SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Doctorado

Metodología para el desarrollo y evaluación de modelos predictivos

presentada por
MC. Wendy Aracely Sánchez Gómez

como requisito para la obtención del grado de
Doctora en Ciencias de la Computación

Director de tesis
Dra. Alicia Martínez Rebollar

Codirector de tesis
Dr. Oscar Mayora Ibarra

Cuernavaca, Morelos, México. Febrero de 2019.

cenidet®
Centro Nacional de Investigación
y Desarrollo Tecnológico

"2019, Año del Caudillo del Sur, Emiliano Zapata"

ESC\FORDOC09

Cuernavaca, Morelos, 30/Enero/2019

ASUNTO: ACEPTACIÓN DEL TRABAJO DE TESIS DOCTORAL

DR. NOÉ ALEJANDRO CASTRO SÁNCHEZ
JEFE DEL DEPARTAMENTO DE CIENCIAS COMPUTACIONALES
PRESENTE

Los abajo firmantes, miembros del Comité Tutorial de la Tesis Doctoral de la alumna **M.C. WENDY ARACELY SÁNCHEZ GÓMEZ**, manifiestan que después de haber revisado su trabajo de tesis doctoral titulado **"METODOLOGÍA PARA EL DESARROLLO Y EVALUACIÓN DE MODELOS PREDICTIVOS"**, realizado bajo la dirección de la **DRA. ALICIA MARTÍNEZ REBOLLAR** y co-dirección del **DR. OSCAR MAYORA IBARRA,** el trabajo se ACEPTA para proceder a su impresión.

ATENTAMENTE
Excelencia en Educación Tecnológica®
"Conocimiento y tecnología al servicio de México"

DRA. ALICIA MARTÍNEZ REBOLLAR
CENIDET

DR. OSCAR MAYORA IBARRA
FBK

DR. NOÉ ALEJANDRO CASTRO SÁNCHEZ
CENIDET

DR. JOAQUÍN PÉREZ ORTEGA
CENIDET

DR. JOSÉ CRISPÍN ZAVALA DÍAZ
CENIDET

DRA. MARÍA YASMÍN HERNÁNDEZ PÉREZ
INEEL

C.c.p.: M.T.I.. María Elena Gómez Torres / Jefa del Depto. de Servicios Escolares
Dr. Gerardo Vicente Guerrero Ramírez / Subdirector Académico
Expediente

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Centro Nacional de Investigación y Desarrollo Tecnológico

"2019, Año del Caudillo del Sur, Emiliano Zapata"

ESC\FORDOC010

Cuernavaca, Morelos, 14/febrero/2019

M.C. WENDY ARACELY SÁNCHEZ GÓMEZ
CANDIDATA AL GRADO DE DOCTORA
EN CIENCIAS DE LA COMPUTACIÓN
PRESENTE

Después de haber sometido a revisión su trabajo final de tesis titulado "**METODOLOGÍA PARA EL DESARROLLO Y EVALUACIÓN DE MODELOS PREDICTIVOS**", y habiendo cumplido con todas las indicaciones que el jurado revisor de tesis le hizo, le comunico que se le concede autorización para que proceda a la impresión de la misma, como requisito para la obtención del grado.

Reciba un cordial saludo.

ATENTAMENTE
EXCELENCIA EN EDUCACIÓN TECNOLÓGICA®
"CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO"

DR. NOÉ ALEJANDRO CASTRO SÁNCHEZ
JEFE DEL DEPTO. DE CIENCIAS COMPUTACIONALES

ESTADOS UNIDOS MEXICANOS
S. E. P.
CENTRO NACIONAL DE
INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
CIENCIAS COMPUTACIONALES

Agradecimientos

Mi más profundo agradecimiento a mi directora de tesis, la Dra. Alicia Martínez Rebollar, por formarme como investigadora, por su dedicación en este trabajo doctoral, por la motivación que siempre me transmitió, por la confianza que depositó en mí durante todo mi trayecto por CENIDET desde la maestría, y por todas sus enseñanzas tanto académicas como personales: ¡Gracias eternas doctora!

Al Dr. Oscar Mayora y al Dr. Venet Osmani, quienes fortalecieron ampliamente la investigación y me brindaron una gran oportunidad de trabajar al lado de ellos en la Fondazione Bruno Kessler.

A los miembros del comité tutorial: Dr. Joaquín Pérez Ortega, Dr. Noé Castro Sánchez, Dra. Yasmín Hernández Pérez, Dr. Javier Ortiz, y Dr. Crispín Zavala, quienes me acompañaron a lo largo del doctorado y me aportaron sus conocimientos y comentarios que permitieron enriquecer mi formación doctoral.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), al personal que labora, por brindarme los recursos para mi formación como investigadora y como ser humano. Me siento orgullosa de ser egresada de este honorable centro de investigación.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico que me brindó para realizar mis estudios de doctorado y la estancia en el extranjero.

Al Instituto Tecnológico de Matamoros, en especial a la Lic. Ana Isabel Lerma González, por su apoyo brindado para la realización de mis estudios de doctorado.

A mi esposo Alberto Arturo Flores Hernández, quien me animó a realizar juntos los estudios de doctorado, que me apoyó en todo momento y que durante este periodo unimos nuestras vidas.

A mi papá, mis hermanos Perla y Luigi, mi mami y a mi demás familia, que me siguieron apoyando desde casa. A mi mamá que me sigue apoyando desde el cielo.

A mis amigos y compañeros, por todos los momentos compartidos.

Y al más importante, a mi Dios, que no me suelta de su mano.

¡Gracias totales!

Resumen

El modelado predictivo es el proceso de desarrollar un modelo matemático que genera una predicción precisa. Hay muchos dominios en los que se podría implementar el modelado predictivo. Muchos investigadores se han centrado en los problemas de salud mental de los trabajadores, como el estrés. El estrés es una respuesta fisiológica a los desafíos mentales, emocionales u otros desafíos físicos que enfrentan los humanos en sus actividades de la vida real, incluso en su entorno laboral, que es una de las principales fuentes de estrés en la actualidad. Los desarrolladores de software representan uno de los grupos más expuestos al estrés. Para evitar que el estrés se vuelva crónico y cause un daño irreversible, es necesario detectar los diferentes niveles de estrés.

El estrés ha sido estudiado hace muchos años por diferentes enfoques. Los expertos en psicología han utilizado cuestionarios y los expertos médicos han usado sensores fisiológicos estacionarios para reconocer el estrés. Han surgido nuevos métodos computacionales en los que se crean modelos predictivos basados en datos de diferentes fuentes tecnológicas. Existen algunas metodologías que permiten el desarrollo de aplicaciones de modelos predictivos como CRISP-DM, pero no proporciona información detallada para dominios específicos.

Esta investigación se centra en la definición de una metodología para el desarrollo y evaluación de modelos predictivos. La metodología es una extensión de la metodología CRISP-DM para aplicaciones de minería de datos. La metodología fue validada a través de un estudio de caso sobre reconocimiento de estrés. Por lo tanto, se desarrolló una aplicación de minería de datos para encontrar patrones y determinar el nivel de estrés de un desarrollador de software en un período de tiempo determinado utilizando métodos computacionales. Se ha hecho una comparación de la metodología CRISP-DM y la metodología propuesta. Se han encontrado similitudes entre las fases de las metodologías.

Se compararon los modelos predictivos construidos bajo diferentes esquemas. Como se esperaba, el modelo predictivo individual mostró el mejor resultado con una F-M de 0.88. El modelo general obtuvo una F-M de 0.58, mayor que el modelo de usuarios similares el cual obtuvo una F-M de 0.51. Se utilizaron cinco clasificadores: k-NN, Naïve Bayes, Random Forest, C4.5 y AdaBoost. Todos los clasificadores obtuvieron una F-M alrededor de 0.50, excepto Naïve Bayes que obtuvo una F-M ligeramente más baja de 0.42. Se hizo la comparación de los modelos predictivos construidos por métodos computacionales. El método de computadora mostró mejores resultados que los métodos vestibles con una F-M de 0.82, 6% más alta que el método vestible en modelos predictivos individuales, y con una F-M de 0.58, 4% más alta que el método vestible en modelos predictivos generales. El modelo predictivo general que utiliza todos los atributos también obtuvo una F-M de 0.58. Por lo tanto, el método de computadora se puede utilizar para el reconocimiento del estrés sin perder precisión y *recall*.

Se hizo la comparación de los modelos predictivos construidos por país. Los resultados fueron similares con un promedio de F-M de 0.555. Finalmente, se hizo la comparación de los modelos predictivos construidos por método de computadora y por país. El método de computadora mostró mejores resultados que el método vestible, excepto en un experimento. El promedio de F-M del método de computadora fue de 0.67.

Abstract

Predictive modeling is the process of developing a mathematical tool or model that generates an accurate prediction. There are many domains in which predictive modeling could be implemented. Many researchers have focused on the mental health issues of workers like stress. Stress is a physiological response to mental, emotional, or other physical challenges that humans confront in their real-life activities, including in their working environment which is considerate one of today's major sources of stress. Software developers represent one of the groups most exposed to stress. To prevent stress from becoming chronic and provoking irreversible damages, it is necessary to detect the different levels of stress.

Stress has been studied many years ago by different approaches. Psychology experts have mostly used questionnaires or surveys to quantify stress. Medical experts have utilized stationary physiological sensors to recognize physiological or neurological changes on the body in the presence of stress. With the emergence of new technologies, novel computational methods have appeared in which predictive models are created based on data from different sources such as smartphone, wearables, computer, vision, and audio devices. There are some methodologies that allow the development of predictive model applications. An example of these methodologies is CRISP-DM but it does not provide detail information for specific domains.

This research focuses on the definition of a methodology for the development and evaluation of predictive models. The methodology is an extension of CRISP-DM methodology for data mining applications. The methodology was validated through a case study on stress recognition. Therefore, a data mining application to find patterns to determine the level of stress of a software developer in a certain period of time was developed. We have used computational methods. We have compared the CRISP-DM methodology and our proposed methodology. We have found similarities between the methodologies' phases.

We have compared the predictive models built by different schemes. As expected, individual predictive model scheme showed the best result with an F-measure of 0.88. General model scheme obtained an F-measure of 0.58, higher than the similar subject scheme which obtained an F-measure of 0.51. We used five classifiers: *k*-NN, Naïve Bayes, Random Forest, C4.5, and AdaBoost. All the classifiers obtained an F-measure around 0.50, except for NB which obtained a slightly lower F-measure of 0.42.

We have compared the predictive models built by computational methods. Computer method showed better results than wearable methods with an F-measure 0.82, 6% higher than wearable method in individual predictive models, and with an F-measure of 0.58, 4% higher than wearable method in general predictive models. General predictive model using all attributes also obtained an F-measure of 0.58. Hence, the computer method may be used for stress recognition without losing precision and recall.

We have compared the predictive models built by country. The results were similar with an average F-measure of 0.555. Finally, we have compared the predictive models built by computational method and by country. Computer method showed better results than wearable method, except in one experiment. The average F-measure of computer method was 0.67.

Contents

Chapter 1. Introduction

Predictive modeling is the process of developing a mathematical tool or model that generates an accurate prediction [1]. Predictive modeling is also a process through which a future outcome or behavior is predicted based on the past and current data at hand. Geisser defines predictive modeling as the process by which a model is created or chosen to try to best predict the probability of an outcome [2]. The process includes taking the current information, sift through data looking for patterns that are relevant to our problem, and return answers. The term predictive modeling may stir associations such as machine learning, pattern recognition, and data mining but the ultimate objective is the same: to make an accurate prediction.

Predictive modeling is a technique that uses mathematical and computational methods to predict an event or outcome. A mathematical approach uses an equation-based model that describes the phenomenon under consideration. The model is used to forecast an outcome at some future state or time based upon changes to the model inputs. The model parameters help explain how model inputs influence the outcome. The computational predictive modeling approach differs from the mathematical approach because it relies on models that are not easy to explain in equation form and often require simulation techniques to create a prediction. This approach is often called black box predictive modeling because the model structure does not provide insight into the factors that map model input to the outcome.

There are many domains in which predictive modeling could be implemented. An example is the transportation system. Train stations, buses, and cars could be equipped with technology for collecting information and predict the time arrivals. Another example is in the healthcare domain. Several research studies highlighted the significance of monitoring human physical activities on improving the healthcare and treatment processes [3]. Some diseases require continuous monitoring for the patient's activities to measure their reflections on health status. Many researchers have focused on the mental health issues of workers like stress.

In many countries, the number of persons suffering stress has increased as a consequence of the accelerated rhythm in the actual life [4]. The impact of this psychological disease is so big that the World Health Organization recognizes it as one of the great epidemics of modern life [5]. Stress is very common and it has a high cost in terms of employees' health, absenteeism and lower performance [3]. The cost of work-related stress, anxiety, and depression in Europe is estimated by the European Commission to be €617 billion per year. Also, stress in the United States has an annual cost of US$200 billion [6]. According to the Fourth European Working Conditions Survey (EWCS), work-related stress was reported by 22% of workers from 27 Member states of the European Union [7]. Moreover, higher prevalence of stress has been reported in North America, where 55% of the population has reported increased workload having a significant impact on physical and mental health as described in APA Survey [8]. In Mexico, according to survey carried out by Regus, a human resources company, and Mexican Social Security Institute, 75% of Mexicans who suffer stress attributed it to their work activities [9].

Stress is a physiological response to mental, emotional, or other physical challenges that humans confront in their real-life activities [10], including in their working environment which is considerate one of today's major sources of stress [11]. Job stress appears in demanding situations in means of content, organization and work environment [12]. The employees feel the job demands exceed their capabilities to keep them under control [12], [13]. These demands are not only related to high workload or long working hours, but also to high perceived stress, low social support from colleagues and managers, or to the individual characteristics of each one like the education and competitiveness [14].

The effects of stress appear at different levels such as behavioral, peripheral, physiological, and cognitive [11]. Usually, it causes physiological reactions like an increase of heart rate, sweating, hand trembling or mouth dryness [15]. It contributes to negative habits like smoking, drinking, bad diet and insomnia [16]. Long-Term exposure to stress may result in the development of gastric ulcers, or increased sensitivity to infections. It also leads to anxiety, tension, low self-esteem, social isolation, irritability, depression, and job-burnout which is a state of mental and physical exhaustion [17]–[19].

Software developers represent one of the groups most exposed to stress [20], [21]. The main task of software developers is to transform general informal understanding of a goal into a formal model using operators and components so that it is interpretable by the computer [22]. Software development is complex, purely intellectual, and accomplished through cognitive processing abilities [23]. Therefore, software developers work requires high mental concentration which exposes them to a major probability of stress [24].

There have been many efforts to engineer software construction processes but software development is deeply dominated by human factors [25], [26]. Software developers are creative human beings and their capabilities are affected by their emotions and moods [27]–[29]. For this reason, software developers are forced to develop efficient code in the short-term in order to accomplish deadlines which increase their stress levels.

Software developers have to deal with high workload [30]. Frequently, they feel pressure caused by their managers or clients and do not receive support from their colleagues [31]. They lack control over their daily duties and they are interrupted many times throughout the day. Moreover, software developers must update their programming knowledge constantly due to the rapidly evolving technologies [21], [32]. Stress has been identified as one of the determinants of Information Technology professionals turnover intention [33]. Frequently, software developers must take control of other people to work.

Software developers also have social incentives to contribute to many projects; prolific contributors gain social recognition and economic rewards. Software developers have long been pushing the limits on multitasking [34] because of the innate modularity of the development process and the independence of module processing. Multitasking, however, comes at a cognitive cost: frequent context-switches can lead to distraction, sub-standard work, and even greater stress [35]. All of these situations lead programmers to be stressed. To prevent stress from becoming chronic and provoking irreversible damages, it is necessary to detect the different levels of stress in a certain period using computational methods.

There are some methodologies that allow the development of predictive model applications. An example of these methodologies is CRISP-DM [36], which is an industry standard based on the practical, real-world experience of how people conduct data mining projects. CRISP-DM methodology is described in terms of a hierarchical process model, consisting of sets of tasks described at four levels of abstraction from general to specific: phase, generic task, specialized task, and process instance. CRISP-DM methodology does not attempt to capture all of the possible routes through the data mining process; it only describes the first two levels of abstraction. To develop a specific application, it is necessary to investigate and to define the other two levels of abstraction which is not always an easy task.

1.1 Problem Statement

Stress has been studied many years ago by different approaches. Psychology experts have mostly used questionnaires or surveys to quantify stress, for example, the "Perceived Stress Scale" [37], the "Job Content Questionnaire" [38], the "NIOSH Generic Job Stress Questionnaire" [39]–[41], the "Copenhagen Psychosocial Questionnaire" (COPSOQ) [42], and the "Workplace Stress Survey" [43]. These questionnaires have the benefits that they are inexpensive, do not represent too much effort from the researcher, and the data are easy to compile by predefined answers [44]. Also, they are very practical because they could be applied by the researcher or by other people even remotely without affecting validity or reliability. However, questionnaires have some disadvantages. The subjects' answers are highly subjective, and they could unconsciously hide information [45]. Also, the questionnaires are defined by a group of researchers who could let pass an important issue to the subject and not include questions about it [11]. Therefore, recognizing stress is a highly challenging task.

