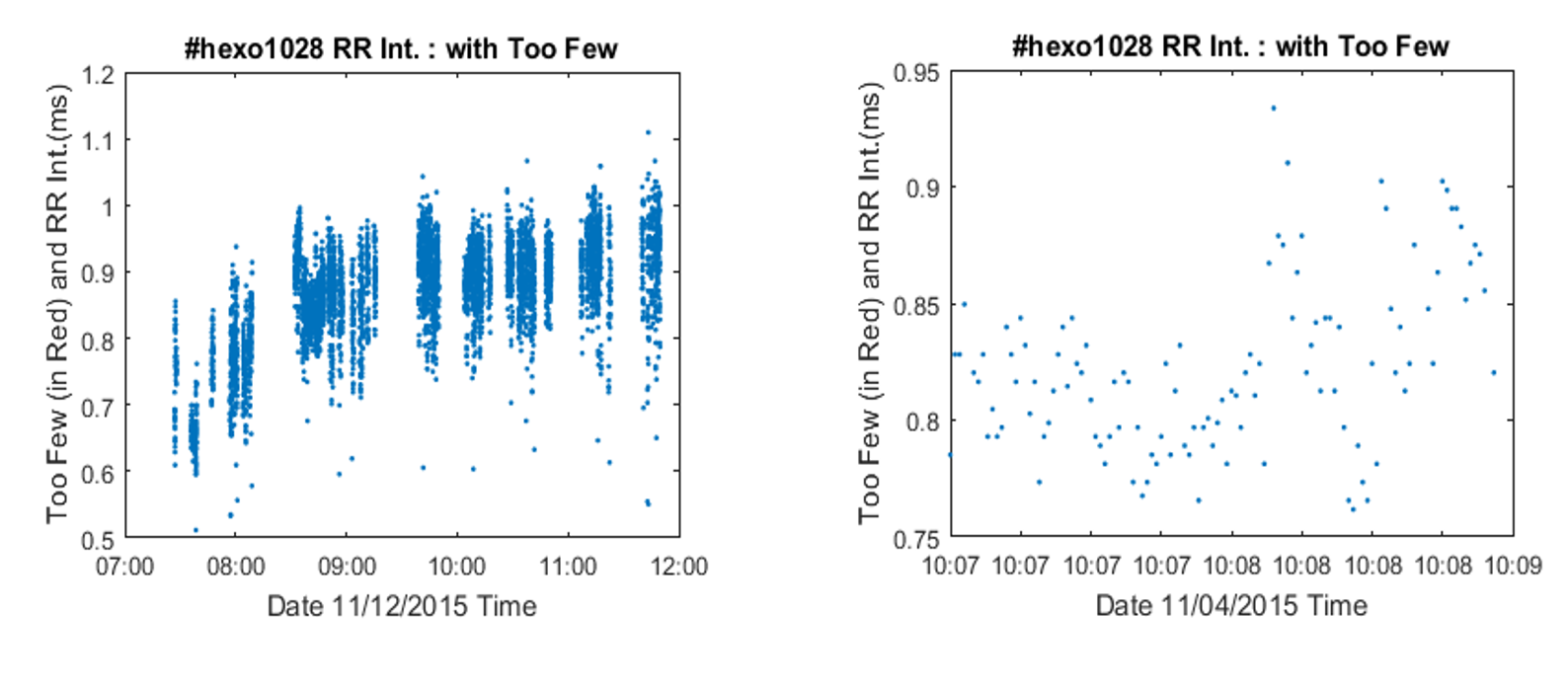


Figure 13. Illustration of Gaps and Insufficient Data

1. Valid User Selection

Although our total user ID numbering went to thirties, not every single one was valid. Some user dropped out midway and some may not followed our in-residence study protocols correctly. The ways of our screening invalid users out were threefold.

First of all, if the user somehow quit or paused during the in-residence study, I won’t keep the folder of that user in our building path, which means none of the unfinished data would be examined. Secondly, every time when a user finished her/his study, we would review the data. If none of the data contains alcohol drinking activities, this user may unsuitable to our study. But the data may still good and helpful with other studies, so I would keep the data but then put down the user ID to a blacklist, upon which would be looked by following steps and skipped over. Finally, not all days of data for a passed user are good, which means that we may drop certain whole days of a particular user. We screen the valid days of user according to data valid rate. The way we examined data valid rate is represented in figure 14:

Figure 14. Valid-Minute Rate Screening

The orange lines show how many data in minutes before it go through the whole cleaning procedure, meanwhile the blue curves show how many of them left afterwards. If the difference is too much, or there is nearly no data left, it indicates that day of data is also useless, and I would put it into blacklist, too. Note that the figure shows the minutes of each user for the whole study period. I also apply this valid-rate screening to each day, if the valid rate for a day is lower than 25%, which means 75% of the data is removed, the whole day with its user ID would be both recorded into the blacklist.

This is a two-phase screening, the blacklist here is actually generated when we review all the processed data after we run the whole pipeline for a first time. When the black list is generated, no more changes are needed in future.

* + 1. Results for Cleaned Data

We now would take a look at the cleaned data and some quick statistics for it. Table 1 shows the results, in minutes, after we apply methods to clean the raw signals, for RR Interval.

Table 1. Statistics of RR Interval Data for the Two Sensor Suites

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Users | Drink  days | Valid  days | Total  days | Drink  rate | Valid  rate |
| SEM | 9 | 24 | 76 (35725 mins) | 101 (46087 mins) | 31.58% | 77.52% |
| Hexoskin | 13 | 34 | 140 (47243 mins) | 236 (74902 mins) | 24.29% | 63.07% |
| Total | 21 | 58 | 216 (82968 mins) | 337 (120989) | 26.85% | 68.57% |

We can tell some daily routines and drinking habits of the users from this table as being examined. Data cleaning would furthermore drop some days of data based on our rules. 76 out of 101 days for SEM and 140 out of 236 for Hexoskin are defined as valid days, which count as 35725 and 47243 minutes respectively. The data valid rates are 77.52% for SEM and 63.07% for Hexoskin. It may be known from here that almost 1/3 of the raw data are not able to be used in final steps. Among these valid days, SEM users have 24 days containing alcohol drinking activities, and that days for Hexoskin users are 34. The drinking rates are 31.58% and 24.29% respectively. Moreover, SEM users wore the suites and produced averaged 7.8 hours of valid data per day and Hexoskin users produce 5.6 hours of that. In addition, we also see that for both of the two, one user are screened out make the total users 9 for SEM and 13 for Hexoskin.

* 1. Feature Extraction

After the raw data go through our cleaning procedure, outliers, low-confidences and extreme values are removed, data points from different sources are matched up with timestamp accordingly, confounders caused by intensive activities are screened out and alcohol drinking episodes duration are labeled. The next step is to calculate the features. I looked up many materials, learnt from biologists and psychological professors to determine what are useful physiological features that most likely to be affected by drinking activities. Refer to work [4, 5 and 6], many features extracted from ECG and RIP proven to be helpful. Guided by these, following features are chosen to train the classification models. Here I will introduce these features first by category and then visualize them.

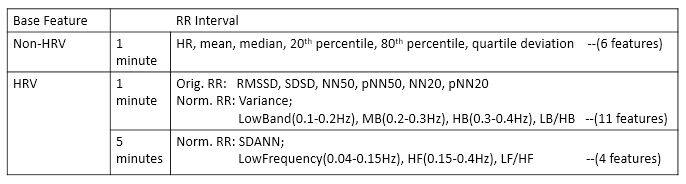
* + 1. Feature Definition

1. HRV (Heart Rate Variability) related features

Heart rate variability refers to the complex beat-to-beat variation in heart rate produced by the interplay of sympathetic and parasympathetic neural activity at the sinus node of the heart [9]. A full ECG signal may contain more aspects of body reactions to the environment, but usually the data is massive and hard to be examined. HRV features can be extracted from timing of heartbeats alone [10]. This means that we can calculate HRV features based on RR Interval signals, regardless of drifts or spikes of ECG itself. Although these features only capture limited subsets of the information in ECG signal, we decide to try this on in our early stage since so many similar works shown its feasibility.

Now we can have the full list of our heartbeats based features. As shown in table 2, all these features are computed from a base signal – RR Interval. RR Interval is the fraction of heartbeats that are considered as normal beat lengths, usually corresponding to the peak of the QRS complex of the ECG waves. We can have features from itself such as mean, median and these are also called Non-HRV features in our summary table. The others are like the variability of RR and showing a variety of characteristics of RR.

Table 2. HRV Features



For non-HRV features, HR is the reverse of RR by definition, the rests are pretty straightforward. Quartile deviation is the difference of upper quartile Q3 and lower quartile Q1.

For HRV features, we then divided them into two domains, time-domain and frequency-domain. Each domain can be viewed based on a certain windows size. Gathering them up, we would have 1-minute window time-domain features, such as RMSSD, SDSD, and NN50, etc. Also we could have 1-minute window frequency-domain features which contain Low-Band Energy, Middle-Band Energy, High-Band Energy and their ratio LB/HB, similar for 5-minute window size features.

RR intervals are also called NN intervals in some circumstances. A short description for the features in table are as below:

* RMSDD: the root mean square difference between adjacent RR Intervals.
* SDSD: the standard deviation of the successive differences between adjacent NNs.
* NN50: the number of pairs of successive RRs that differ by more than 50ms.
* pNN50: the proportion of NN50 divided by total number of NNs.
* SDANN: the average and standard deviation of NNs.
* Band Energy: the total spectral power of all RR Intervals within pre-defined bands. LowBand is 0.1-0.2Hz, MiddleBand 0.2-0.3Hz and HighBand 0.3-0.4Hz. And LB/HB is the ratio of power in the LB to HB.
* Frequency Energy: actually these are similar with above, the only difference is these are based on 5-minute window. I use name Frequency instead of Band as above.

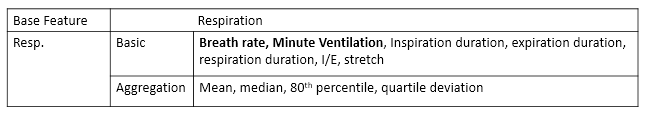
Another thing that worth to be mentioned here is, to account for between-person differences, we had a z-score normalization procedure before this step. So some of the features are calculated on the normalized data while some are on original readings, based on the properties of each of the features.

1. Respiration related features

RIP records the whole aspects of our breath. In order to compute respiration related features, we need to identify each of the breathing cycles. Here we define a breathing cycle as an inhalation and an exhalation period. In the RIP graph, the breathing cycle would look like a cycle from a particular valley to the next valid valley. We need some peak-peak detection algorithm to do so. Fortunately, our sensor suite already provides a few features similar to that and we can simple use those and produce more.

We intended to investigate 11 features as shown in the table 3. Breathing Rate is the number of breathing cycles per minute. Minute Ventilation is the volume of inhaled air recorded in a minute. Stretch is the volume of air breathed from the last breathing cycle. Inhalation Duration refers to the length of time for the maximum expansion of the inspiration part of a breathing cycle. Others are straightforward. I/E is the ratio of Inhalation duration and Exhalation duration.

Table 3. Respiration Features



1. Other features

There are still other features concern about ECG signal. There is substantial evidence showing that human heart would respond differently under the influence of activities such as alcohol drinking [11]. These ECG based features, other than those we mentioned above, are proven to be useful on the acute physiological effects on the heart. Also, Respiratory sinus arrhythmia (RSA) is another feature sometimes may be helpful in physiological data classification [4]. RSA is heart rate variability in synchrony with respiration, by which the RR Interval on an ECG is shortened during inspiration and prolonged during expiration. It is calculated by subtracting the shortest RR Interval from the longest RR Interval within each breathing cycle.