On the other hand, medical experts have utilized stationary physiological sensors to recognize physiological or neurological changes on the body in the presence of stress. Stationary refers that the subject cannot move without restrictions and cannot wear the sensor, on the contrary of new wearable sensors described later. Researchers have utilized many physiological features to recognize stress like cortisol [46]–[49], skin conductivity [50]–[56], heart rate variability [56]–[59], respiratory rate [50], [55], [60], electrocardiogram [50], [52], [61], [62], electroencephalography [61], [63]–[67], electromyograms [50], [68], blood volume pulse [56], [68], and body temperature [51], [53], [56], [69]. Physiological sensors are very precise and their validity is nowadays acknowledged [11], [50]. However, these sensors involve wires and other hardware that may limit movement. Consequently, subjects feel uncomfortable and they could feel stressed by the monitoring itself, and this could affect the data collected.

With the emergence of new technologies, novel computational methods have appeared [11]. These novel methods for stress recognition were developed in the last years as a result of an unprecedented evolution in consumer electronics and miniaturization. Others were made possible from a better understanding of stress and its effects on a human being at several levels: physiological, behavioral or physical. The first computational method to recognize stress is using smartphones features such as call logs, SMS logs, phone usage, accelerometer sensor, human-smartphone interactions, among others [19], [70]–[77]. The second computational method to recognize stress is using physiological features such as electrocardiography (ECG), heart rate (HR), and galvanic skin response (GSR) from wireless wearable devices [78]–[81]. The third computational method to recognize stress is using computer vision to detect stress using features such as pupil dilation, facial expressions, eye-tracking, and head movements [51], [82]–[85]. The fourth computational method to recognize stress is using speech features based on vocal cues such as speed, rhythm or intonation [86], [87]. And finally, the fifth computational method to recognize stress is using computer features such as mouse speed, mouse inactivity, mouse click rate, keystrokes dynamics, typing speed, key typing events, computer logging, among others [88]–[92]. All these different computational methods have their own characteristics, advantages, and disadvantages but all of them allow continuous monitoring of the users in their workplaces [93]. The monitoring is carried out in a non-intrusive manner, limiting any impact on subjects' routines and consequently their typical behavior. This means that users must be able to carry out their daily activities as if they were not being monitored.

The stress recognition is carried out by building predictive models based on collected data from a sample of subjects. This process could be done following a methodology like CRISP-DM, but it does not describe the steps in detail. Which steps are essential to building predictive models in healthcare domain? Which are the different techniques to apply in each step?

There have been works that have analyzed and compared the different methods for stress recognition. Alberdi et al. [14] have reviewed the recent works carried out in the stress recognition using three main methods, namely, psychological, physiological and behavioral methods in order to give hints about the most appropriate techniques to be used. Carneiro et al. [11] have evaluated and contrasted these novel methods in order to facilitate the stakeholder's decision towards which one to use based on how much their organization values aspects such as privacy, accuracy, cost-effectiveness or intrusiveness. But which method is better? Which one has better accuracy? Which one is more suitable to apply in software developers?

In recent related works, it has been common to build individual models and general models to classify stress levels [77], [94]. Individual models are trained and evaluated for each of the users using their own data. The general model consists of building the model with data from all the users. There is another interesting scheme that has been used in [73]: similar-users which consists of building the model with data from just a subset of similar users. The benefit to using similar-user scheme is that for any two users, their behavioral patterns across stress levels may be different. For example, a user may tend to be more active when he is stressed but another one may tend to be more sedentary when stressed. Which of these schemes works better? Individual models will have a better performance but how much difference is there between the performance of the individual model versus the general model? How well does the similar-users model will performance versus the other two schemes? Which scheme is more suitable to apply in software developers?

Related works have built models with data from all users. But what happens when these users are from different countries? Do they behave differently? Does the model created with data from users from one country perform well when testing on users from another country?

1.2 Solution overview

This research focuses on the definition of a methodology for the development and evaluation of predictive models. The methodology is an extension of CRISP-DM methodology for data mining applications. The methodology is validated through a case study on stress recognition. Therefore, a data mining application to find patterns to determine the level of stress of a person in a certain period of time is developed.

Two computational methods for stress recognition are compared which are wearable-based and computer-based methods. The wearable method offers convenient information about physiological, activity and sleep features. And the computer method is suitable to recognize stress on software developers who spent almost all day working with their computers. These methods are non-intrusive and cheap. The performance among the models using only wearable data, using only computer data, and using data from both methods is compared.

Models using three schemes were built which are individual models, general models, and similar-users. Individual models were trained and evaluated for each of the users using their own data with cross-validation technique. General models were built with data from all the users with leave-one-out technique, that is to say, the model was built with all users except one, and then the model was evaluated with data from the user that was left out. Similar-user models were built with data from a subset of similar users. The subset of similar users was defined through *k*-means clustering technique and Silhouette clustering quality index to determine the final number of optimal groups.

Also, models built using data of subjects from different countries were compared. We have built a general model with data from users from one country and evaluated with data from users from another country and vice versa.

1.3 Research goals

The main research goal of this thesis is to define a methodology for the development of predictive models and to evaluate the methodology through a case study on stress recognition. This research goal has been satisfied by dealing with the following sub-goals.

a) To extend the CRISP-DM methodology for data mining applications on stress recognition.
b) To build predictive models for stress recognition.
c) To compare the performance of stress predictive models using different schemes.
d) To compare the performance of stress predictive models built from different computational methods.
e) To compare the performance of stress predictive models of subjects from two countries.

1.4 Hypothesis

The hypotheses of the present research are:

- The stress predictive models can be developed by following the proposed methodology.
- The individual models perform better than general models. Similar models perform better than general models.
- Stress models, built from participants of one country, can be used to estimate the stress of participants from another country.

1.5 Organization

The rest of this document is organized as follows: Chapter 2 presents the theoretical framework which serves as the basis of the present work. Chapter 3 presents the related work that address the issue of stress recognition from the point of view of computer science. Chapter 4 presents the proposed methodology for the development and evaluation of predictive models. Chapter 5 presents the evaluation of the methodology through the case study on stress recognition. Finally, Chapter 6 presents conclusions and future work.

Chapter 2. Theoretical Framework

This Chapter presents the theoretical concepts this research work is based on.

2.1 Predictive modeling

Predictive modeling is the process of developing a mathematical tool or model that generates an accurate prediction [1]. Predictive modeling is a process that uses data mining and probability to forecast outcomes. Each model is made up of a number of predictors, which are attributes that are likely to influence future results. Once data has been collected for relevant predictors, a statistical model is formulated. The model may employ a simple linear equation, or it may be a complex neural network, mapped out by sophisticated software. As additional data becomes available, the statistical analysis model is validated or revised.

Predictive models are created to predict future unseen cases. The objective of predictive modeling is not to understand why something will or will not occur. Instead, we are primarily interested in accurately projecting the chances that something will or will not occur. Notice that the focus of this type of modeling is to optimize prediction accuracy. For example, it is not important to know why an e-mail filter thinks a message is spam. Rather, the important thing is to know that the filter accurately trashes spam and allows important messages to pass through the mailbox. Therefore, there is a tension between prediction and interpretation. While the primary interest of predictive modeling is to generate accurate predictions, a secondary interest may be to interpret the model and understand why it works. The unfortunate reality is that as we push towards higher accuracy, models become more complex and their interpretability becomes more difficult. This is almost always the trade-off we make when prediction accuracy is the primary goal.

The best predictive models are fundamentally influenced by a modeler with expert knowledge and context of the problem. This expert knowledge should first be applied in obtaining relevant data for the desired research objectives. While vast databases of information can be used as a substrate for constructing predictions, irrelevant information can drive down the predictive performance of many models. Subject-specific knowledge can help separate potentially meaningful information from irrelevant information, eliminating detrimental noise and strengthening the underlying signal. Undesirable, confounding signal may also exist in data and may not be able to be identified without expert knowledge.

The construction of an effective predictive model is laid with intuition and deep knowledge of the problem context, which are entirely vital for driving decisions about model development. Predictive modeling is not a substitution of intuition, but rather a complement. Traditional experts make better decisions when they are provided with the results of statistical prediction. Those who cling to the authority of traditional experts tend to embrace the idea of combining the two forms of knowledge by giving the experts statistical support.

Predictive modeling is one of the many names that refer to the process of uncovering relationships within data for predicting some desired outcomes. Since many scientific domains have contributed to this field, there are synonyms for different entities [1]:

- The terms instance, sample, data point, or observation, refer to a single, independent unit of data, such a customer, a patient, or compound. The term sample can also refer to a subset of data points, such as the training set sample.
- The training set consists of the data used to develop models while the test or validation sets are used solely for evaluating the performance of a final set of candidate models.
- The predictors, attributes, features, variables, independent variables, or descriptors are the data used as input for the prediction equation.
- Outcome, dependent variable, target, class, or response refer to the outcome event or quantity that is being predicted.

One of the most frequently overlooked challenges of predictive modeling is acquiring the right data to use when developing algorithms. By some estimates, data scientists spend about 80% of their time on this step. While predictive modeling is often considered to be primarily a mathematical problem, users must plan for the technical and organizational barriers that might prevent them from getting the data they need. Often, systems that store useful data are not connected directly to centralized data warehouses. Also, some lines of business may feel that the data they manage is their asset, and they may not share it freely with data science teams.

Another potential stumbling block for predictive modeling initiatives is making sure projects address real business challenges. Sometimes, data scientists discover correlations that seem interesting at the time and build algorithms to investigate the correlation further. However, just because they find something that is statistically significant does not mean it presents an insight the business can use. Predictive modeling initiatives need to have a solid foundation of business relevance.

2.2 CRISP-DM methodology

Cross-industry standard process for data mining, known as CRISP-DM, is an open standard process model that describes common approaches used by data mining experts. It is the most widely-used analytics model [95]. CRISP-DM was conceived in 1996 and became a European Union project under the ESPRIT funding initiative in 1997.

The CRISP-DM methodology is described in terms of a hierarchical process model, consisting of sets of tasks described at four levels of abstraction from general to specific: phase, generic task, specialized task, and process instance. Figure 2.1 depicts the four-level breakdown of the CRISP-DM methodology.

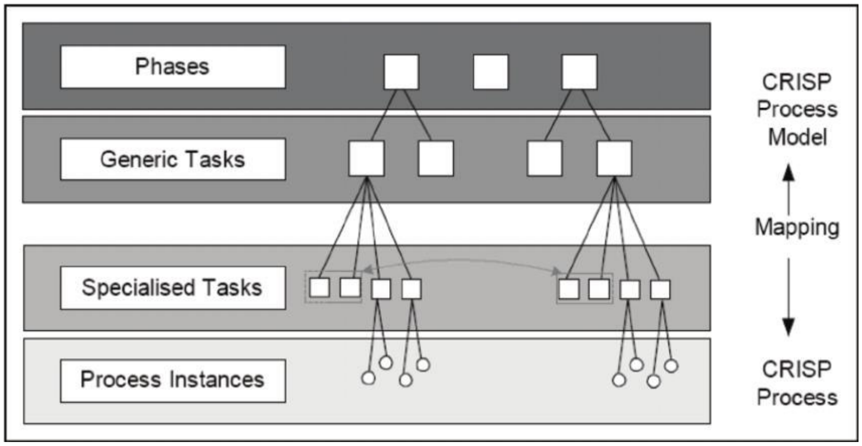


Figure 2.1 Four-level breakdown of the CRISP-DM methodology

At the top level, the data mining process is organized into a number of phases; each phase consists of several second-level generic tasks.

This second level is called generic because it is intended to be general enough to cover all possible data mining situations. The generic tasks are intended to be as complete and stable as possible. Complete means covering both the whole process of data mining and all possible data mining applications. Stable means that the model should be valid for yet unforeseen developments like new modeling techniques.

The third level, the specialized task level, is the place to describe how actions in the generic tasks should be carried out in certain specific situations. For example, at the second level there might be a generic task called clean data. The third level describes how this task differs in different situations, such as cleaning numeric values versus cleaning categorical values, or whether the problem type is clustering or predictive modeling. The description of phases and tasks as discrete steps performed in a specific order represents an idealized sequence of events. In practice, many of the tasks can be performed in a different order, and it will often be necessary to repeatedly backtrack to previous tasks and repeat certain actions. Our process model does not attempt to capture all of these possible routes through the data mining process because this would require an overly complex process model.

The fourth level, the process instance, is a record of the actions, decisions, and results of an actual data mining engagement. A process instance is organized according to the tasks defined at the higher levels, but represents what actually happened in a particular engagement, rather than what happens in general.

The CRISP-DM reference model

The current process model for data mining provides an overview of the life cycle of a data mining project. It contains the phases of a project, their respective tasks, and the relationships between these tasks. At this description level, it is not possible to identify all relationships. Relationships could exist between any data mining tasks depending on the goals, the background, and the interest of the user and most importantly on the data.

The life cycle of a data mining project consists of six phases, shown in Figure 2.2. The sequence of the phases is not rigid. Moving back and forth between different phases is always required. The outcome of each phase determines which phase, or particular task of a phase, has to be performed next. The arrows indicate the most important and frequent dependencies between phases. The outer circle in Figure 2.2 symbolizes the cyclical nature of data mining itself.

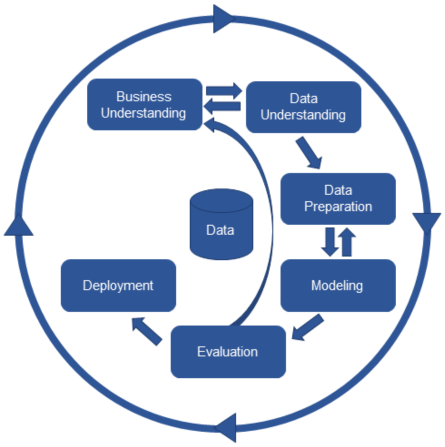


Figure 2.2 Phases of the CRISP-DM reference model

Data mining does not end once a solution is deployed. The lessons learned during the process and from the deployed solution can trigger new, often more-focused business questions. Subsequent data mining processes will benefit from the experiences of previous ones. In the following, we briefly outline each phase:

Business understanding

This initial phase focuses on understanding the project objectives and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

Data understanding

The data understanding phase starts with initial data collection and proceeds with activities that enable you to become familiar with the data, identify data quality problems, discover first insights into the data, and/or detect interesting subsets to form hypotheses regarding hidden information.

Data preparation

The data preparation phase covers all activities needed to construct the final dataset [data that will be fed into the modeling tool(s)] from the initial raw data. Data preparation tasks are likely to be performed multiple times and not in any prescribed order. Tasks include table, record, and attribute selection, as well as transformation and cleaning of data for modeling tools.

Modeling

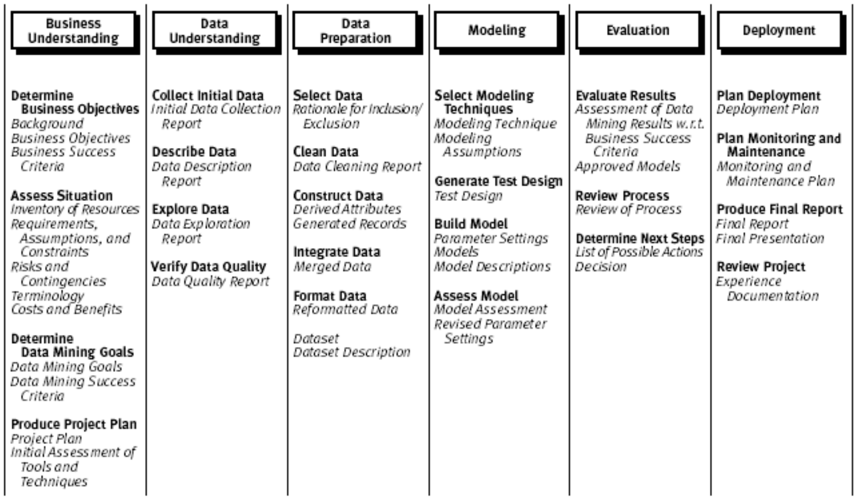
In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements on the form of data. Therefore, going back to the data preparation phase is often necessary.

Evaluation

At this stage in the project, you have built a model (or models) that appears to have high quality from a data analysis perspective. Before proceeding to final deployment of the model, it is important to thoroughly evaluate it and review the steps executed to create it, to be certain the model properly achieves the business objectives. A key objective is to determine if there is some important business issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached.