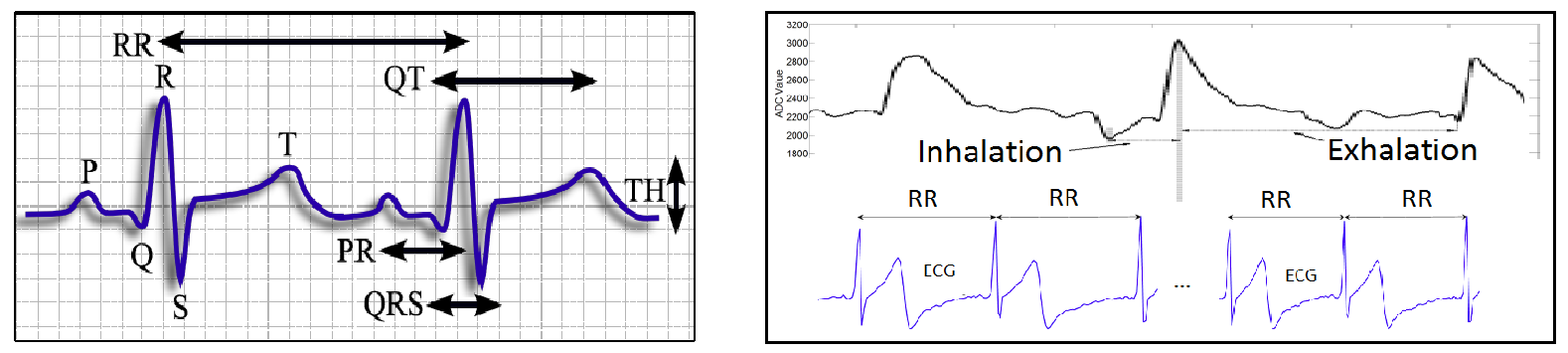


Figure 15. Illustration of Other ECG features (left) and RSA (right)

* Other ECG based features:

There comes definitions to many different waves according to figure 15. Besides RR interval, we can have QT which refers to Q to T interval, so as PR, QRS and TH. Each of them makes a single feature, and the ratio between two of them could be another. There are papers [11] study and utilize these as features. As a preliminary work, we haven’t involved these into our model building yet.

* RSA: the difference between the shortest RR Interval and longest RR Interval within each breathing cycle.
  + 1. Visualization

Although we may not be able to find any evidence of alcohol drinking from physiological data by eyeball, it would still be interesting to visualize these features.

First of all, figure 16 is a histogram showing the distributions of each of the 23 features, take Hexoskin data as an example:

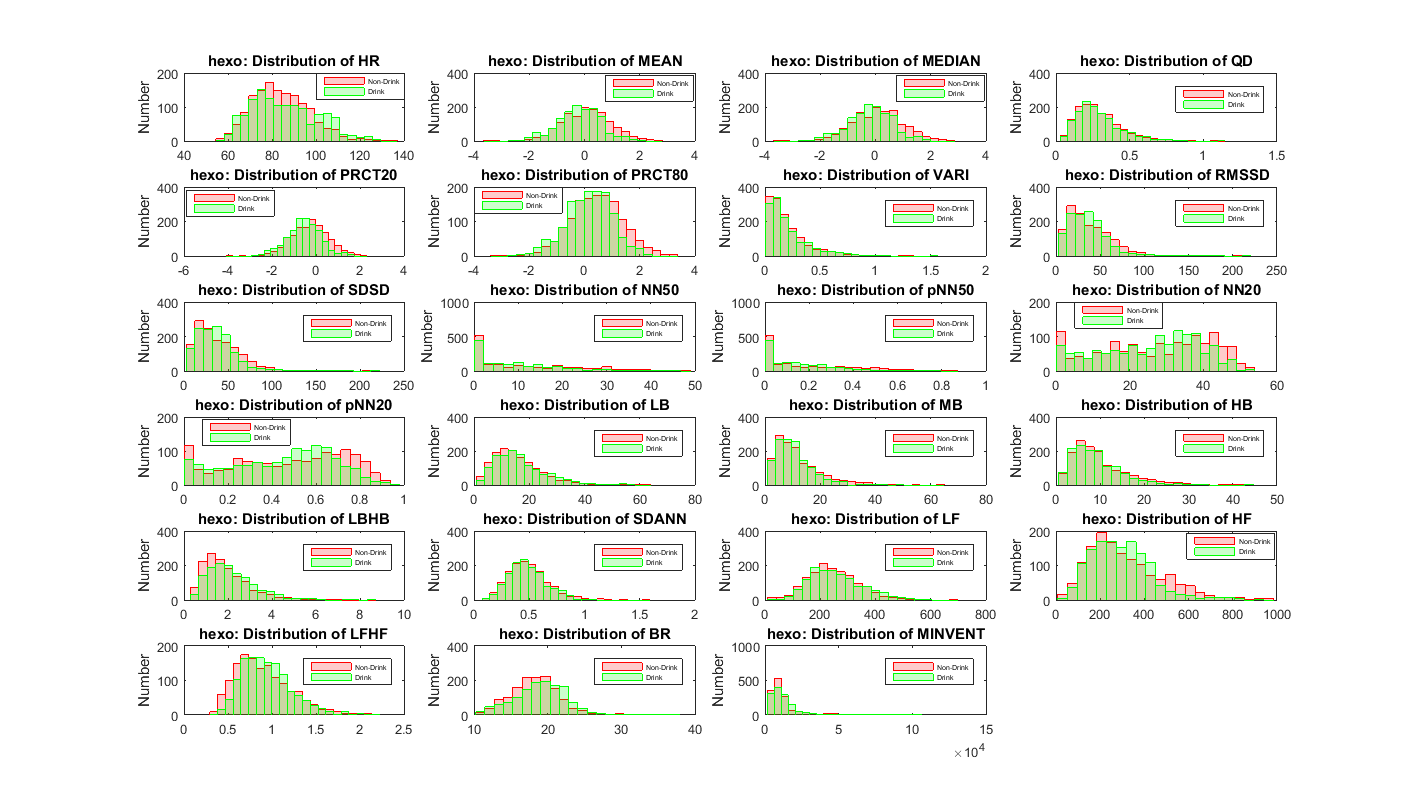


Figure 16. Distributions of All 23 Features, Hexoskin Suite

In this graph, blue color shows the minute of features with non-drinking class, while the red color represents all features under drinking class. On the top-left sub graph, we can see that, heart rate for Hexoskin users ranges from 53.8 to 136 bpm, and breathing rate (row 6 column 2) focuses on around 20 cycle per minute. For the rests, since some of them are normalized before calculation, we cannot directly understand the digits easily. Finally, we can learn from the whole graph that there seems to be just very little differences between the two classes, which would make it very hard to build a suitable classification model.

And remember the above figure is for all of the users in Hexoskin suite for the whole study days. We now take a close look at one particular day for a specific user of Hexoskin. Figure 17 shows user 1021’s HRV related features on August 19, 2015, with three times of alcohol drinking activities, one Initial Drinking followed by two Drinking Follow-ups, happened in about 2 hours.

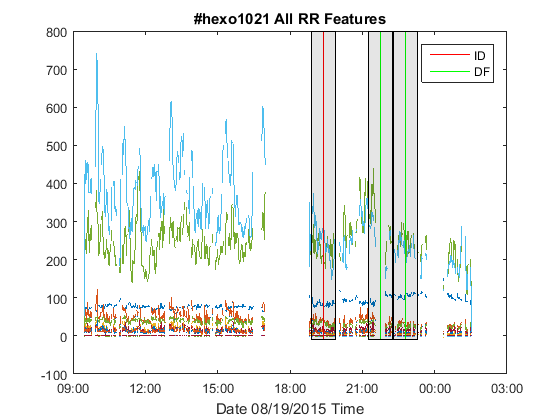


Figure 17. HRV Features for One User Whole Day

* + 1. Feature Analysis

So far, we had calculated a few features of physiological data for all the participants, including HRV features for SEM suite and both HRV and RIP features for Hexoskin suite. Now we would like to make some initial analysis on those features and trying to make some discoveries. We examine the morphological view of features with help of heat map first, then we try to investigate based on statistical inferences.

1. Morphological Features

First we still focus on the distribution of each feature and its relationship with drinking activities. I learn the way from paper [6] that dosage level of alcohol intakes may be a huge factor, so I plot the histogram of each feature, represented as heat-map, with different dosage levels.

Here is the procedures. Back to our survey questions asking about drinking activities, we had recorded ‘How many drinks did you have’ for each time the user reported a drink activity. Levels range for 1 through 6 and above 6. And because there were so few survey reports with 4 or more drinks in one drinking episode, here I re-categorized the dosage levels to a single dosage, 2 or 3 dosages, and 4 or above, representing in the histo-heatmap. Take the whole user datasets of SEM suite for example, histogram to the feature of Heart Rate represented as heat map looks like figure 18.

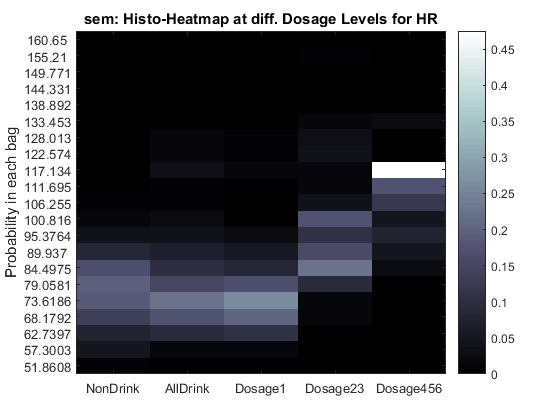


Figure 18. Histogram Represented as Heat-map for Heart Rate, SEM Suite

Heart Rate ranges from 51.8 to 160.6 bpm and is split into 20 bags horizontally. Meanwhile the color bar on its right shows the percentage of data fall into each bag, according to drinking, non-drinking and three new dosage levels. We could see a clear trend that heart rate of non-drink is a bit lower than that with drink, and the more dosage of drink, the higher of heart rate. This shows that the feature of Heart Rate is affected by alcohol drinking activities as expected. The Hexoskin suite dataset shows the similar result and its graph is omitted at this time.

1. Pairwise Correlations

Then considering that since we have so many features all for similar type, there might be highly correlations among these features.

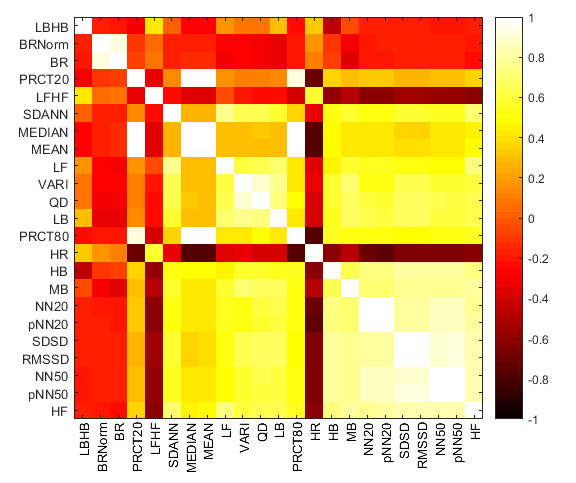


Figure 19. Pairwise Correlations of All 23 Features, Hexoskin Suite

For example in Non-HRV category we have both mean and median of the cleaned signals as two features. Clearly they could be closely correlated. I try to examine these relations by computing all pairwise correlations among the features. After cleaning up, Hexoskin suite datasets give 23 features for HRV/Non-HRV, Breathing Rate and Minute Ventilation. Here I plot their pairwise correlations in square matrix, as figure 19. Color bar represents the correlations from -1 to 1.

Features are sorted here by the sum of its absolute correlation values against others. Remember that -1 means negative correlated and this could also mean the two features are also much likely to each other. That’s the reason why I sort by absolute values in the graph. It also make the sequence meaningful. Feature in front place would have the least correlation with others and may contain much information comparing to others.

Another interesting view is, we could count how many features in different features categories rank higher to investigate the importance of that category.

Table 4. Ranks of Features in Each Category, Hexoskin Suite

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Non-HRV | 1min HRV | 5mins HRV | Respiration |
| 1st ~ 5th | 1 | 1 | 1 | 2 |
| 6th ~ 10th | 2 | 1 | 2 |  |
| 11th ~ 15th | 3 | 2 |  |  |
| 16th ~ 20th |  | 5 |  |  |
| 21st ~ 23rd |  | 2 | 1 |  |
| Total | 6 | 11 | 4 | 2 |

For example, in the first 5 places, PRCT20 is from Non-HRV, LBHB is from 1-minute HRV, LFHF if from 5-minute HRV, and finally BR and BRNorm they actually are the same feature before and after normalization and is from Respiration category. The exactly dispersed distribution of features shows that all the categories we tried to look into are somehow important. A table of sequence rank for all features is shown below.

1. Principal Component Analysis

Principal component analysis (PCA) is a statistical procedure converting a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables which is called principal components. In our feature space, there might be many features correlated to others as we discussed above. So it is a good place to apply PCA here. We already calculated pair wised correlation between every two features, what we need to do is to apply them into an orthogonal transformation. Weka can easily do this for us and here we show the Eigen Decomposition result in table 5.

Table 5. Principal Component of 25 Raw Features

|  |  |  |  |
| --- | --- | --- | --- |
| Eigenvalue | Proportion | Cumulative |  |
| 12.12956 | 0.48518 | 0.48518 | -0.26pnn50-0.26nn50-0.259hf-0.257rmssd-0.257sdsd... |
| 3.7931 | 0.15172 | 0.63691 | -0.409prct20-0.355median-0.353mean-0.288prct80... |
| 2.24631 | 0.08985 | 0.72676 | -0.424brnorm-0.395br-0.356minv-0.336minvnorm... |
| 1.6162 | 0.06465 | 0.79141 | 0.504minvnorm+0.473minv+0.309lfhf+0.234prct80... |
| 1.43361 | 0.05734 | 0.84875 | -0.538br-0.5brnorm+0.325minv+0.28minvnorm... |
| 0.84088 | 0.03364 | 0.88239 | 0.67 lbhb-0.361hb-0.313vari+0.256nn20+0.255pnn20... |
| 0.67652 | 0.02706 | 0.90945 | 0.492lf+0.454sdann-0.323vari+0.279hf-0.237qd... |
| 0.47814 | 0.01913 | 0.92857 | 0.718lfhf-0.319hf-0.265mb+0.207nn20+0.204pnn20... |
| 0.41771 | 0.01671 | 0.94528 | -0.428qd-0.378nn20-0.354pnn20+0.279sdsd+0.279rmssd... |
| 0.28063 | 0.01123 | 0.95651 | -0.628hr+0.333sdann-0.263mb+0.263rmssd+0.263sdsd... |

The original feature space size is 25. After decomposition transformation, the space size is reduced to 10, however, based on the cumulative numbers, they still carry 95.65% of the information as all of the raw feature space does.

* 1. Classification

We want to train and test a physiological classifier model using the features we got in the above steps. Basically, the features are 1-minute or 5-minute measurements from ECG and RIP. Overall, we compute a total of 25 features that are used to train the classifiers.

* + 1. Data Labeling

Before training the models, we need to decide and label ground truth of classes. We want to try a binary classifier with drinking vs non-drinking classes, although drinking can be distinguished by different dosage levels, we do this as preliminary work. Another concern is that comparing to the whole data of non-drinking class, we don’t have too much data for drinking, which means that if we divided them into more classes, the data for each class would be too little. What’s more, a binary classifier is much simpler and easier to start up.

For a regular episode of human, drinking can lasts for hours, which means that an episode of alcohol in our study cannot be represented as a single timestamp, instead it will be a duration of time consumption. If you look at the figures containing labels of drinking in the previous sections, you would find we can only get timestamps of the drinking from each of the surveys. So expending the duration of time for each drink episode is needed in this concern.

Another issue about expansion is expending parameters, how and by what method should we expend a drinking episode. After consulting from Psychological professors and referring related works, we got the idea that alcohol would stay and effect in human body for a duration about two hours, then the effects may decay. We would consider a one-hour expansion in total for each time a drink activity happened to users. It may be covered and reprehensive for the relatively strong phase of alcohol effect instead of the whole duration of drinking. After a few trials, we then decided to expend the recorded timestamp as the center, marking half hour before and half hour after that, consisting a total of one hour’s data as drinking class.

The drinking class is labeled as 1, while non-drinking class as 0 in the following experiments and results. Figure 20 illustrates how we do the expansion and labeling job. Vertical red line labels the timestamp of an Initial Drinking reported by Hexoskin user 1021 on August 05, 2015, and the shadowed area indicates the time duration we adjusted for this drink episode. As narrated above, half hour before and half hour after consisting a one-hour episode in total.