Deployment

Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that the customer can use it. It often involves applying “live” models within an organization’s decision-making processes—for example, real-time personalization of Web pages or repeated scoring of marketing databases. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise. In many cases, it is the customer, not the data analyst, who carries out the deployment steps. However, even if the analyst will carry out the deployment effort, it is important for the customer to understand up front what actions need to be carried out in order to actually make use of the created models. Figure 2.3 presents an outline of phases accompanied by generic tasks (bold) and outputs (italic).



Business Understanding

Determine Business Objectives
Background
Business Objectives
Business Success Criteria

Assess Situation
Inventory of Resources
Requirements, Assumptions, and Constraints
Risks and Contingencies
Terminology
Costs and Benefits

Determine Data Mining Goals
Data Mining Goals
Data Mining Success Criteria

Produce Project Plan
Project Plan
Initial Assessment of Tools and Techniques

Data Understanding

Collect Initial Data
Initial Data Collection Report

Describe Data
Data Description Report

Explore Data
Data Exploration Report

Verify Data Quality
Data Quality Report

Data Preparation

Select Data
Rationale for Inclusion/ Exclusion

Clean Data
Data Cleaning Report

Construct Data
Derived Attributes
Generated Records

Integrate Data
Merged Data

Format Data
Reformatted Data

Dataset
Dataset Description

Modeling

Select Modeling Techniques
Modeling Technique
Modeling Assumptions

Generate Test Design
Test Design

Build Model
Parameter Settings
Models
Model Descriptions

Assess Model
Model Assessment
Revised Parameter Settings

Evaluation

Evaluate Results
Assessment of Data Mining Results w.r.t. Business Success Criteria
Approved Models

Review Process
Review of Process

Determine Next Steps
List of Possible Actions
Decision

Deployment

Plan Deployment
Deployment Plan

Plan Monitoring and Maintenance
Monitoring and Maintenance Plan

Produce Final Report
Final Report
Final Presentation

Review Project
Experience Documentation

Figure 2.3 Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model

2.3 Stress

Although researchers have studied stress for more than a century, the stress definition is still debated [96]. Hans Selye defined stress as the non-specific response of the body to any demand for change [97]. Thenceforth, other definitions that take into account the coping abilities of each individual have been exposed [98], including the one of McEwen that defines stress as events, that are threatening to an individual, and which elicit physiological and behavioral responses [99].

Selye [97] distinguished the concepts eustress and distress, as positive and negative stress, respectively. Eustress appears with positive changes or demands that do not pose a problem for coping with or to adapt ourselves to the new situation. It can help us meet our goals and increase productivity [100]. Distress can be really harmful and can carry negative consequences. It is the most investigated aspect of stress and it is what in general terms is understood by stress. Besides, three levels of stress can be distinguished depending on the time of exposure to stressors. Acute stress is the innate flight-or-fight response in face of short-lasting exposure to stressors and it is not considered harmful [101]. Episodic stress appears when stressful situations occur more frequently, but they cease from time to time. It is associated with a very stressful and chaotic life. Finally, chronic stress, which is the most harmful, takes place when stressors are persistent and long-standing, such as family problems, job strain or poverty [100]. In order to avoid stress to reach the highest level and help to diminish the risks [102], it is necessary to detect and treat it in its earlier stages, i.e. when it is still acute or episodic stress.

Work-related stress has been defined as the emotional, cognitive, behavioral and physiological reaction to aversive and noxious aspects of work, work environments, and work organizations. It is a state characterized by high levels of arousal and distress and often by feelings of not coping [103]. Work-related stress is experienced when the demands of the work environment exceed the employees' ability to cope with or control them [104]. These demands are not only related to high workload or long working hours, but also to high perceived stress, low social support from colleagues and managers, or to the individual characteristics of each one like the education and competitiveness [105], [106]. Therefore, work-related stress, which refers to the stress that has been caused by work, can be understood as a particular example of stress. It follows the same characteristics as general stress and its response patterns, and effects can be evidenced and measured in the same way.

When work-related stress arises and it is not treated, it can cause big long-term physical and mental problems on the worker [104], but also economic losses in the companies. Musculoskeletal disorders, depression, anxiety, increased probability of infections [107], chronic fatigue syndrome, digestive problems, diabetes, osteoporosis, stomach ulcers, and coronary heart disease are only some examples of chronic stress' long-term consequences. These health problems bring consequences to enterprises, where absenteeism, staff turnover, and tardiness increase, decreasing production. The problem of presenteeism also arises, where employees attend their workplace, but they don't work at 100% of their capabilities. The cost of work-related stress, anxiety, and depression in Europe are estimated by the European Commission to be €617 billion per year. Also, stress in the United States has an annual cost of US$200 billion [6]. According to the Fourth European Working Conditions Survey (EWCS), work-related stress was reported by 22% of workers from 27 Member states of the European Union [7]. Moreover, higher prevalence of stress has been reported in North America, where 55% of the population has reported increased workload having a significant impact on physical and mental health as described in APA Survey [8].

2.4 Data mining

Data mining, in its most fundamental form, is to extract interesting, nontrivial, implicit, previously unknown and potentially useful information from data [108]. Data mining is also the process of searching for knowledge in data from different perspectives [109]. Data mining lies at the intersection of computer science, optimization, and statistics, and often appears in other disciplines. Here knowledge can refer to any kinds of summarized or unknown information that is hidden underlying the raw data. For instance, it can be a set of discriminative rules generated from the data collected on some patients of a certain disease and healthy people. These rules can be used for predicting the disease status of new patients.

Many people treat data mining as a synonym of another popular concept, knowledge discovery from data (KDD), while others view data mining as merely an essential step in the process of knowledge discovery [108]. The knowledge discovery process is an iterative sequence of the following steps:

1. Data cleaning, to remove noise and inconsistent data.
2. Data integration, where multiple data sources may be combined.
3. Data selection, where data relevant to the analysis task are retrieved from the database.
4. Data transformation, where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations.
5. Data mining, an essential process where intelligent methods are applied to extract data patterns.
6. Pattern evaluation, to identify the truly interesting patterns representing knowledge based on interestingness measures.
7. Knowledge presentation, where visualization and knowledge representation techniques are used to present mined knowledge to users.

In general, data mining tasks can be classified into two categories: descriptive and predictive. Descriptive data mining tasks characterize a target data set in concise, informative, discriminative forms. Predictive mining tasks conduct the induction and inference of the current data to make future predictions [110].

Data mining is the process of discovering interesting patterns and knowledge from large amounts of data. The data sources can include databases, data warehouses, the web, other information repositories, or data that are streamed into the system dynamically.

2.5 Machine learning

Machine learning has been defined as the field of study that gives computers the ability to learn without being explicitly programmed [111]. Machine learning is a data analysis technique that teaches computers to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of instances available for learning increases.

A more engineering-oriented definition of machine learning is the following: a computer problem is said to learn from experience *E* with respect to some task *T* and some performance measure *P*, if its performance on *T*, as measured by *P*, improves with experience *E* [112]. For example, a diagnostic system is a machine learning program that can learn to identify failures of a product from a training dataset involving examples of failures. Each training example is also called a training instance. In this case, the task *T* is to identify failures of the product, the experience *E* is the training data, and the performance measure *P* needs to be defined. This performance measure is called accuracy and is often used in classification tasks. A diagnostic system based on machine learning techniques automatically learns which attributes are good predictors of product failure by detecting failure patterns in the training dataset. Figure 2.4 shows a high-level overview of the machine learning approach to diagnosis.

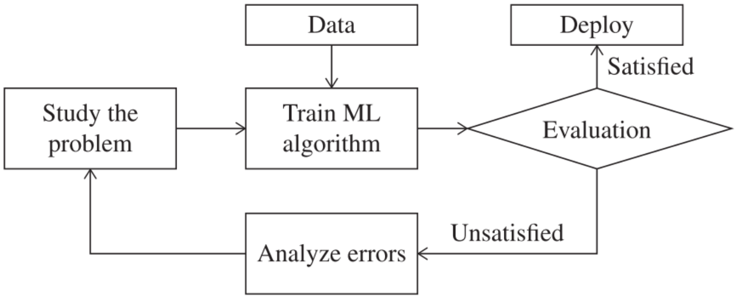


Figure 2.4 A high-level overview of the machine learning approach to diagnosis

Machine learning algorithms can be divided into the following four categories depending on the amount and type of supervision they need while training: supervised, unsupervised, semi-supervised, and reinforcement learning [112]. In supervised learning, the training data fed to the machine learning algorithms include the desired solutions, called labels or classes. Classification is a typical supervised learning task. A diagnostic system is a good example of classification: it is trained with many attributes or features along with their class, e.g. faulty or healthy, and it must learn how to classify new variables or features.

Another typical task is to predict a target numeric value, such as remaining of product useful life, given a set of features called predictors. This sort of task is called regression. To train the machine learning algorithms, the training dataset must contain predictors with associated labels.

Note that some regression algorithms can be used for classification and vice versa. For example, logistic regression is commonly used for classification, as it can output a value that corresponds to the probability of belonging to a given class. The supervised machine learning algorithms widely used for both classification and regression involve *k*-Nearest Neighbor (*k*-NN), naïve Bayes classifiers, support vector machines (SVMs), neural networks, decision trees, random forests, linear regression, and logistic regression.

Unlike supervised learning, in supervised learning the training dataset is unlabeled. The major tasks using unsupervised learning are clustering, e.g. *k*-means, fuzzy *c*-means, hierarchical cluster analysis, and self-organizing map; and dimensionality reduction, e.g. principal component analysis, locally linear embedding, and *t*-distributed stochastic neighbor embedding. In fact, clustering has been widely used for anomaly detection (also outlier detection) under the assumption that the majority of the instances in the dataset are normal and facilitate the detection of anomalies in an unlabeled test data by looking for instances that seem to fit least to the remainder of the dataset. Dimensionality reduction is primarily used for simplifying the data without losing too much information. In fact, it is often a good idea to reduce the dimensions of the dataset before it is fed to machine learning algorithms, e.g. supervised ML algorithms for classification. This is mainly because the dataset will run much faster, the data will take up less disk and memory space, and in some cases, it may also perform better. Likewise, unsupervised learning has been used to output two-dimensional or three-dimensional representation of the high-dimensional data that can be easily plotted.

Semi-supervised learning is a class of supervised learning tasks and techniques that make use of unlabeled data for training; the training dataset involves a lot of unlabeled data and a little bit of labeled data.

Reinforcement learning is the task of getting an agent that can observe the environment, to select and perform actions, and obtain rewards in return, or penalties in the form of negative rewards). The agent then learns by itself what is the best strategy, called a policy, to get the most rewards over time; the policy defines what action the agent should choose when it is in a given situation.

2.6 Waikato Environment for Knowledge Analysis (WEKA)

Weka [113]. Also known as Waikato Environment is a machine learning software developed at the University of Waikato in New Zealand. It is best suited for data analysis and predictive modeling. It contains algorithms and visualization tools that support machine learning. Weka has a GUI that facilitates easy access to all its features. It is written in Java programming language. Weka supports major data mining tasks including data mining, processing, visualization, regression etc. It works on the assumption that data is available in the form of a flat file. Weka can provide access to SQL Databases through database connectivity and can further process the data/results returned by the query.

Classification

Classification is a data mining technique capable of identifying the category to which a new instance belongs, based on a previous set of instances categorized in classes. The classification algorithm makes a generalization of the known instances to predict the category of the new one. The classification allows us to extract information about the data and expresses it through a model. WEKA has several classification algorithms. Some of them are described below.

- ***Decision Stump.*** It is a single-level decision tree type classification algorithm, which performs classification based on a single attribute. The resulting tree contains only one attribute with n outputs that represent each of the classes.
- ***OneR.*** It is a classification algorithm that generates a rule for each attribute. Subsequently, select the rule with the minimum error rate to perform the classification.
- ***NaiveBayes.*** It is a classification algorithm based on Bayes' theorem, which constructs models that predict the probability of possible results.
- ***C4.5 (J48).*** It is a classification algorithm that generates a decision tree from the data through partitions performed recursively. It has the option of generating pruned and unpruned trees, that is, it has the possibility of generating more general models. The generated tree contains nodes and in each of them a criterion must be met to move to the next node. In this way, the tree is traversed until it reaches a leaf, which represents the class it predicts.

WEKA allows the application of several types of tests. The types of tests used in this investigation are described below.

1. ***Use training set.*** This type of test is performed using the same set in which the predictive model is constructed. When performing this type of test, the result can not be considered as a generalization, since the training and tests are carried out in all cases and, therefore, the classification results almost without errors. It is only used to make comparisons of classification algorithms.
2. ***Cross-validation.*** This type of test is done by dividing the total of the instances in folders (folds), according to the number of folders that you specify. For the construction of the model, the instances of a folder are considered test data and the rest, training data. This is repeated with each of the folders. The calculated errors will be the average of all the executions. The tests were performed with 2 and 10 folders.
3. ***Percentage Split.*** This type of test is similar to cross-validation. It is done by dividing the data into two groups, according to the specified percentage (%). The indicated value is the percentage of training instances to build the model, and the rest is used as test data.

WEKA shows us different metrics in the execution of each of the classifiers. The metrics that were considered in this work are the following:

1. **Percentage** of correctly classified instances.
2. ***Kappa.*** It is a metric that measures the concordance of the class predicted with the correct class. The value of 1 means complete concordance.
3. ***Precision.*** It is the proportion of the number of instances of class x, among the number of instances that were classified as class x. The value of 1 means greater precision.

4. ***Recall.*** Is the proportion of the number of instances that are classified as class x, among the number of instances of class x. The value of 1 means greater integrity.

In addition, it shows us the next block of information:

- ***Summary*:** It includes several measurements calculated as the percentage of instances correctly classified, that is, the precision; the Kappa statistic that measures what the prediction fits the real class, 1.0 means total adjustment; the mean error the mean square error, the relative error and the relative quadratic error.
- ***Detailed Accuracy by Class*:** includes the accuracy parameters of each class
  - ***True Positive (TP) Rate*** is the proportion of examples that were classified as class x, among all the examples that really have class x, that is, how much of the class has been captured. In the confusion matrix, it is the value of the diagonal element divided by the sum of the corresponding row.
  - ***False Positive (FP) Rate*** is the proportion of examples that were classified as class x, but actually belong to another class, among all the examples that do not have class x. In the confusion matrix, it is the sum of the column minus the value of the diagonal element divided by the sum of the rows of the other classes.
  - ***Precision*** is the proportion of examples that obtain the class x and that really belong to that class, among all those that were classified as class x, although in reality they did not belong to that class. In the matrix is the element of the diagonal divided by the sum of the relevant column.
  - ***F-Measure*** is a combined measure of *Precision* and *Recall.*

Confusion Matrix. Also called contingency table, is formed by as many rows and columns as there are classes. The number of instances correctly classified is the sum of the diagonal of the matrix and the rest are classified incorrectly.

Attribute selection

The selection of relevant attributes allows us to explore which subsets of attributes are those that can best categorize the instance class. The objective is to select the smallest subset of attributes that have the most weight in determining whether the instance belongs to one class or another, so that the classification percentage is not significantly affected and the resulting distribution is closest to the original.

The intention is to eliminate redundant, irrelevant or noisy data and avoid slow processes due to an insignificant amount of information. As a result, there is an improvement in predictive performance. The training and processing time and storage needs are reduced. At the same time, you get a better visualization and understanding of the data.

WEKA offers several techniques for the selection of attributes. In the attribute selection tab, the selection is made through the configuration of two components: the evaluation method and the search method.

- **Attribute Evaluator:** is the function that determines the quality of the set of attributes to discriminate the class. There are two types of evaluation methods:
    - ***Wrapper.*** This evaluation method directly uses a specific classifier to measure the quality of the subset of attributes, through the error rate of the classifier. The method involves the classifier to explore the best selection of attributes that optimizes its performance. They need a complete process of training and evaluation in each case of search, and as a consequence, it is very expensive.
    - ***CfsSubsetEval.*** This evaluation method does not use a specific classifier. The procedure that follows is to calculate the correlation of the class with each attribute and eliminate attributes that have a very high correlation as redundant attributes.
- **Search Method:** is the way to perform the search for sets. If you want to carry out the exhaustive evaluation of all the subsets, an incomparable combinatorial problem appears as soon as the number of attributes grows. To do so, these strategies appear that allow searching in a more efficient way:
    - ***ForwardSelection.*** This search method is characterized by its speed. It is a sub-optimal search method in climbing. The procedure is as follows: the best attribute is chosen first, then it adds the following attribute that contributes the most and continues this way until it reaches the situation in which adding a new attribute worsens the situation.
    - ***BestSearch.*** This search method allows us to search for interactions between more complex attributes. Its procedure is to analyze what improves and worsens a group of attributes when adding elements, with the possibility of making retracements to explore in more detail.
    - ***ExhaustiveSearch.*** This search method lists all the possibilities and evaluates them to select the best.

Chapter 3. Related Work

Stress has been studied from different viewpoints for many years. The traditional methods to detect it are questionnaires or surveys, used mostly by psychology, and physiological sensors used mostly by medical approaches.

Psychology researchers have developed and tested several questionnaires for stress recognition such as the Perceived Stress Scale [37], the Job Content Questionnaire [38], the Workplace Stress Survey [43], among others. These questionnaires have some benefits. They are inexpensive, do not represent too much effort from the researcher, and the data are easy to compile by predefined answers. Also, they are very practical because they could be applied by the researcher or by other people even remotely without affecting validity or reliability. However, questionnaires have some disadvantages. The subjects' answers are highly subjective, and they could unconsciously hide information. Also, the questionnaires are defined by a group of researchers who could let pass an important issue to the subject and not include questions about it.

Medical researchers have utilized stationary physiological sensors to recognize changes in the body in the presence of stress. A stationary sensor refers that the subject cannot move without restrictions and cannot wear the sensor, on the contrary of new wearable sensors described later. Stress have been recognized through the monitoring of physiological features such as: heart rate variability [101], [114]; body temperature [115]; skin galvanic response [116] also called skin conductivity; respiration rate [117], [118]; ambulatory activity respiration [119]; subcutaneous cortisol [120]; electro-encephalic activity [54]; among others. Physiological sensors are very precise and their validity is nowadays acknowledged [11], [50]. However, these sensors involve wires and other hardware that may limit movement. Consequently, subjects feel uncomfortable and they could feel stressed by the monitoring itself, and this could affect the data collected.

Novel methods have appeared thanks to the constant development of new technologies. These novel methods allow continuous monitoring in a non-intrusive manner, limiting any impact on subjects' routines and consequently their typical behavior. Several works have been analyzed and are presented in in-depth detail in the following subsections.

The criterion used for this analysis was based on which methods authors have used, how was collected the data, which techniques were utilized, and the results obtained.

3.1 Stress recognition by computer method

Computer method provides an excellent opportunity for stress recognition in domains in which people interact with a computer such as laboratories, workplaces or academia. The advantages offered by computer features are that they have a very low cost since they are generally based on existing and inexpensive hardware; and the diversity of features that can be extracted which may include physical, behavioral and physiological measures, considering minor hardware modifications.

Sometimes, people have expressed concerns when monitoring computer features because monitoring systems could register all that is done with the mouse and the keyboard. In order to avoid privacy concerns, when considering the keyboard, monitored features should focus on how is written rather than what is written. When considering the mouse, features should focus on how people click or move the mouse rather than where people clicked or moved to. Similarly, when considering the applications, features should focus on time spent on a predefined application category rather than which applications were used.

There are different types of features that can be obtained from the computer, for example, features extracted from the peripherals keyboard and mouse. Carneiro et al. have carried out several studies based on these data. They proposed a system for mental fatigue detection by monitoring an individual's use of the mouse and the keyboard [121]. Data were collected from 20 subjects in two moments for each user: at the beginning (without fatigue) and at the end of the day (with expected mental fatigue). The data collected in the first phase was compared with the second phase. The Mann-Whitney test was used to perform the analysis. Results showed that there is a significant difference in the features Keydown Time, Errors per Key Pressed, Time Between Keys, Mouse Velocity and Mouse Acceleration. In another study, they built a model by training *k*-Nearest Neighbor (*k*-NN) algorithm [122]. Data was collected from 27 subjects in the wild for over a month. They obtained an accuracy of 96.85%. Also, they recognized stress relying on the observation of the individual's interaction with the computer such as the keyboard, and the mouse data [123]. They focused on desk jobs and others similar, in which people spend long hours interacting with the computer. They collected data from 24 subjects in the wild. They trained a neural network that takes as input features that describe the user's interaction patterns and provides as output an estimation of the user's mental fatigue. The main result is that the trained neural network correctly classified 81% of the instances. Furthermore, they measured the levels of stress in humans by analyzing their behavioral patterns when interacting with technological devices such as smartphones, tablets, computers, and video cameras [124]. The features gathered were: touch pattern, touch accuracy, touch intensity, and touch duration from the touchscreens; the amount of movement from the video camera; and acceleration from the smartphone. Data was collected from 19 subjects. They used a J48 tree to classify touches as stressed or not achieving an accuracy of 78%.

Others researchers have also used features from keyboard and mouse. Ulinskas et al. [88] measured the fatigue using keystroke dynamics data. They used the RHU Keystroke dataset of keystroke data collected by El-Abed et al. [125]. The dataset consists of keystroke data from 53 subjects have participated in the acquisition process by typing the password “rhu.university” 15 times (3 sessions, 5 trials each) with 3 to 30 days separating each session. They used the *k*-Nearest Neighbor classifier and achieved an accuracy of 91% using key release-release data. Khan et al. [126] studied the relationship between the mood of computer users and their use of keyboard and mouse to create a generic or individualized mood measure. They carried out a field study with 26 subjects and a controlled study with 16 subjects. In the field study, interaction data and self-reported mood measurements were collected during normal computer use over several days. In the controlled study, subjects worked on a programming task while listening to high or low arousing background music. Besides subjective mood measurement, galvanic skin response data was also collected. They calculated the Pearson correlation and Fisher z score for the analysis. Results found no generic relationship between the interaction data and the mood data. However, the results of the studies found significant average correlations between mood measurement and personalized regression models based on keyboard and mouse interaction data. In another study, Vizer [127] proposed stress recognition by monitoring everyday keyboard interactions. Data was collected from 9 subjects in a laboratory condition. Keystroke and linguistic features were extracted from the samples and models were constructed for each subject. Correct classification rates ranged from 62% to 88% with a mean of 72%.

The keyboard and the mouse are one of the most common channels of communications for computer users. The users are in continuous contact with them. Some researchers have leveraged this situation and have adapted pressure sensors to the peripherals for stress recognition. For example, Hernandez et al. [116] studied stress recognition using a pressure-sensitive keyboard and a capacitive mouse. Data was collected from 24 subjects in laboratory conditions. The subjects performed several computerized tasks consisting of expressive writing, text transcription, and mouse clicking. The results showed significantly increased typing pressure (>79% of the subjects) and more contact with the surface of the mouse (75% of the subjects) during stressful conditions.

There have been other researchers who have recognized stress relying on several computer data sources. For example, Liao et al. [128] developed a real-time non-invasive system which is capable of monitoring physical appearance such as: facial expression, eye movements, and head movements extracted from video via visual sensors; physiological conditions collected from an emotional mouse; and behavioral data from user interaction activities with the computer. Data was collected from 5 subjects. They built a model using a Dynamic Bayesian Network. They calculated the correlation between inferred stress levels and ground-truth stress. They achieved an average of 0.86 using all features. Also, Huang et al. [129] analyzed the pattern of human gaze behaviors surrounding a mouse-click event using only data collected from the common computer webcam and mouse. Their experiments included 20 subjects. They achieved an F-measure of 0.74 using user-dependent model, and an F-measure of 0.79 using the user-independent model by means of the random forest algorithm. Koldijk et al. [89] developed automatic classifiers to infer stress from a multimodal set of sensor data: computer interactions, facial expressions, body postures, and physiology. Their experiments included 25 subjects. They recognized neutral and stressful working conditions with an accuracy of 90% by means of SVM.

Stress has also been recognized by monitoring users' behavior on their computers. Karunaratne et al. [130], [131] studied if the relationship between the stress and the human-computer interaction parameters is non-linear. They identified the best computer features. They collected computer data such as: logging in to social networking sites, making typing errors, checking emails, scroll the window up and down, and switching between tasks. They performed correlation analysis and multiple linear regression analysis. They found that Multiple Linear Regression is not suitable to predict the stress. They concluded that the relationship between the stress and the human-computer interaction parameters is non-linear, and that is better to use machine learning techniques for stress recognition.

3.2 Stress recognition by smartphone method

Nowadays, smartphones are part of our everyday lives and we always carry them with us. Smartphones provide unobtrusive and cost-efficient access to previously inaccessible sources of data related to daily social behavior [19]. Their low cost and their availability make them easily reach a significant number of users.

Smartphones are able to sense a wealth of behavioral data through their built-in sensors like accelerometers. For example, Garcia-Ceja et al. [73] used data only from the smartphone's built-in accelerometer to detect behavior that correlates with subjects stress levels. Their experiment included 30 subjects and they used a Naive Bayes and Decision Trees to classify self-reported stress levels achieving a maximum overall accuracy of 71% for user-specific models and an accuracy of 60% for the use of similar-user models. Maxhuni et al [132] proposed the use of intermediate models for stress recognition using accelerometer data from smartphones. The intermediate models represent the mood state of a person which is used to build the final stress prediction model by means of the Self-Training algorithm. Data was collected from 29 subjects from two different organizations for over 5 weeks. They obtained an accuracy of 78.2% to classify stress levels.

Smartphones were designed to communicate with other people in different ways. These data were also used by researchers for stress recognition. Bogomolov et al. [19] have recognized daily stress based on human behavior metrics derived from smartphone activity such as call log, SMS log, and Bluetooth interactions. They collected data from 117 subjects for more than eight weeks. The machine algorithms used were Random Forest and Gradient Boosted. They obtained an accuracy of 72.39% and a Cohen's k metrics of 0.37 for a 2-class problem. In another study, Bogomolov et al. [70] recognized daily stress as a two-class classification problem (stressed and not stressed) based on information concerning people activities, as detected through their smartphones; but also they considered weather conditions and personality traits. People activities were represented by features extracted from calls, SMS logs, and Bluetooth interactions. Data was collected from 117 subjects living in a married graduate student residency of a major US university. They built models using Random Forest, Generalized Boosted Model, Support Vector Machines, and Neural Networks algorithms. Their models have achieved 90.68% to 92.52% accuracy on the training set and 72.51% to 84.86% accuracy on the test set.

Ferdous et al. [133] used verbal interventions captured by smartphones. They studied the correlation between perceived stress levels and verbal interactions. They monitored 28 subjects over 6 weeks in the wild. Results of the Pearson product-moment correlation coefficient showed that 60% of subjects obtained a positive correlation between perceived stress levels and verbal interaction, while this correlation is observed for over 90% of highly stressed subjects.

Some researchers considered a combination of different type of data obtained from smartphones. Hernandez-Leal et al. [134] modeled stress behavior based on physical activity level, location, social interaction, and social activeness. They proposed a transfer learning technique for building a model of a subject with scarce data. It is based on the comparison of decision trees to select the closest subject for knowledge transfer. Their experiment included 30 subjects within two organizations. They used decision tree algorithm and achieved an accuracy of up to 61.24%. In another study of the same authors [10], they proposed a combination of the techniques semi-supervised learning, ensemble methods and transfer learning to build a stress model of a subject with scarce data. They obtained data from 30 subjects from two organizations in the wild. They found that using information from similar subjects can improve the accuracy of the subjects with scarce data. They obtained an increased accuracy of ≈ 10% to 71.58% compared to not using any transfer learning technique. Sysoev et al. [76] proposed stress recognition by analyzing behavioral and contextual data received from a smartphone such as audio, gyroscope and accelerometer features, light condition, screen mode (on/off), current stress level self-assessment, and the current activity type. Data was gathered in real-life situations. They built three models: two models considering the current activities of the subject and the other one without. They trained a collection of algorithms for machine learning and achieved an average accuracy of 73% using only behavioral and contextual data, and an accuracy of 75.81% using the standing activity.

Other features that can be obtained from smartphones are those referred to the users' apps usage. Ferdous et al. [72], investigated the relationship between the pattern of mobile apps usage of the users at work and their perceived stress levels. They monitored the duration spent on the apps by the subjects. The apps considered as used were those which screen statuses were on. Data was collected from 28 subjects over a period of 6 weeks. Authors built user-centric models by training an SVM classifier with individual app usage behavior. They achieved an average accuracy of 75% and a precision of 85.7%. Vildjiounaite et al. [71] proposed an unsupervised method for detecting stress on the basis of mobile phone usage data of several application categories like communication, infotainment, entertainment, well-being, etc., and also, they collected location data. They developed a stress detector using discrete hidden Markov models. The detector requires neither additional hardware nor data labeling. Data was collected from 30 subjects in real life. They achieved an accuracy from 60 to 70% in their experiments.

Finally, there are other behavioral features that referrers to the interaction of the users with the smartphone. Ciman et al. [74], [75] proposed stress recognition based on two different approaches. The first one was based on smartphone gestures analysis, for example, tap, scroll, swipe and text writing. It was evaluated in laboratory conditions with 13 subjects. They obtained an F-measure from 79 to 85% with within-subject model and an F-measure from 70 to 80% with global model. The second one was based on smartphone usage analysis. It was tested in-the-wild with 25 subjects. They obtained an F-measure from 77 to 88% with within-subject model and an F-measure from 63 to 83% with global model. They used Decision Tree, k-Nearest Neighbor, Support Vector Machine and Neural Network algorithms for classification.

3.3 Stress recognition by wearable method

Wearable method is one of the latest trends on stress recognition. In the last years, there was a major development in wearable devices used for acquiring physiological signs. They present great advantages because they can be worn as a regular fashion accessory or clothing.

Wearable devices have also been used for stress recognition. For example, Gjoreski et al. [135] developed a smartphone application capable of monitoring physical activity and mental stress using data provided by a commercial wrist device equipped with standard bio-sensors and an accelerometer. The stress detection module was trained with the SVM algorithm using data from 21 subjects in a laboratory setting and tested on 5 subjects in a real-life setting. They achieved an accuracy of 92% for detection of stressful events. Condori-Fernandez et al. [136] proposed an arousal-based statistical approach for detecting stress in real-time. They used the E4-wristband for gathering electrodermal activity (EDA). Data was collected from 12 subjects in laboratory settings. They obtained an accuracy of 79.17%, a precision of 60%, and a recall of 50%. Egilmez et al. [137], focused on analyzing the effect of replacing other body sensing platforms with their wrist-worn equivalent on stress prediction accuracy. Data was collected from 9 subjects with multiple body sensors. They were also asked to wear a commercially available Android smartwatch equipped with a Galvanic Skin Response sensor, a chest-based heart rate sensor, and a finger-based commercial GSR sensor. Authors developed 2 models: an intended-stress model, and a self-reported stress model. They used several classifiers for building the models. The results for the intended-stress model were an accuracy of 59% using Naïve Bayes and Random Forest algorithms. The results for self-reported stress model were an accuracy of 78% using the Random Forest algorithm. Han et al. [138], proposed detecting work-related stress based on electrocardiogram and respiration signals measured by a wearable device. They detected three levels of stress: no stress, moderate stress, and high perceived stress. Data was collected from 39 subjects. They built a model using Random Forest and Support Vector Machine algorithms. They improved the accuracy from 78% to 84% in three states classification. And in binary stress detection, the accuracy is 94%.

Some researchers have combined data obtained from wearables and from another source. For example, Cvetkovic et al. [139], [140] developed a system for the management of physical, mental and environmental stress. The system collects data with wearable and environmental sensors interprets it and provides recommendations to the user through a smartphone application. Their experiments included 21 subjects with a standardized stress-inducing experiment. They trained a model using the Random Forest algorithm. The mental stress module achieved an accuracy of 92% with context and an accuracy of 76% with no context. In another study [141], the same authors developed an algorithm for real-time activity recognition which can utilize acceleration and physiological data from the wristband, smartphone or both devices. They developed an algorithm for mental stress detection that utilizes heart rate, heart rate variability and galvanic skin response data from the wristband, as well as the outputs of the physical activity monitoring. The physiological sensor data were fed into a laboratory stress detector. Sano et al. [117] focused their analysis on finding physiological or behavioral markers for stress. They monitored 18 subjects over 5 days. They used a wrist sensor to collect accelerometer and skin conductance data, and a smartphone to collect call and SMS log, location and screen on/off. They found that activity level, SMS and screen on/off are statistically significant features associated with stress using a correlation analysis. They used machine learning to classify whether the subjects were stressed or not. They obtained an accuracy of more than 75%.

3.4 Stress recognition by other methods

Stress has been recognized by using other methods. For example, Giannakakis et al. [142] developed a framework for the detection and analysis of stress/anxiety emotional states through video-recorded facial cues. Features under investigation included eye-related events, mouth activity, head motion parameters and heart rate estimated through camera-based photoplethysmography. The classification methods that were used and tested are *k*-Nearest Neighbor, Generalized Likelihood Ratio, Support Vector Machines, Naïve Bayes classifier, and AdaBoost classifier. In each experiment phase, they achieved an accuracy between 80% and 90%. The best classification accuracy was presented in the social exposure phase using Adaboost classifier achieving an accuracy of 91.68%.

In another study, Cho et al. [143], [144] proposed DeepBreath a deep learning model which automatically recognizes people's psychological stress level from their breathing patterns. They tracked a person's breathing patterns as temperature changes around his/her nostril using a low-cost thermal camera. They used a deep Convolutional Neural Network. They achieved an accuracy of 84.59% in discriminating between two levels of stress, and an accuracy of 56.52% in discriminating between three levels.