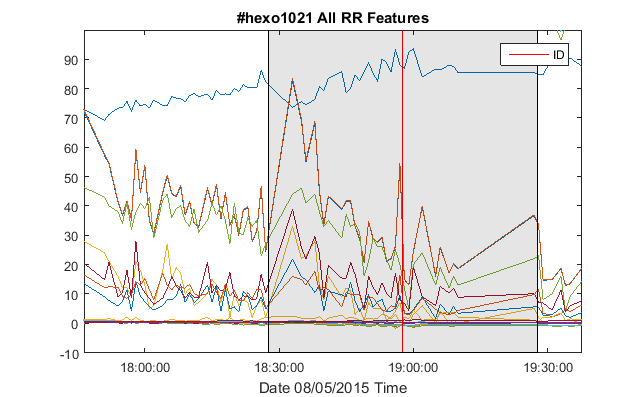


Figure 20. Illustration of Drinking Timestamp Expansion, to One Hour

* + 1. Classifiers

As mentioned in previous sections, we have 21 HRV related features for SEM suite and 25 features of both HRV and RIP features for Hexoskin suite and labels for them. Selected features and labels are used in WEKA, which is popular as a visualized machine learn toolbox, to train the classifiers. Weka supports so many different classifiers, I tried three types of classifiers on our dataset, a J48 Decision tree, an adaptive boosting J48 decision tree (AdaBoost J48) and a support vector machine (SVM). J48 decision tree is a simple and standard classifier, the model when built up may be understandable by us. And also, it doesn’t require too much computational resources. We can then try to interpret what happened based on the structure of the trained model. An adaptive boosting version of J48 is a generalized technique for improving the performance of the original decision tree classifier. Besides, SVM is known to perform well for high dimensional feature space and we can apply different kernel types with it. Other similar classifiers are also tried during the process, such as Random Forest, Naïve Bayes. The result are similar or worse and will not be listed in this thesis.

* + 1. Performance Measurements

Evaluation measurement is the process of assessing how well the results satisfied our query intent. We would consider the following parameters as the evaluate measurement for our experiments, they are: Confusion Matrix, Accuracy, Kappa and ROC. Confusion Matrix, also called Error Matrix, is a specific table layout that allows visualization of the performance of a classifier. Accuracy is used as a statistical measure of how well a binary classification test correctly identifies. Kappa is another statistic of measuring precision. It tells the agreement between two raters who each classify n items into c mutually exclusive categories. The equation for calculating kappa is:



Where Po is observed proportionate agreement, Pe is the probability of random agreement. A kappa above 0.75 is as substantial agreement of a good precision, 0.4 to 0.75 as fair to good and below 0.4 is as poorly precision. Finally, Receiver Operation Characteristic (ROC) curve is a graphical plot that illustrates the performance of binary classifier system as its discrimination threshold is varied. The numbers in our following tables represent the area under this curve in a 1x1 graph.

# **EXPERIMENTAL RESULTS**

* 1. Experimental Design

The final experiments are processed with the help of Weka. In Weka I trained the three types of classifiers, a J48 decision tree, an Adaboost J48 decision and a SVM. I used 10-fold cross validation to obtain the performance evaluation matrices of all classifiers, based on different situations of the two sensor suites.

In order to make the results comparable, I set the same benchmark across all the situations, which is 50%. This means I always keep the class of drinking and the class of non-drinking half and half. Since the total available valid samples of drinking minutes are much less than that of non-drinking, there needs a way to draw the same amount of drinking data from non-drinking dataset.

There are many ways to do the draw. In some related work they set different weights to the classes due to their portion regarding the total. In this thesis, I just apply a simpler one, it just draws the similar numbers of drinking data from the data points of non-drinking set. The drawing is random and I always make five or more different rounds to compare their results. Based on my study, the results were stable with just a few difference.

Another possible improvement to this drawing method, come up with my advisors, is a little more sophisticated yet may more account for the cycle of a day. In this new way we still randomly select same amount of data from non-drinking dataset, however, the dataset is divided into several candidate groups, which are: Morning, Afternoon, Evening and Night. Referring to the drinking data points and their time in a day, the counterpart of non-drinking data points would only be selected from the same candidate groups, i.e. the same time of a day. The reason to do so is, human may have internal routine in a day, and this especially reflect on our physiological reactions. When we wake up in the morning our heart rate might be the lowest in a day, and become higher until noon, and reduce again thereafter. This new way may allow the selection follow the trend and make the classification more accurate. However, this thesis didn’t go that far, yet we can always use current implementation as the comparison of the new drawing method. If time allowed, these would be the first I want to try to improve the performance.

* 1. Classification Results

Let’s go over some quick numbers about the datasets which we want them to be classified, before I can presenting the final results. We have two types of sensor suites: SEM and Hexoskin. For some raw file data format issue, we only have HRV features calculated for SEM, of which are 21 features. For Hexoskin we have both HRV and RIP features, of which are 25 features, the 4 more than SEM are Respiration features. From table 6, we can quickly learn the general statistics of the two:

Table 6. Quick Statistics of Data to Be Classified

|  |  |
| --- | --- |
| SEM | Hexoskin |
| HRV/Non-HRV features only | Both HRV and RIP features |
| 76 days with 24 drinking days | 140 days with 34 drinking days |
| -- | 1320 minutes of drinking data |
| 1777 minutes of drinking data | 2125 minutes of drinking for HRV only |

We notice that although Hexoskin suite group has more drinking days than SEM, when consider both ECG and RIP features, the total minutes labeled as drinking is smaller than those of SEM. The reason is, the ECG and RIP came from different sensor channels, when we merge the two sources of data together, we actually get the intersection of the two, but many of them could not be simultaneous completed all the time. The intersection would usually be smaller than either of the two, which leads to a relatively smaller group of drinking dataset for Hexoskin. In addition, the number of Hexoskin suite group for HRV only is 2125 minutes, which is more than that of SEM as expected, the comparison is shown in the bottom of table.