Finally, Lefter et al. [145] proposed a new method for automatic stress prediction based on a decomposition of stress into a set of intermediate level attributes. They investigated how speech and gestures convey stress, and how they can be used for automatic stress recognition. The analyzed features were extracted from the intended semantic message such as spoken words for speech, symbolic meaning for gestures; and stress conveyed by the modulation of either speech and gestures such as intonation for speech, speed, and rhythm for gestures. They found that speech modulation is the best performing intermediate level attribute for automatic stress prediction. They achieved an accuracy of 66% using all data, and an accuracy of 67% using only gesture data.

3.5 Summary

In this chapter, several works for stress recognition were described. A comparison of all related works is presented in Table 3-1. Each column describes the analysis criterion.

Table 3-1 Comparison of related works

Work	Method	Data collection	Techniques used	Results
Carneiro et al. [121]	Computer	20 subjects in two moments for each user.	Mann-Whitney test.	A significant difference in the features Key down Time, Errors per Key Pressed, Time Between Keys, Mouse Velocity and Mouse Acceleration.
Carneiro et al. [122]	Computer	27 subjects in the wild for over a month.	*k*-Nearest Neighbor algorithm.	An accuracy of 96.85%.
Carneiro et al. [123]	Computer	24 subjects in the wild.	Neural network algorithm.	An accuracy of 81%.

Work	Method	Data collection	Techniques used	Results
Carneiro et al. [124]	Computer and smartphone	19 subjects.	J48 algorithm.	An accuracy of 78%.
Ulinskas et al. [88]	Computer	53 subjects.	*k*-Nearest Neighbor algorithm.	An accuracy of 91%.
Khan et al. [126]	Computer	A field study with 26 subjects and a controlled study with 16 subjects.	Pearson correlation, and Fisher's z score.	Significant average correlations between mood measurement and personalized regression models based on keyboard and mouse interaction data.
Vizer [127]	Computer	9 subjects in a laboratory condition.	Logistic regression algorithm.	An accuracy from 62% to 88% with a mean of 72%.
Hernandez et al. [116]	Computer	24 subjects in laboratory conditions.	Wilcoxon Rank Sum test and Kruskal-Wallis test.	Significantly increased typing pressure (>79% of the subjects) and more contact with the surface of the mouse (75% of the subjects) during stressful conditions.
Liao et al. [128]	Computer	5 subjects.	Dynamic Bayesian Network.	Correlation average of 0.86 using all features.
Huang et al. [129]	Computer	20 subjects.	Random Forest algorithm	An F-measure of 0.74 using user-dependent model, and an F-measure of 0.79 using user-independent model.
Koldijk et al. [89]	Computer	25 subjects.	SVM	An accuracy of 90%.
Karunaratne et al. [130], [131]	Computer	Questionnaire survey among 769 subjects.	Multiple Linear Regression	The relationship between the stress and the human-computer interaction parameters is non-linear, and that is better to use machine learning techniques for stress recognition.
Garcia-Ceja et al. [73]	Smartphone	30 subjects.	Naive Bayes and Decision Trees algorithms.	An accuracy of 71% for user-specific models and an accuracy of 60% for the use of similar-user models.
Maxhuni et al [132]	Smartphone	29 subjects from two different organizations for over 5 weeks.	Self-Training algorithm.	An accuracy of 78.2%.
Bogomolov et al. [19]	Smartphone	117 subjects for more than eight weeks.	Random Forest and Gradient Boosted algorithms.	An accuracy of 72.39% and a Cohen's k metrics of 0.37 for a 2-class problem.
Bogomolov et al. [70]	Smartphone, weather conditions, and personality traits.	117 subjects living in a married graduate student residency of a major US university.	Random Forest, Generalized Boosted Model, Support Vector Machines, and Neural Networks algorithms.	An accuracy from 90.68% to 92.52% on the training set, and from 72.51% to 84.86% on the test set.

Work	Method	Data collection	Techniques used	Results
Ferdous et al. [133]	Smartphone	28 subjects over 6 weeks in the wild.	Pearson correlation.	60% of subjects obtained a positive correlation between perceived stress levels and verbal interaction, while this correlation is observed for over 90% of highly stressed subjects.
Hernandez-Leal et al. [134]	Smartphone	30 subjects within two organizations in the wild.	Decision tree algorithm.	An accuracy up to 61.24%.
Maxhuni et al [10]	Smartphone	30 subjects from two organizations in the wild.	Decision tree algorithm.	An increased accuracy of ≈ 10% to 71.58% compared to not using any transfer learning technique.
Sysoev et al. [76]	Smartphone	In real-life situations.	A collection of algorithms for machine learning.	An average accuracy of 73% using only behavioral and contextual data, and an accuracy of 75.81% using the standing activity
Ferdous et al. [72]	Smartphone	28 subjects over a period of 6 weeks.	SVM algorithm.	An average accuracy of 75% and a precision of 85.7%.
Vildjiounaite et al. [71]	Smartphone	30 subjects in real life.	Hidden Markov Models.	An accuracy from 60 to 70%.
Ciman et al. [74], [75]	Smartphone	First data collection:13 subjects in laboratory conditions. Second data collection: 25 subjects in the wild.	Decision Tree, *k*-Nearest Neighbor, Support Vector Machine and Neural Network algorithms.	An F-measure from 79 to 85% with within-subject model and an F-measure from 70 to 80% with the global model on smartphone gestures analysis. An F-measure from 77 to 88% with within-subject model and an F-measure from 63 to 83% with the global model on smartphone usage analysis.
Gjoreski et al. [135]	Wearable	21 subjects in a laboratory setting and tested on 5 subjects in a real-life setting.	SVM algorithm	An accuracy of 92% for detection of stressful events.
Condori-Fernandez et al. [136]	Wearable	12 subjects in laboratory settings.	ADaptive WINdowing method.	An accuracy of 79.17%, a precision of 60%, and a recall of 50%.
Egilmez et al. [137]	Wearable	9 subjects.	Naïve Bayes and Random Forest algorithms.	An accuracy of 59% using Naïve Bayes and Random Forest algorithms for the intended-stress model. An accuracy of 78% using Random Forest algorithm for self-reported stress model.
Han et al. [138]	Wearable	39 subjects.	Random Forest and Support Vector Machine algorithms.	An improved accuracy from 78% to 84% in three states classification. And in binary stress detection, the accuracy is 94%.

Work	Method	Data collection	Techniques used	Results
Cvetkovic et al. [139], [140]	Wearable and environmental sensors	21 subjects.	Random Forest algorithm.	An accuracy of 92% with context, and an accuracy of 76% with no context.
Sano et al. [117]	Wearable and smartphone	18 subjects over 5 days.	Correlation analysis.	Statistically significant features: activity level, SMS and screen on/off.
Giannakakis et al. [142]	Facial expression	23 subjects.	*k*-Nearest Neighbor, Generalized Likelihood Ratio, Support Vector Machines, Naïve Bayes and AdaBoost algorithms.	An accuracy of 91.68%.
Cho et al. [143], [144]	Thermal camera.	-	Convolutional Neural Network.	An accuracy of 84.59% in discriminating between two levels of stress, and an accuracy of 56.52% in discriminating between three levels.
Lefter et al. [145]	Speech and gestures.	-	Bayes Net (BN) algorithm.	An accuracy of 66% using all data, and an accuracy of 67% using only gesture data.

Chapter 4. Methodology for the development and evaluation of predictive models

4.1 Overview of our proposal

This chapter presents the proposed methodology for the development and evaluation of predictive models. The methodology is described at three levels of abstraction which goes from general to specific: phase, general task and specific task. At the top level, the methodology is organized into three phases: data collection, data processing, and predictive models' development (see Figure 4.1). The second level consists of a number of general tasks which must be done. The third level describes how generic tasks should be carried out.

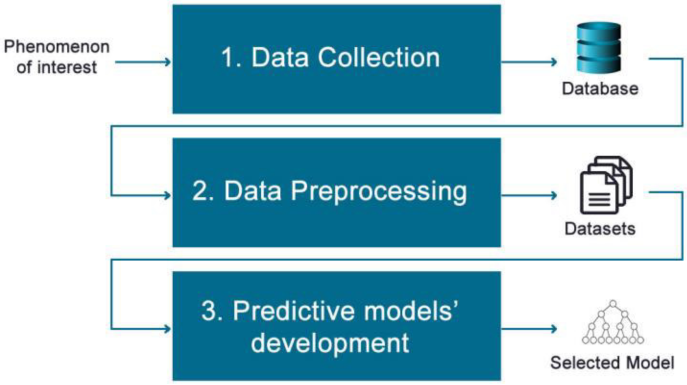


Figure 4.1 Methodology for development and evaluation of predictive models

4.2 Phase 1. Data collection

Phase 1 of the methodology is data collection. Data collection is the process of gathering data, information or any attributes of interest in order to analyze them and make conclusions. The input of this phase is the phenomenon of interest and the output is the data collected. Phase 1 consists of four general tasks which are depicted in Figure 4.2.

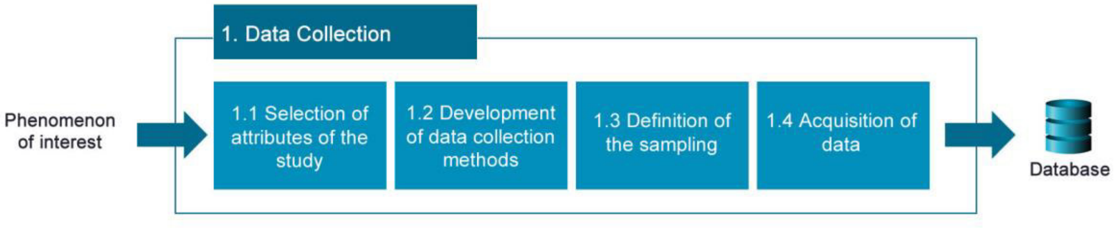

1. Data Collection
Phenomenon of interest
1.1 Selection of attributes of the study
1.2 Development of data collection methods
1.3 Definition of the sampling
1.4 Acquisition of data
Database

Figure 4.2 General tasks of phase 1

4.2.1 Selection of attributes of the study

The objective of this general task is to select the attributes of the study that will be collected in this first phase. The selection of the attributes should be done after a thorough understanding of the phenomenon of interest through a study of the state of the art. The specific tasks are the following.

a) Analysis of the nature of the phenomenon

Search and read about the phenomenon in books and scientific journals. Recognize the main characteristics of the phenomenon, its causes, and implications. Identify which of the characteristics could be measurable in some way, which characteristics provide discriminant information about the phenomenon. Search who are affected by the phenomenon and classify them. List their common characteristics and identify the possible discriminant attributes. Search if there is a classification of the phenomenon and understand each type. Consider that these attributes must capture the behavior of a person or an object in the presence and in the absence of the phenomenon of study in order to develop the predictive model. List all the identified attributes.

b) Study of state of the art

Search related works in books and scientific journals. Identify which attributes have been using to solve the problem or a similar problem. Identify how is the data collected, if it is collected in a manual or automatic manner, and what instruments have been used. Expand your research to other areas of knowledge. Maybe a problem has been studied from different angles and with different instruments. List all the identified attributes.

c) Create new attributes

Think which attributes have not been studied to solve the problem. Consider other attributes that could be interesting to study. Innovate the traditional methods and propose new attributes to collect. Even if in the end those attributes do not contribute too much. These attributes could not be easy to see. Think outside the box.

d) Selection of ground truth

Data collected need to be labeled with ground truth. The ground truth is used to check the results of machine learning for accuracy against the real world. The ground truth is a label that describes a class of the phenomenon of interest. For example, if the phenomenon is the emotions of a person, the classes could be "happy", "angry", or "sad".

Select the ground truth for your research. The ground truth is often given by a self-report of the subject or by a psychology scale.

e) Selection of attributes

Finally, select the attributes of study that are more important and more innovative. The attributes should cover the project objectives. All selected attributes will be collected and you should consider the project resources.

4.2.2 Development of data collection methods

The objective of this general task is to develop data collection methods. The development of the methods goes hand in hand with the selected attributes and the project resources. The specific tasks are the following.

a) Selection of data collection methods

Data can be collected by different methods like observation, scales, surveys, interviews, etc. Surveys could have different types of answer such as open answer, single answer selection, multiple answer selection, range selection, etc. Select the more convenient questions to your research considering the time for answering by the subjects and the easiness to compile the data collected.

These traditional methods have been used for many years. However, they present some disadvantages. Some data are easy to remember, as the age of the subject, the number of children, the number of rooms in the house; people can accurately remember them. But some other type of data could bring problems, like the number of calls received five days ago, or the number of times you have visited the supermarket; people could forget the data or they could give inaccurate information. Moreover, there are data that are impossible to know, such as the number of keys that you have given today, the number of clicks, or the number of steps taken. For this type of information, it is necessary to use more precise and objective data collection methods.

Thanks to the advancement of technology, it is now possible to collect large amounts of data continuously through sensors embedded on devices such as: smartphones, tablets, computers, wearable devices, etc.; or sensors embedded on the ambient such as: temperature sensors, light sensors, surveillance cameras, microphones, beacons, etc. These methods provide objective data, unlike the traditional methods where bias may occur. These methods also have disadvantages since the sensors could lose information or noise may appear in the data. However, there are techniques to manage and correct these errors. These techniques will be described in the next phase.

To select the data collection methods, review the list of attributes and classify them by source. It is possible to combine traditional and new data collection methods. Consider that data must be saved in a structured way in order to form data instances.

Think the best way to collect each type of attributes considering the time you have for developing the data collection methods, and the project resources. Consider also that data could be collected manually or automatically, depending on the attributes selected.

Also consider looking for data collection methods that are available to use, like surveys, scales, software, or commercial devices. You could use these available methods and save development time. Just be sure that you will not have license problems, and that the method gives you the data you really need. If it does not give you all the data, you could always adjust the method by customizing.

b) Study of state of the art

Search related works in books and scientific journals. Look for the data collection methods used by researchers. Maybe they have used some methods that you have not considerate. List the selected method for collecting all attributes.

c) Development of data collection methods

Develop the selected methods for collecting data. Try to collect all your attributes as objective as possible. Evaluate the method by carrying out a pilot test, or at least, small tests with people near to you. You need to know how the data is provided by the subject, how is saved, how the subject has felt in the process if there was something unclear, or an error. You could adjust your methods before launching the data collection process.

For the survey method, carefully design the questions that have to be answered by the subjects. The questions should be simple, not ambiguous, and easy to answer. The type of information you need, the depth of information you need, and the amount of time your respondents have available will all influence your choice of survey type. Be sure that your instrument can collect all your attributes in the format you need. It is better if you have expert advice.

For scale method, follow the instructions of the scale. Normally, scales are validated in a specific language, and you must use the exact items. If some items result ambiguous, you should look for another version of the scale or another scale. Be sure that the scale you are using has a good validity index.

For the interview method, carefully design the questions that have to be answered by the subjects. The interviews provide more wealth in the data collected since the interviewer can ask more questions if something is unclear. However, this method is more expensive. You could record the audio of the interview to avoid missing information provides by the subject, but you should ask him.

For the observation method, design a format where the observer will fill in. It is necessary to define exactly what information must include the observation.

For automatic methods, classify the attributes by source, and develop modules for each type of attributes. Maybe you will have a large monitoring system composed of several modules which collect data from attributes from the same source, e. g., a module that collects data from a smartphone, from a camera, or from a wearable device. Define a system architecture. The modules do not have to be only in one programming language. You could merge different languages and technologies. It is more convenient that all data be stored in a common database, which could be local or on a server.

Develop the instrument for collecting the ground truth. If you collect data but do not collect the ground truth, the data could not be used. The ground truth is often given by a self-report of the subject or by a psychology scale. You could include the self-report in your monitoring system to ask the subject n times a day.

Be careful with sensitive data. Sensitive data is defined as information that is protected against unwarranted disclosure. The following personal data is considered 'sensitive' and is subject to specific processing conditions: personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs; trade-union membership; genetic data, biometric data processed solely to identify a human being; health-related data; and data concerning a person's sex life or sexual orientation. If you want to collect this type of data, you must tell the subjects exactly what attributes you are collecting, and they must sign an inform consent.

4.2.3 Definition of the sampling

The objective of this general task is to define the sampling. Sampling consists of selecting part of a population to observe so that one may estimate something about the whole population [146]. The specific tasks are the following.

a) Selection of the type of sampling

There are two major types of sampling: probability sampling and non-probability sampling. Probability sampling is a type of sampling where each member of the population has a known probability of being selected in the sample. When a population is highly homogeneous, its every member has a known chance of being selected in the sample. Using this type of sampling, you will be more able to generalize the results of your study. However, probability sampling methods tend to be more time-consuming and expensive than non-probability sampling. This sampling technique includes sample selection which is based on random methods. The techniques that are based in this category are random sampling, stratified sampling, systematic sampling, and cluster sampling.