* + 1. ECG Features Only

1. SEM Suite

Table 7 shows the performance of J48 decision tree classifiers when trained and tested only with ECG features for SEM suite. Class 0 is for non-drinking while class 1 for drinking. We can see that most of the points are classified correctly, and the classification accuracy is 78.5%, precision is 0.5696, which is an acceptable result. Confusion matrix is shown at the right of the table.

Table 7. Classification Results on ECG Features, SEM Suite

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Kappa** | **ROC** | **Confusion Matrix** | | |
|  |  |  |  |  | **0** | **1** |
| SEM  (1777) | 78.5% | 0.5696 | 0.801 | **Non-drink** | 1377 | 423 |
| **Drink** | 347 | 1430 |

The Results for all the three classifiers are summarized in table below. What we can learn from the table is, among the three machine learning methods, Adaboost J48 decision tree classifier gives the highest performance, with 82.3% accuracy.

Table 8. Results Comparison on ECG Features, SEM Suite

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Accuracy** | **Kappa** | **ROC** |
| J48 Decision Tree | 78.5% | 0.5696 | 0.785 |
| J48 with Adaboost | **82.3%** | 0.6467 | 0.823 |
| SVM (normalized) | 66.8% | 0.337 | 0.668 |

1. Hexoskin Suite

Classification results for Hexoskin suite data on ECG related features only are collated in table 9. There are 2125 minutes of data as drinking class and randomly select 2200 minutes for that of non-drinking.

Table 9. Classification Results on ECG Features, Hexoskin Suite

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Kappa** | **ROC** | **Confusion Matrix** | | |
|  |  |  |  |  | **0** | **1** |
| J48 Decision Tree | 66.8% | 0.3357 | 0.701 | **Non-drink** | 1489 | 711 |
| **Drink** | 725 | 1400 |
| J48 with Adaboost | 68.6% | 0.371 | 0.748 | **Non-drink** | 1508 | 692 |
| **Drink** | 668 | 1457 |
| SVM (normalized) | 60.1% | 0.2021 | 0.601 | **Non-drink** | 1262 | 938 |
| **Drink** | 789 | 1336 |

Similar results are given with Hexoskin suite for ECG features only. However, if we compare it with SEM results, we could find it’s a little lower than those of SEM. I infer that the reason might be, Hexoskin dataset contains more participants than SEM does, which could bring more between-user differences. Our z-score normalization method is trying to address this, but since it is a pretty rough way, it may not do the job very well. Another reason I could consider is since SEM sensor suite is FDA approved, yet Hexoskin is not. Maybe SEM holds a higher capacity of getting stable and accurate data than that of Hexoskin does.

* + 1. ECG and RIP Features

Respiration features are only calculated for Hexoskin suite, and would be examined as followed.

1. Classification Results

Remind that Hexoskin dataset has 13 valid users with 140 days of data, among the days, 34 of them contain records of alcohol drinking. With ECG features only, the drinking days produce 2125 minutes of valid labeled data. And with both ECG and RIP features, that number reduced to 1320 minutes. I draw the similar amount of data from non-drinking and make the benchmark to 50%. The results for Adaboost J48 decision tree classification are shown in table 10.

Table 10. Classification Results on ECG and RIP Features, Hexoskin Suite

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Kappa** | **ROC** | **Confusion Matrix** | | |
|  |  |  |  |  | **0** | **1** |
| ECG & RIP  (1320) | 74.1% | 0.4821 | 0.819 | **Non-drink** | 1039 | 361 |
| **Drink** | 343 | 977 |
| ECG Only  (1320) | 70.9% | 0.4179 | 0.771 | **Non-drink** | 1004 | 396 |
| **Drink** | 395 | 925 |
| RIP Only  (1320) | 59.6% | 0.1855 | 0.628 | **Non-drink** | 1013 | 387 |
| **Drink** | 712 | 608 |
| ECG & BR  (1320) | 71.2% | 0.423 | 0.785 | **Non-drink** | 1011 | 389 |
| **Drink** | 395 | 925 |
| ECG & MV  (1320) | 72.5% | 0.4499 | 0.802 | **Non-drink** | 1043 | 357 |
| **Drink** | 390 | 930 |

From the table we can see, with both types of the features, the classifier gives the highest accuracy 74.1%, which indicates with addition information provided by Breathing Rate and Minute Ventilation, the performance is improved.

I also try to isolate the feature category with ECG only and RIP only, and to make them comparable, the total amount minutes of drinking class is still 1320, which is under the same dataset. We can see that with ECG features only gives 70.9% accuracy, which as acceptable, but RIP features only is not. These indicate that unlike the result brought by work [4], for our alcohol drinking study, respiration itself cannot be the only feature category doing the inference job well.

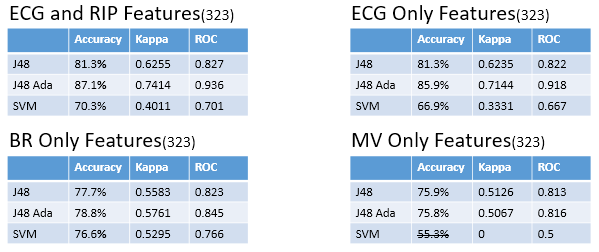
A mix of both ECG and RIP features will range between the two results given above. ECG features either with Breathing Rate or with Minute Ventilation would give about 72% of accuracy.

1. Classification Results on Case Study

In order to take a closer look at the properties of human alcohol drinking habits, I decided to review the procedure from another perspective. The case study would consider the most recent five users, which are 1032, 1031, 1030, 1029 and 1028. During the data collection phase, we actually updated the application to fix some bugs at time, while my colleagues getting to know better about how to train the participants and have more tips about how to avoid certain common mistakes on using the sensor suites. Based on these experiences, we feel like that the most recent users would have higher quality and accuracy of data collected. That’s another reason why I want to exam more in detailed with these users.

Table 11 shows the classification results for ECG and RIP features (top left), for the most recent 5 users. As comparison, I also list the results for ECG features only (top right), Breathing Rate feature only (bottom left) and Minute Ventilation feature only (bottom right).

Table 11. Classification Results for Case Study, Hexoskin Suite



The results shown, decision tree classifiers works better in ECG related features and in RIP related features, SVM works best on Breathing Rate. Breathing Rate alone seems can also provide an acceptable result, which is agreed with some other related work. However, Minute Ventilation only seems failed on SVM classifier.

# **CONCLUSIONS**

This thesis describes the whole system of alcohol drinking craving study, from the data collection distributed client-server system, to the data cleaning and analysis pipeline, and gives preliminary results on classification between drinking and non-drinking based just on physiological readings.

The mobile ambulatory system utilize the convenience brought with recent wearable technologies. It can provide in-resident sensing in an afforded way, of which the quality is still comparable with traditional clinical ways. And I applied OOP design along with the implementation, came up with algorithms to guarantee the delivery of certain randomized surveys, also, implemented a way to ensure the security of data transmission, both strength and efficient.

The data preprocessing and analyzing section shown the characters of real-world data and the power of machine learning methods. The raw digits collected were massive and in a mess. Considered the sources of a variety of noises, I tried to eliminate those in steps and finally get a clean enough, yet much less in size data for the following procedure. The classification results is good enough to reveal the influence of alcohol drinking on human physiological readings.

The work and its predecessors reveal the feasibilities of remote sensing inference and context-aware assessment, and may have other applications when it comes with other aspects of studies.

# **FUTURE WORK**

More work has been done in part of [18], where I also participated in. Although we have made some progress, there is still much future work needed and to improved.

In work [4], the pater mentioned a way commonly used in stock marketing curves analysis, which call MACD line. Moving Average Convergence/Divergence (MACD) is a trading indicator originally used in technical analysis of stock prices. It followed the long term and short term trends of the cycles, which is also proven to be helpful on human physiological data on cocaine usage. We can also try to apply it on our alcohol usage in some degree.

Another recent hot topic on data science is deep learning. We come up with the idea to use tensor flow to handle our data as well. We would transform our physiological data, which is considered as 1-D signals, to 2-D graphs using moving Fourier transform. That way all the features are hidden inside tons of pictures and would be revealed by deep learning and neural networks

Features would also be refined and calculated, to facilitate the process and make improvements.

# **REFERENCES**

[1]. George Carey Simos. "How Much Data Is Generated Every Minute On Social Media?" Experts Talk. 2015.

[2]. IDC Analyze the future. "IDC: Smartphone OS Market Share 2015, 2014, 2013, and 2012" 2015.

[3]. Ertin, Emre, et al. "AutoSense: unobtrusively wearable sensor suite for inferring the onset, causality, and consequences of stress in the field." Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems. ACM, 2011.

[4]. Plarre, Kurt, et al. "Continuous inference of psychological stress from sensory measurements collected in the natural environment." Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on. IEEE, 2011.

[5]. Hovsepian, Karen, et al. "cStress: towards a gold standard for continuous stress assessment in the mobile environment." Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2015.

[6]. Hossain, Syed Monowar, et al. "Identifying drug (cocaine) intake events from acute physiological response in the presence of free-living physical activity." Proceedings of the 13th international symposium on Information processing in sensor networks. IEEE Press, 2014.

[7]. Raj Jain, "Internet of Things and Smart Cities: Challenges and Issues," Keynote at 2nd IEEE International Workshop on Sensors and Smart Cities (SSC2016), St. Louis, MO, May 18, 2016

[8]. Ganesan, Ramachandran, Mohan Gobi, and Kanniappan Vivekanandan. "A Novel Digital Envelope Approach for A Secure E-Commerce Channel." IJ Network Security 11.3 (2010): 121-127.

[9]. Ingjaldsson, Jon T., Jon C. Laberg, and Julian F. Thayer. "Reduced heart rate variability in chronic alcohol abuse: relationship with negative mood, chronic thought suppression, and compulsive drinking." Biological psychiatry 54.12 (2003): 1427-1436.

[10]. Liu, David, and Mark Ulrich. "Listen to Your Heart: Stress Prediction Using Consumer Heart Rate Sensors." (2014).

[11]. Natarajan, Annamalai, et al. "Detecting cocaine use with wearable electrocardiogram sensors." Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, 2013.

[12]. Atallah, Louis, et al. "Sensor placement for activity detection using wearable accelerometers." 2010 International Conference on Body Sensor Networks. IEEE, 2010.

[13]. Rahman, Md Mahbubur, et al. "Are we there yet?: Feasibility of continuous stress assessment via wireless physiological sensors." Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics. ACM, 2014.

[14]. Adams, Phil, et al. "Towards personal stress informatics: Comparing minimally invasive techniques for measuring daily stress in the wild." Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[15]. Kennedy, Ashley P., et al. "Continuous in-the-field measurement of heart rate: correlates of drug use, craving, stress, and mood in polydrug users." Drug and alcohol dependence 151 (2015): 159-166.

[16]. Shi, Ruiqi, et al. "mAAS--A Mobile Ambulatory Assessment System for Alcohol Craving Studies." Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual. Vol. 3. IEEE, 2015.

[17]. Andres A. Perez, et al. "A Smartphone-based System for Clinical Gait Assessment." The 2nd IEEE International Conference on Smart Computing (SMARTCOMP 2016), May 2016

[18]. Peng, Sun, et al. "ADA - Automatic Detection of Alcohol Usage for a Mobile Ambulatory Assessment System" The 2nd IEEE International Conference on Smart Computing (SMARTCOMP 2016), May 2016