Non-probability sampling is a type of sampling where each member of the population does not have known probability of being selected in the sample. In this type of sampling, each member of the population does not get an equal chance of being selected in the sample. Non-probability sampling is adopted when each member of the population cannot be selected or the researcher deliberately wants to choose members selectively. For example, to study the impacts of domestic violence on children, the researcher will not interview all the children but will interview only those children who are subjected to domestic violence. Hence, the members cannot be selected randomly. In non-probability sampling, there is a significant risk of ending up with a non-representative sample which produces non-generalizable results. However, non-probability sampling methods tend to be cheaper and more convenient, and they are useful for exploratory research and hypothesis generation. This sampling technique is not based on random selection. Some examples are quota sampling, purposive sampling, and convenience sampling.

Search and select the type of sampling more convenient to your research based on the project objectives, the project budget, and the time you have to do your project. Select your sample as representative as possible.

b) Definition of subjects' profile

Choose the segment of the population that you want to study based on the phenomenon of interest. Define the profile of the subjects, that is to say, define the characteristics that subjects must have in order to take part in the data collection process. For example, define their age range, their occupation, marital status, etc. Also, you could define if the subjects must have some device like a smartphone, or if they must use some software like Windows OS, or if they must have some skills using technology, etc.

4.2.4 Acquisition of data

The objective of this general task is to carry out a process to collect data from the sample. The specific tasks are the following.

a) Data collection process design

Design the data collection process by thoroughly defining the activities that must be done before, during, and after your data collection procedure. It is necessary to define the activities that will be carried out by all the people involved: subjects, researchers, assistants, etc. The activities that could be done before the data collection procedure are the following:

- To train people to carry out the data collection procedure in case that you will have assistants.
- To give to the subjects a general description of the research, the objective of the study, and the data collection procedure.
- To inform the subjects the data that will be collected and how will be collected.
- To inform the subjects about their rights, risks, and benefits they obtain, and how will their data be treated.
- To sign an inform consent by the subjects.
- To fill a survey by the subjects to collect personal data like age, weight, height, marital status, number of children, or any data relevant to your research.
- To train the subjects to answer a scale, or to use technological devices.
- To give to the subjects the needed technological devices.
- To install software in computers, tablets, or smartphones.
- To install sensors in the environment.

The activities that could be done during the data collection procedure are the following:

- To answer a scale, a self-report, or questions in an interview by the subject.
- To act normal while the monitoring is occurring, either by human observer, or by technological monitoring.
- To charge technological devices.

- To check if data is being correctly saved while the monitoring is occurring by the researchers.
- To check for missing data in order to correct the monitoring sensor.
- To answer possible doubts from the subjects.

The activities that could be done after the data collection procedure are the following:

- To answer the last questionnaire.
- To uninstall the software
- To give back the technological devices.
- To give reviews about the data collection.
- To define the rewards for the subjects.

Define the duration of the data collection process. The predictive model is generated through a training process where the predictive model learns from the data available. The predictive model has to learn from data instances of each class of the phenomenon of interest. For example, if the phenomenon is depression, the classes could be "depressed" or "not depressed", or they could be depression levels like "low", "medium", and "high". The predictive model has to learn from each class in order to be able to predict any of the class outputs. Therefore, the data collection process must last a favorable period of time that allows collecting enough data for each class. In some cases, it is difficult to collect data from a particular class, for example, class "depressed" or "highly depressed". Furthermore, there are problems where a class imbalance is not just common, it is expected. The first recommendation is to collect as many of the minority data instances as you can. Then, there are some techniques for resampling and undersampling that you could apply. These techniques are described in the next phase.

b) Reaching the subjects

Reach the subjects according to your sampling type and subject profile defined. You could ask for permission for collecting data in institutes, companies, or schools, depending on the phenomenon of interest. In order to get permission, you could offer them compensation like a course, a conference, training, or some material resources. You could also try to reach subjects where they usually attend. For example, if your research is about older adults, you could reach them in senior clubs or nursing homes.

c) Data collection procedure

Prepare all the materials that you need for the data collection procedure. Arrange a date with the subjects. Thoroughly carry out all the activities defined in your data collection process design.

4.3 Phase 2. Data preprocessing

Phase 2 of the methodology is data preprocessing. Data preprocessing is the conversion of data into a usable and desired form called dataset. Raw data is highly susceptible to noise, missing values, and inconsistency. The quality of data affects the data mining results. In order to help improve the quality of the data and, consequently, of the mining results raw data is preprocessed so as to improve the efficiency and ease of the mining process. The input of this phase is the collected database and the output are *n* datasets. Phase 2 consists of five general tasks which are depicted in Figure 4.3.

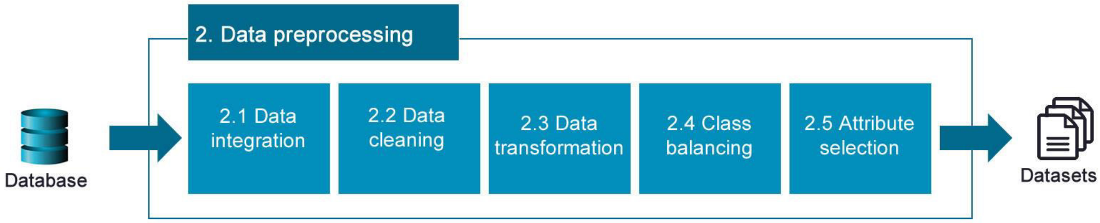


Figure 4.3 General tasks of phase 2

4.3.1 Data integration

The objective of this general task is to compile the data and to generate a set of instances. Data mining often requires data integration, that is to say, the merging of data from multiple data stores. The specific tasks are the following.

a) Data merging

Compile the data from the different sources you used in a unified view to generate a set of instances. Data instances capture the behavior of a subject or an object through a set of values of all the attributes of study in a certain period of time, for example, instances of every minute, of every hour, or of every day. Table 4-1 shows an example of data instances.

Table 4-1 Example of data instances

Date	Incoming calls	Outgoing calls	Number of places visited	Time spent out of home	…
07/23/2016	3	1	3	6	
07/24/2016	1	0	1	1	
07/25/2016	2	3	2	5	
…					

The instances must include the attributes from the different data collection methods you used. For example, attributes from an interview, from a scale, or from the sensors embedded in the smartphone.

b) Data summarization by time-windows

Data may have been stored in instances minute by minute but the ground truth may have been collected in every three-hour time window. It is necessary to summarize the behavior collected in each time window to assign the ground truth reported by the subject. Summarize the data by summing the attributes collected in each time window. Table 4-2 shows an example of data summarization by a ten-minutes time window.

Table 4-2 Example of data summarization by time windows

Datetime	Keystroke	Clicks	Scrolls	Pixels	Ground truth (class)
7/5/17 10:51	17	9	1007	2188	
7/5/17 10:52	4	13	4123	2480	
7/5/17 10:53	39	19	0	1356	
7/5/17 10:54	4	23	4162	2835	
7/5/17 10:55	60	18	0	2135	
7/5/17 10:56	71	10	4165	1480	
7/5/17 10:57	107	1	0	272	
7/5/17 10:58	0	7	0	926	
7/5/17 10:59	0	3	0	576	
7/5/17 11:00	0	4	4203	1188	Low
Instance→	**302**	**107**	**17660**	**15436**	**Low**

4.3.2 Data cleaning

The objective of this general task is to clean the data. Data instances can be incomplete, that is to say, they are lacking attributes values. There may have been human or computer errors occurring at data entry. This situation also occurs when the respondent does not respond to certain questions due to stress, fatigue or lack of knowledge. The respondent may not respond because some questions are sensitive. Data instances can be noisy, that is to say, containing errors, or outlier values which deviate from the expected. Data instances can be inconsistent, that is to say, containing discrepancies in codes or names. Dirty data can cause confusion for the mining process. Although most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust. Therefore, you can apply some data cleaning methods. The specific tasks are the following.

a) Filling in missing values

A missing value occurs when no data value is stored for the attribute. Missing value is a common occurrence and can have a significant effect on the conclusions that can be drawn from the data. Look for missing values in your database and fill in by one of the methods described below.

1. *Ignore the instance.* This is usually done when the class label is missing. This method is not very effective unless the instance contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.
2. *Fill in missing value manually.* In general, this approach is time-consuming and may not be feasible given a large dataset with many missing values.
3. *Use a global constant to fill in the missing value.* Replace all missing attribute values by the same constant, such as a label like "unknown". But, if missing values are replaced with a constant, then the mining program may mistakenly think that they form an interesting concept since they all have a value in common. Hence, although this method is simple, it is not recommended.
4. *Use the mean.* Use the attribute mean to fill the missing value.
5. *Use the mean of the class.* use the attribute mean for all samples belonging to the same class as the given tuple.
6. *Use the most probable value.* This may be determined with inference-based tools using a Bayesian formalism or decision tree induction.

b) Smoothing noisy data

Noise can negatively affect the predictive model performance in terms of classification accuracy, building time, size and interpretability. The presence of noise in the data may affect the intrinsic characteristics of the problem. Noise may create small clusters of instances of a particular class in parts of the instance space corresponding to another class, remove instances located in key areas within a concrete class or disrupt the boundaries of the classes and increase overlapping among them. Several approaches have been studied to deal with noisy data. Smooth the noisy data of your database through one or more of the following techniques.

1. *Robust methods.* These are techniques characterized by being less influenced by noisy data. For example, the C4.5 algorithm uses pruning strategies to reduce the possibility that trees overfit to noise in the training data.
2. *Data polishing methods.* Their aim is to correct noisy instances prior to training an algorithm. This option is only available when datasets are small because it is generally time-consuming.
3. *Noise filters.* These filters identify noisy instances which can be eliminated from the training data. They are used with many algorithms that are sensitive to noisy data and require data processing to address the problem.
4. *Binning methods.* Binning methods smooth a sorted data value by consulting the neighborhood, or values around it. The sorted values are distributed into a number of buckets or bins. Because binning methods consult the neighborhood of values, they perform local smoothing.
5. *Clustering.* Outliers may be detected by clustering, where similar values are organized into groups or clusters.

6. *Combined computer and human inspection.* Outliers may be identified through outlier patterns searched by computer and confirmed by humans. This is much faster than having to manually search through the entire database.
7. *Regression.* Data can be smoothed by fitting data to a function such as with regression. Linear regression involves finding the best line to fit two variables so that one variable can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two variables are involved and the data are fit to a multidimensional surface. Using regression to find a mathematical equation to fit the data helps smooth out the noise.

c) The correctness of inconsistent data

There may be inconsistencies in the data recorded for some transactions. Some data inconsistencies may be manually corrected using an external reference. For example, errors made at data entry may be corrected by performing a paper trace. This may be coupled with routines designed to help correct the inconsistent use of codes. Knowledge engineering tools may also be used to detect the violation of known data constraints. For example, known functional dependencies between attributes can be used to find values contradicting the functional constraints. Correct the inconsistencies of your database.

4.3.3 Data transformation

The objective of this general task is to transform the collected data. There are always some data forms that are best suited for specific learning algorithms. Data transformation is an approach to transform the original data into a preferable data format for the input of certain learning algorithms before the processing. Therefore, you can apply some data transformation methods. The specific tasks are the following.

a) Data normalization

The data should be normalized to avoid dependency on the choice of measurements units on data attributes. This means transforming or mapping the data to a smaller specific range such as [-1, 1] or [0, 1]. All attributes gain an equal weight after this process. Normalize your data. There are many normalization methods like min-max normalization, z-score normalization, and normalization by decimal scaling.

b) Data aggregation

Summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly or annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities. Do this task as you needed.

c) Data generalization

It is the process of transform low-level values to higher-level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher-level concepts, like city or country. Similarly, values from numerical attributes, like age, may be mapped to higher-level concepts, like youth, middle-age, and senior. Do this task as you needed.

4.3.4 Class balancing

The objective of this general task is to balance the classes of the data. In an unbalanced database, classes are not equally represented [147], and this imbalance of classes can cause a bias toward the majority class during the predictive model development. The specific tasks are the following.

a) Data resampling and undersampling

Balance your database by applying one of the following techniques for resampling or undersampling.

1. *Resample* [113]. This technique produces a random subsample of a dataset using either sampling with replacement or without replacement. This is a hybrid technique randomly removes objects from the majority class while applying an oversampling in the minority class to obtain a fully balanced sample. This technique can use resampling with replacement or without replacement. This technique has a parameter B that specifies the desired balance level between the classes; the values close to one obtain samples with more balance between the classes.
2. *Spread subsample* [113]. This technique adjusts the class distribution through a random subsample of the majority class instances. The distribution is calculated depending on the value Spread which is determined by the user. The spread parameter specifies the imbalance ratio 1:IR, that is to say, for each instance of the minority class how many exist in the majority class.
3. *SMOTE* [147]. Synthetic minority oversampling technique SMOTE creates synthetic instances to over-sample the minority class, and it also under-samples the majority class if necessary. It creates synthetic instances between the k near neighbors of each object belonging to the minority class. Synthetic instances are calculated by the difference of the feature vector of the instance under consideration with its nearest neighbor, then these differences are multiplied randomly by zero or one. This method has a parameter P that specifies the percentage of synthetic objects to be created relative to the number of objects, of the original sample, that belong to the minority class.

A dataset contains a subset of attributes and/or a subset of instances. Create a dataset from the balanced database. This first dataset must contain all attributes and all instances that you have collected.

4.3.5 Attribute selection

The objective of this general task is to select a subset of relevant attributes to be used in the construction of the predictive model. This process is also known as variable selection, feature selection, or variable subset selection. Attribute selection techniques have several benefits, but basically, they are used because they simplify models, making them easier to interpret. Also, the attribute selection process reduces training times and their storage needs. At the same time, a better view and understanding of the data is obtained. The goal of the attribute selection process is to determine which data have redundant or irrelevant features that can be removed without much loss of information [148]. Attribute selection analysis can be carried out with specialized software such as SPSS [149], Weka [113], R [150], Orange [151], etc. The specific tasks are the following.

a) Correlation analysis

Correlation is an analysis that measures the strength of association between two variables and the direction of the relationship. In terms of the strength of the relationship, the value of the correlation coefficient varies between +1 and -1. A value of ± 1 indicates a perfect degree of association between the two variables. As the correlation coefficient value goes towards 0, the relationship between the two variables will be weaker. The direction of the relationship is indicated by the sign of the coefficient; a + sign indicates a positive relationship and a – sign indicates a negative relationship. According to [152], the strength of the correlation coefficient is conventionally interpreted as shown in Table 4-3.

Table 4-3 Strength of correlation coefficient

Correlation coefficient value	Interpretation
0.00 – 0.19	Very weak correlation
0.20 – 0.39	Weak correlation
0.40 – 0.59	Moderate correlation
0.60 – 0.79	Strong correlation
0.80 – 1.00	Very strong correlation

In order to carry out the correlation analysis, define a null hypothesis H_0 and an alternative hypothesis H_1. The null hypothesis means that there is no correlation between the attribute and the class. Therefore, the alternative hypothesis means that there is a correlation between the attribute and the class. Also, set a significance level. It is usually set at or below 5%. The significance level is the probability of rejecting the null hypothesis given that it was true. The *p-value* is the probability of obtaining a result at least as extreme given that the null hypothesis was true. The correlation analysis must be carried out by calculating one of the following coefficients.

1. *Pearson product-moment correlation coefficient.* The Pearson correlation coefficient is a measure of the strength of a linear association between two continuous variables and is denoted by r. Basically, a Pearson product-moment correlation attempts to draw a line of best fit through the data of two variables, and

the Pearson correlation coefficient, r, indicates how far away all these data points are to this line of best fit. It is known as the best method of measuring the association between variables of interest because it is based on the method of covariance. It gives information about the magnitude of the association, or correlation, as well as the direction of the relationship.

2. *Spearman's correlation.* Spearman's correlation coefficient measures the strength of a monotonic relationship between two ranked variables. A monotonic relationship is a relationship that does one of the following: i) as the value of one variable increases, so does the value of the other variable; or ii) as the value of one variable increases, the other variable value decreases.
3. *Kendall rank correlation.* The Kendall rank correlation coefficient evaluates the degree of similarity between two sets of ranks given to the same set of objects. This coefficient depends upon the number of inversions of pairs of objects which would be needed to transform one rank order into the other. In order to do so, each rank order is represented by the set of all pairs of objects, and a value of 1 or 0 is assigned to this pair when its order corresponds or does not correspond to the way these two objects were ordered. This coding schema provides a set of binary values which are then used to compute a Pearson correlation coefficient.

Carry out the correlation analysis between each attribute and the class. Choose the coefficient based on the data type of each attribute. Create a second dataset with the set of attributes which are correlated to the class.

b) Application of attribute selection methods

There are several attribute selection methods. You can use as many as you want. For each method that you use, create a dataset with the resulting attributes. You will have *n* datasets. It is recommended to use at least two attribute selection methods so you can compare the classifier performance of the two datasets. The methods are described below.

1. *Principal components analysis.* Principal components analysis [153] is a statistical method that linearly transforms an original set of features into a substantially smaller set of uncorrelated features representing most of the information in the original set of features. Its goal is to reduce the dimensionality of the original set.
2. *Correlation-based Feature Selection.* Correlation-based Feature Selection [154] evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.
3. *Info Gain.* Info Gain evaluates the worth of an attribute by measuring the information gain with respect to the class. Information Gain is calculated by the feature's contribution to decreasing overall entropy [155]. The Info Gain method gives a ranked list according to the highest information gain score.
4. *Wrappers for Feature Subset Selection.* The Wrapper method [156] evaluates attribute sets by using a learning scheme. Cross-validation is used to estimate the

accuracy of the learning scheme for a set of attributes. This method must be evaluated for each classifier.

4.4 Phase 3. Predictive models' development

Phase 3 of the methodology is predictive models' development. Predictive models' development is carried out through learning algorithms. The input of this phase are the *n* datasets from the previous phase, and the output is the predictive models. Phase 3 consists of two general tasks which are depicted in Figure 4.4.

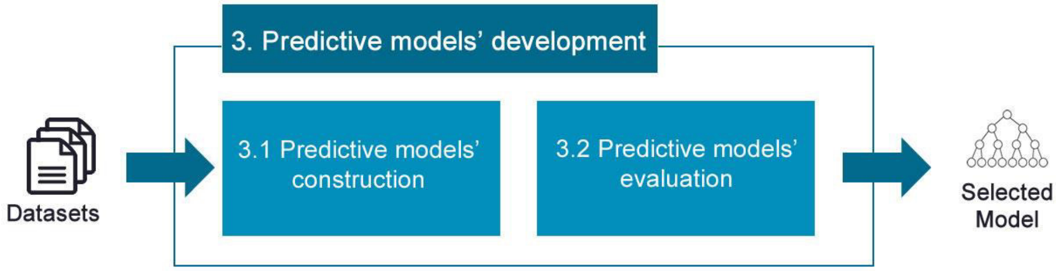

3. Predictive models' development
Datasets
3.1 Predictive models' construction
3.2 Predictive models' evaluation
Selected Model

Figure 4.4 General tasks of phase 3

4.4.1 Predictive models' construction

The objective of this general task is to build predictive models. The predictive models' construction comprises two stages. The first stage is training *m* learning algorithms using the *n* datasets from the previous phase. The second stage is testing the built predictive models. The specific tasks are the following.

a) Selection of model schemes

There are some schemes that can be used to build diverse predictive models. Each scheme shows the prediction performance from different points of view to make comparisons and to select the best predictive model or group of predictive models. Each scheme also uses a particular technique to split the dataset into training and testing sets. Select the schemes that are suitable for your research.

- *General model - Cross-validation.* General model scheme is very used in many types of research. A general model is built by training a learning algorithm using two-thirds of the data and using the remaining one-third for testing.
- *General model – Leave-One-Out.* This scheme uses the leave-one-out technique. That is, a general model for a subject *i* is built by training a learning algorithm using data from all other subjects *j*, where $i \neq j$. The predictive model is tested with data from subject *i*. When the predictive models of all subjects are built, an average of predictive models' performance is calculated.

- *Individual models.* In the individual model scheme, an individual model for a subject *i* is built by training a learning algorithm using data only from subject *i*. This scheme uses cross-validation technique. In cross-validation, the data is split into *k*-folds. In turn, each fold is used for testing and the remainder is used for training. The procedure is repeated *k* times so that, in the end, every instance has been used exactly once for testing. It is standard procedure to repeat the cross-validation process 10 times 10-fold cross-validation and average the results. This involves invoking the learning algorithm 100 times on datasets that are all nine-tenths the size of the original [148]. When the predictive models for each subject are built, an average of all individual predictive models' performance is calculated.
- *Similar subjects.* In the similar user scheme, a model for a subject *i* is built by training a learning algorithm using data from a subset of similar subjects. The similar subjects are defined by using a clustering algorithm which creates groups of subjects with similar behavior. The predictive model is built using the leave-one-out technique within each cluster.

b) Creation of training and testing sets

Create the training and testing sets according to the selected model schemes.

For general models – cross-validation scheme, create a training set containing the two-thirds of the data, and testing set containing the remaining one-third of the data. Some workbenches like Weka have the option to split the dataset, so you only have to load your dataset.

For general models – leave-one-out scheme, create a training set containing data of all subjects except one, and testing set containing data from the leave-out subject. Create this pair of sets for each subject.

For individual models' scheme. Create a training set containing data only from one subject. This scheme uses cross-validation, so it is not necessary for a testing set. You must have a set for each subject.

You can also create new datasets selecting subjects by age, country, gender, or any attribute of your interest. For example, if you have data from subjects of different ages, you may want to create a predictive model for children, another for adults, and a third one for the elderly. So, take a dataset, split the subjects by age, and create the corresponding training and testing sets.

c) Selection of classifiers

Classification is the process of creating a model capable of predicting a class, and it is called a predictive model. The predictive model is created by training a learning algorithm which finds relationships between unknown instances and a set of correctly classified instances [157]. Classification is a supervised learning technique because the class of each training instance is provided. It contrasts with unsupervised learning or clustering, in which the class of each training instance is not known, and the number of classes to be learned may not be known in advance.

There are a wide variety of classifier methods to match the nature of data. The most common classifier methods are decision trees, rule-based methods, probabilistic methods, SVM methods, instance-based methods, and neural networks.

Probabilistic methods. Probabilistic methods are the most fundamental among all data classification methods. Probabilistic methods use statistical inference to find the best class for a given example. In addition to simply assigning the best class, probabilistic algorithms will output a corresponding posterior probability of the test instance is a member of each of the possible classes. The posterior probability is defined as the probability after observing the specific characteristics of the test instance. On the other hand, the prior probability is simply the fraction of training records belonging to each particular class, with no knowledge of test instance. After obtaining the posterior probabilities, it is used as decision theory to determine class membership for each new instance. Some of the methods are Naïve Bayes [158], logistic regression, probabilistic graphical models, and conditional random fields. An overview of probabilistic methods for classification are found in [159], [160]

Decision trees. Decision trees create a hierarchical partitioning of the data, which relates the different partitions at the leaf level to the different classes. The hierarchical partitioning at each level is created with the use of split criterion. The split criterion may either use a condition on a single attribute, or it may contain a condition on multiple attributes. The former is referred to as a univariate split, whereas the latter is referred to as a multivariate split. The overall approach is to try to recursively split the training data so as to maximize the discrimination among the different classes over different nodes. Some of the methods for decision tree construction include C4.5 [161], ID3 [162], and CART [163].

Rule-based methods. Rule-based methods are close to decision trees, except that they do not create a strict hierarchical partitioning of the training data. Rather, overlaps are allowed in order to create greater robustness for the training model. Any path in a decision tree may be interpreted as a rule, which assigns a test instance to a particular class. In fact, a number of methods such as C4.5, create related models for both decision tree construction and rule construction. The corresponding rule-based method is referred to as C4.5 Rules [164]. Rule-based methods can be viewed as more general models than decision tree models. While decision trees require the induced rule sets to be non-overlapping, this is not the case for rule-based methods. In rule-based methods, a set of rules is mined from the training data in the training stage. During the testing stage, it is determined which rules are relevant to the test instance and the final result is based on a combination of the class values predicted by the different rules. Some of the rule-based methods are Classification Based on Associations (CBA) [165], CN2 [166], and RIPPER [167].

Instance-based learning. In instance-based learning, the first stage of constructing the training model is often dispensed with. The test instance is directly related to the training instances in order to create a classification model. Such algorithms are referred to as lazy learning methods because they wait for knowledge of the test instance in order to create a locally optimized model, which is specific to the test instance. The advantage of such methods is that they can be directly tailored to the particular test instance, and can avoid the information loss associated with the incompleteness of any training model. An example of a very simple instance-based method is the nearest neighbor algorithm *k*-Nearest Neighbor [168]. In the nearest neighbor algorithm, the top k nearest neighbors in the training data are found to the given test instance. The class with the largest presence among the k nearest neighbors is reported as the relevant class.

SVM classifiers. SVM methods use linear conditions in order to separate out the classes from one another. The idea is to use a linear condition that separates the two classes from each other as well as possible. An SVM classifier may be considerate a single level decision tree with a very carefully chosen multivariate split condition. Since the effectiveness of the approach depends only on a single separating hyperplane, it is crucial to define this separation carefully. The major downside of SVM methods is that they are slow. However, they are very popular and tend to have higher accuracy in many domains such as text [169].

Neural networks. Neural networks attempt to simulate biological systems, corresponding to the human brain. In the human brain, neurons are connected to one another via points, which are referred to as synapses. In biological systems, learning is performed by changing the strength of the synaptic connections, in response to impulses. This biological analogy is retained in an artificial neural network. The basic computation unit in an artificial network is a neuron or a unit. These units can be arranged in different kinds of architectures by connections between them. The most basic architecture of the neural network is a perceptron, which contains a set of input nodes and an output node. The output unit receives a set of inputs from the input units. There are different input units, which is exactly equal to the dimensionality of the underlying data. The data is assumed to be numerical. Categorical data may need to be transformed into binary representations, and therefore the number of inputs may be larger. The output node is associated with a set of weights, which are used in order to compute a function of its inputs. Each component of the weight vector is associated with a connection from the input to the output unit. The weights can be viewed as the analog of the synaptic strengths in biological systems. A number of implementations of neural network methods have been studied in [170]–[174] [35, 57, 66, 77, 88], and many of these implementations are designed in the context of text data. Both neural networks and SVM classifiers use a linear model that is quite similar. The main difference between the two is in how the optimal linear hyperplane is determined. Rather than using a direct optimization methodology, neural networks use a mistake-driven approach to data classification [170].

There are other classification methods like Random Forest [175], and AdaBoost [176]. You could also consult the top ten algorithms in data mining in [177]–[179]. Select the classification methods suitable for your research. It is important to select several classification methods in order to compare the performance among them and to select the best ones. It is recommended to use five classifier methods at least.

Select a classifier like Zero Rule or One Rule [180] as your baseline. A baseline provides a point of comparison for the more advanced methods. Zero Rule predicts the mean for a numeric class or the mode for a nominal class. One Rule uses the minimum-error attribute for prediction, discretizing numeric attributes.

d) Classification

Classifier algorithms can be implemented in a programming language or you can use a data mining workbench. Some of the best programming languages are R, Python, Java, and Julia.

R [150]. R is a language that dates back to 1997. It was a free substitute to exorbitant statistical software such as SAS or MATLAB. R programming language can be used to sift through very complex data sets and to create very sleek graphics that represent numbers in a few code lines. This language has the most ideal asset as well as a supportive ecosystem that has been developed around it. The R language is an influential open source useful programming language. At its essential, R is a numerical programming language that offers imposing tools for data mining plus analysis. It allows you to make high-level graphics plus offers an interface toward other languages. This means R is greatest suited to creating data and visual analytics over customization letterings and commands, in place of the typical numerical tools that offer tick boxes plus drop-down menus for consumers. New features and packages were frequently incorporated into its robust function sets. R is ranked as being the most popular language with regard to data science. Its use is on the rise especially for data mining and financial modeling. In financial modeling, it is used as an effective visualization tool.

Python [181]. Python is also a suitable programming language for data mining with more practical capabilities and fast data mining capabilities to make a good product. It can be used for statistical analysis that was initially the forte of R. It has emerged as an excellent option in the processing of data creating a trade-off between sophistication and scale. Python is an ideal language for building tools for performing data processing on a medium scale. It is fitted with adequate features and toolkits as well as a wealthy data community. In fact, most banks are utilizing Python to build new products and interface. It is flexible and broad thus people can assemble it easily. However, it should be noted that Python is not the highest in performance. Sometimes, this language powers up large-scale infrastructure.

Julia [182]. Most of the data mining is currently done by SAS, R, MATLAB, and Java but this still leaves a gap that Julia fills. This programing language has been widely adopted by the data mining industry because of a number of reasons: i) Julia is an expressive language, ii) it is a fast language, and iii) it is a high-level language. Julia is intended to offer the ease of use and productivity of Python with the mathematical prowess of MATLAB and the performance of C so you can do it all in one. It supports parallel distributed computing and can be used interactively with data science notebooks like Jupiter. It also supports Lisp-like macros. It is very promising as its data community is in the infancy stage and needs additional packages to be at the same level with Python and R.

Java [183]. One of the most practical languages to have been designed, a large number of companies, especially big multinational companies use the language to develop backend systems and desktop apps. It is been touted as a mainstay of the enterprise software stack as the demand for Java skills is only increasing with time. It is believed to be a good choice for someone starting out as a programmer, as it is a relatively simple and readable programming language. Having witnessed the biggest rise in demands in the last few years, Java skills have been in demand particularly for software engineers, software architects, and DevOps engineers. Java does not have a similar quality of visualization similar to R and Python. It is a language which is not finest for numerical modeling, however, if you want to create the big system and moving fast prototyping, Java is the finest language.

There are several data mining workbenches. A data mining workbench is a platform or environment that supports and facilitates a range of machine learning activities reducing or removing the need for multiple tools. Some statistical and machine learning workbenches provide very advanced tools but require manual configuration in the form of scripts and programming. Some of the most used workbenches are:

Weka [113]. Also known as Waikato Environment is a machine learning software developed at the University of Waikato in New Zealand. It is best suited for data analysis and predictive modeling. It contains algorithms and visualization tools that support machine learning. Weka has a GUI that facilitates easy access to all its features. It is written in Java programming language. Weka supports major data mining tasks including data mining, processing, visualization, regression etc. It works on the assumption that data is available in the form of a flat file. Weka can provide access to SQL Databases through database connectivity and can further process the data/results returned by the query.

Orange [151]. Orange is a perfect software suite for machine learning & data mining. It is the best in data visualization. It has been written in Python computing language. As it is a component-based software, the components of orange are called widgets. These widgets range from data visualization and pre-processing to an evaluation of algorithms and predictive modeling. Widgets offer major functionalities such as: showing data table and allowing to select features; reading the data; training predictors and to compare learning algorithms, and visualizing data elements. Additionally, Orange brings a more interactive and fun vibe to the dull analytic tools. It is quite interesting to operate. Data coming to Orange gets quickly formatted to the desired pattern and it can be easily moved where needed by simply moving/flipping the widgets. Orange allows users to make smarter decisions in a short time by quickly comparing and analyzing the data.

RapidMiner [184]. Rapid Miner is one of the best predictive analysis systems developed by the company with the same name as the Rapid Miner. It is written in Java programming language. It provides an integrated environment for deep learning, text mining, machine learning, and predictive analysis. The tool can be used for over a vast range of applications including for business applications, commercial applications, training, education, research, application development, machine learning. Rapid Miner offers the server as both on-premise and in public/private cloud infrastructures. It has a client/server model as its base. Rapid Miner comes with template-based frameworks that enable speedy delivery with a reduced number of errors, which are quite commonly expected in the manual code writing process. Rapid Miner constitutes of three modules, namely: i) Rapid Miner Studio, this module is for workflow design, prototyping, validation etc.; ii) Rapid Miner Server, to operate predictive data models created in the studio; and iii) Rapid Miner Radoop, executes processes directly in Hadoop cluster to simplify predictive analysis.

Knime [185]. KNIME is the best integration platform for data analytics and reporting. It operates on the concept of the modular data pipeline. KNIME constitutes of various machine learning and data mining components embedded together. KNIME has been used widely for pharmaceutical research. In addition, it performs excellently for customer data analysis, financial data analysis, and business intelligence. KNIME has some brilliant features like quick deployment and scaling efficiency. Users get familiar with KNIME in quite lesser time and it has made predictive analysis accessible to even naive users. KNIME utilizes the assembly of nodes to preprocess the data for analytics and visualization.

Implement the selected classifier methods in a programming language or select a workbench at your convenience. Workbenches are a good option when you have limited time. However, implementing the classifier in a programming language will give you more flexibility and customization. Run the classifier algorithms using the training sets as input. You will have *n x m* datasets where *n* is the number of the created training sets, and *m* is the number of classifier methods.

4.4.2 Predictive models' evaluation

The objective of this general task is to evaluate the predictive models and select the best predictive model or group of predictive models based on evaluation metrics. The specific tasks are the following.

a) Selection of evaluation metrics

The evaluation metrics are used to assess the performance of the predictive models. The performance of a predictive model refers to its ability to correctly predict the class into which new or previously unseen data are classified. The metrics that you choose to evaluate your predictive model are very important. The choice of metrics influences how the performance of learning algorithms are measured and compared.

There are four possible outcomes when a predictive model classifies an outcome for an instance. If the instance is positive and it is classified as positive, it is counted as a true positive (TP); if it is classified as negative, it is counted as a false negative (FN). If the instance is negative and it is classified as negative, it is counted as a true negative (TN); if it is classified as positive, it is counted as a false positive (FP). When a predictive model classifies a whole testing set, a two-by-two confusion matrix, also called a contingency table, can be constructed representing the dispositions of the set of instances. This matrix forms the basis for many common metrics. A confusion matrix for binary classification is shown in Table 4-4.

Table 4-4 A confusion matrix for binary classification

	Actual positive	Actual negative
Predictive positive	TP	FN
Predicted negative	FP	TN

The numbers along the major diagonal represent the correct decisions made, and the numbers of this diagonal represent the errors, the confusion, between the various classes.

Accuracy is one of the most common metrics in practice used by many researchers. Through accuracy, the predictive model is measured based on total correctness which refers to the total of instances that are correctly predicted by the predictive model when tested with the unseen data. Accuracy is defined in Eq. (1).

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+FP+TN+FN}} \tag{1}$$

Accuracy is a good metric when the classes in the data are nearly balanced. However, accuracy does not distinguish between the number of correct predictions of different classes. High accuracy may not be acceptable since the predictive model could correctly classify only the negatives instances giving no indication about the ability of the predictive model to classify positive and negative instances.

Conversely, two metrics that separately estimate a predictive model's performance on different classes are *sensitivity* and *specificity*. Sensitivity assesses how well the predictive model classifies positive instances. Specificity measures how well the predictive model classifies negative instances. They are often employed in biomedical and medical applications, and in studies which involve image and visual data. Sensitivity and specificity are defined in Eq. (2 – 3) respectively.

$$\boldsymbol{Sensitivity} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{2}$$

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN+FP}} \tag{3}$$

Another group of metrics is *precision*, *recall*, and *F-measure*. Precision is a metric that tells us what proportion of instances classified as positive, actually are positive. Recall is a metric that tells us what proportion of positive instances were classified as positive. Recall is the same metric as sensitivity. A good predictive model should have a good precision as well as a high recall. F-measure, also called F1 score, is a metric that combines both these aspects. F-measure is calculated by the harmonic mean of precision and recall. Precision, recall, and F-measure are defined in Eq. (4 – 6) respectively.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{4}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{5}$$

$$\boldsymbol{F - Measure} = \frac{2 * \boldsymbol{Precision} * \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}} \tag{6}$$

When running a hypothesis test in statistics, there are two types of errors. The first error occurs when a true null hypothesis is rejected. This error is known as *Type I error* which is the same as *false positive rate*. The second error occurs when a false null hypothesis is accepted. This error is known as *Type I error* which is the same as *false negative rate*. Type I and Type II errors are defined in Eq. (7 – 8).

$$\boldsymbol{Type\ I\ error} = \boldsymbol{false\ positive\ rate} = \frac{\boldsymbol{FP}}{\boldsymbol{FP+TN}} \tag{7}$$

$$\boldsymbol{Type\ II\ error} = \boldsymbol{false\ negative\ rate} = \frac{\boldsymbol{FN}}{\boldsymbol{FN+TP}} \tag{8}$$

An analogy in understanding the two types of error is to consider a defendant in a trial. The null hypothesis is "the defendant is not guilty", and the alternative is "the defendant is guilty”. A Type I error would correspond to convicting an innocent person, and a Type II error would correspond to setting a guilty person free.

The chances of committing these two types of errors are inversely proportional, that is, decreasing Type I error rate increases Type II error rate, and vice versa. Depending on your phenomenon of study, one type of error may be costlier than the other.

Select the evaluation metrics more suitable for your research. It is recommended to use accuracy as a first evaluation metric, and then evaluate the predictive models using a group of metrics like precision, recall, and F-measure.

b) Selection of predictive models

Calculate the evaluation metrics of every predictive model that you built. Select the best predictive model of each subject of each model scheme. Workbenches offer various evaluation metrics in their running environment. If you implemented the classifier algorithms in a programming language, the evaluation metrics can be easily computed in the same application.

For each model scheme, you may have several models from one subject. You will have *n x r* models where *n* is the number of datasets and *r* is the number of learning algorithms. From these *n x r* models, select the best model based on the evaluation metrics. Calculate the evaluation metrics’ average of the best models.

4.5 Comparison between CRISP-DM methodology and our proposed methodology

The CRISP-DM methodology is an open standard process model that describes common approaches used by experts in data mining, mainly in industry projects. The objective of CRISP-DM is to provide a number of steps that fits to general data mining applications. The life cycle of a data mining project in the CRISP-DM reference model consists of six phases which encompasses from the understanding of the project objectives and requirements from a business perspective to activities that go beyond of the creation of the model and the increment of knowledge of the data.

Our proposed method is a domain-specific methodology. The objective of our proposed methodology is to provide a number of steps for the development and evaluation of predictive models in healthcare domain. The methodology consists of three phases which encompasses from the definition of attributes and the development of data collection methods to the selection of the best predictive models. As our methodology is a domain-specific methodology, it provides methods, algorithms, techniques, etc., in more detail than CRISP-DM. Also, our methodology provides specific suggestions for the healthcare domain.

A comparison of the methodologies' phases can be done. Table 4-5 presents a summary of the correspondences among the phases of CRISP-DM and our proposed methodology.

Table 4-5 Summary of the correspondences among the phases of CRISP-DM and our proposed methodology

CRISP-DM	**Our proposed methodology**
1. Business understanding	1. Data collection
2. Data understanding	
3. Data preparation	2. Data pre-processing
4. Modeling	3. Predictive models' development
5. Evaluation	
6. Deployment	---

Comparing the two methodologies phases, we can observe that some phases are similar. The first phase of our methodology "Data collection", includes a study of the phenomenon of interest that can be corresponding to the phase 1 of CRISP-DM "Business understanding". "Data collection" also includes the activities described in the phase 2 of CRISP-DM "Data understanding".

The phase 2 of our methodology "Data pre-processing" and the phase 3 "Data preparation" of CRISP-DM are similar phases between these methodologies.

The phase 3 of our methodology "Predictive models' development" includes the activities described in the phases 3 and 4 of CRISP-DM "Modeling" and "Evaluation".

Chapter 5. Case study: stress recognition

5.1 Introduction

The main research goal of this thesis is to define a methodology for the development and evaluation of predictive models. In this chapter, our proposed methodology has been validated with a real case study: a predictive model for stress recognition.

5.2 Phase 1. Data collection

The phase 1 of the methodology is the data collection. The input of this phase is the phenomenon of interest which is stress, and the output is the data collected.

5.2.1 Selection of attributes of the study

We analyzed the nature of stress, identified the main characteristics, and the different types of stress. We searched the state of the art and identified the attributes used by other researchers. We also proposed new attributes. We wanted to select attributes that can be automatically measured by technology like wearables sensors and computers. We decided that the ground truth was given by a subject self-report. Table 5-1 shows the selected attributes.

Table 5-1 Selected attributes

Method	Attributes
Keyboard	Number of keyboard strokes
	Number of backspace keystrokes
	Number of delete keystrokes
	Number of enter keystrokes
	Number of spacebar keystrokes
Mouse	Number of mouse clicks
	Number of left button mouse clicks
	Number of right button mouse clicks
	Number of total scrolling
	Number of scrolls up wheeling
	Number of scrolls down wheeling
	Number of traversed pixels
Windows	Number of open windows
	Number of switching between windows
	Filtered windows title
Physical activity	Number of steps walked in a day
	Number of floors traversed in a day
	Distance traversed in a day
Heart rate	Heart rate minute by minute
Sleep	Number of minutes asleep
	Number of minutes of deep sleep
	Number of minutes of light sleep
	Number of minutes awake

5.2.2 Development of data collection methods

We decided to use automatic collection methods. Therefore, we developed *LaborCheck*, a software application that is capable of collecting data ubiquitously and continuously. *LaborCheck* consists of a desktop app and a web app. Figure 5.1 shows the *LaborCheck* architecture.

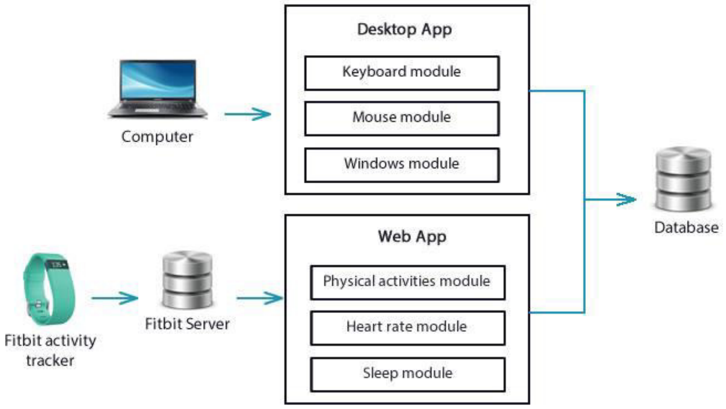
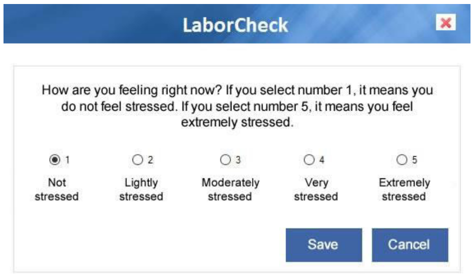


Figure 5.1 *LaborCheck* architecture

The *LaborCheck* desktop app automatically monitors the number and type of user computer interactions that an employee performs during their working day in real work settings. The monitored interactions are related to the keyboard, the mouse, and the windows of the applications used in the computer. The *LaborCheck* desktop app saves the window's title filtered by categories such as Job, Mail, Browser, Communication, and Personal. If it does not belong to any category, it saves No category.

The *LaborCheck* desktop app runs on the Windows operating system and is compatible with Windows 7, Windows 8 and Windows 10. The *LaborCheck* desktop app automatically monitors the employees' activity on a permanent basis. Data are collected every minute and they are temporally stored with a timestamp in a local database. The stored data are sent to our server database every 20 minutes via a web service.

In addition, the *LaborCheck* desktop app triggers a self-report survey of stress every 120-minutes in order to obtain the perceived stress level of the employee. The stress level is reported on a five-point scale where point one represents no stress and point five represents an extremely high-stress level. Figure 5.2 shows the stress self-report survey.



LaborCheck

How are you feeling right now? If you select number 1, it means you do not feel stressed. If you select number 5, it means you feel extremely stressed.

● 1 Not stressed
○ 2 Lightly stressed
○ 3 Moderately stressed
○ 4 Very stressed
○ 5 Extremely stressed

Figure 5.2 Stress self-report

The *LaborCheck* web app monitors employees' physical and physiological activity. To collect this type of data, we rely on a Fitbit® activity tracker. In order to have reliable data, employees have to use the activity tracker all day to monitor the heart rate and physical activity minute by minute, and all night to monitor sleep periods. Data from the Fitbit® activity tracker is stored locally. Afterward, data is synchronized to the Fitbit® server database through the official Fitbit® app. Every day, the *LaborCheck* web app pulls out the data through a Fitbit® API app and stores them in our server database. More information about *LaborCheck* can be found in [186].

5.2.3 Definition of the sampling

We used a non-probability sampling in which the study is restricted to a homogeneous subgroup of the population who have characteristics in common. The subjects must have the following profile:

a) Software developers whose work is focused mainly on a computer.
b) They should use the Windows operating system for working.
c) Ages from twenty and older.

5.2.4 Acquisition of data

We carried out two data collection processes. The first one was in Mexico and the second one was in Italy. We contacted software developers from different companies. We had a meeting with the subjects and explained them the goals and details of the study. Data were collected using the *LaborCheck* app and the Fitbit® activity tracker.

The subjects were told how to use the programs and devices included in the study. The subjects were advised about the data collected and their rights as subjects. They were informed that they could leave the process at any time if they wanted to. They were given a data privacy notice and they signed an informed consent form to participate in the study. The subjects filled in a demographic and work information questionnaire. The *LaborCheck* app was installed on their work computers; the Fitbit® app was installed on their smartphones, and they were given a Fitbit® activity tracker.

The subjects were required to wear the Fitbit® activity tracker all day and all night, except when they took a shower. They had to charge the activity tracker every two days during nonworking hours. They also had to synchronize the Fitbit® activity tracker daily. In addition, the subjects were asked to report their perceived stress. The self-report was displayed in their computers two or three times during the day depending on the working hours of each subject.

Since the study ran on a continuous basis, this period included working hours, mostly between 8 a.m. and 7 p.m. The subjects were asked to work as usual with their computers. In the first data collection process, the sample was constituted by 13 subjects (46% male and 54% female); with ages from 24 to 39 years old (average 28.9 ± 4.07); and 62% were single and 38% were married. The data collection process lasted 8 weeks. In the second data collection process, the sample was constituted by 8 subjects (50% male and 50% female); with ages from 19 to 23 years old (average 20.6 ± 1.3); and 100% were single. The data collection process lasted 4 weeks.

When the data collection processes were finished, the applications were uninstalled and the subjects returned the activity tracker.

5.3 Phase 2. Data preprocessing

Phase 2 of the methodology is data preprocessing. The input of this phase is the collected database and the output are *n* datasets. We applied techniques for data pre-processing in order to obtain high-quality mining results.

5.3.1 Data integration

We have compiled the raw data collected from the different methods: keyboard, mouse, windows, activity, heart rate, and sleep, including the stress levels self-reported by the subjects. The stress levels were classified into three stress categories: low, medium and high. Stress level 1 and 2 were classified as a low-stress category. Stress level 3 was classified as a medium-stress category, and stress levels four and five were classified as a high-stress category.

We generated instances by grouping data in 120-minutes time windows. Each instance was associated with a self-reported stress level.

5.3.2 Data cleaning

The database was analyzed in order to find inconsistencies. When we found several pieces of missing data in an instance, that instance was removed from the database. There were no data about sleep activity from one subject for several days in the data collection process. Therefore, data from this subject were not included in the analysis. However, when only one or two pieces of data were missing, these data were filled with the average attribute in the corresponding 120-minutes time window. As a result, we obtained 1737 instances from the 20 subjects.

5.3.3 Data transformation

We have created new attributes. We have calculated the sum, average, and standard deviation of numeric attributes in each time window. We have calculated the percentage of time spent in each software category. As a result, we obtained 60 attributes which are the following:

- From keyboard method, we have the sum, average and standard deviation of keys pressed, backspace keys pressed, delete keys pressed, enter keys pressed, and spacebar keys pressed; resulting in 15 attributes.
- From the mouse method, we have the sum, average and standard deviation of clicks, left clicks, right clicks, up scrolls, down scrolls, and traveled pixels; resulting in 21 attributes.
- From the windows method, we have the sum, average and standard deviation of open windows, and switching between windows. We also have one attribute for each software category; resulting in 12 attributes.
- From physical activity method, we have the sum, average and standard deviation of steps, floors, and distance traveled; resulting in 9 attributes.
- From the heart rate method, we have the average and standard deviation of heart rate; resulting in 2 attributes.
- From the sleep method, we have the sum of minutes of sleep, minutes of deep sleep, minutes of light sleep, and minutes awake; resulting in 4 attributes.

5.3.4 Class balancing

Our database presented an imbalance of classes. In order to tackle this problem, the data were resampled by applying the synthetic minority oversampling technique SMOTE [147]. Initially, we have obtained 1737 instances from the 20 subjects. After applying SMOTE, we have obtained 3362 instances. Table 5-2 shows the number of instances of each category before and after the stress level classification and the SMOTE technique.

Table 5-2 Stress category instances

Self-reported stress level	Stress category	Before SMOTE	After SMOTE
1, 2	Low	1028	1256
3	Medium	417	1248
4, 5	High	292	858
		1737	**3362**

The resulting database contained 3362 instances, 60 attributes, and a stress category. This database was called dataset DS01.

5.3.5 Attribute selection

Attribute selection was carried out through four attribute selection methods. We have created a dataset for each attribute selection method: dataset DS02 by Principal Component Analysis method; dataset DS03 by Correlation-based Feature Selection method; dataset DS04 by Information Gain method; and dataset DS05 by Wrappers for Feature Subset Selection method. In the end, we obtained five datasets: one dataset that includes all 60 attributes, and four datasets obtained from the attribute selection methods.

5.4 Phase 3. Predictive models' development

Phase 3 of the methodology is predictive models' development. The input of this phase are the *n* datasets from the previous phase, and the output is the predictive models.

5.4.1 – 5.4.2 Predictive models' construction and evaluation

The sub-goals of this research comprises three approaches: comparison of performance, comparison of stress recognition methods (wearable and computer), and comparison by country. We have carried out several experiments to achieve these sub-goals. Table 5-3 shows the experiments carried out.

Table 5-3 Experiments

Experiment	Approach	Scheme	Attributes	Subjects
1.1	Performance	Individual model	All	All
1.2	Performance	Similar subjects	All	All
1.3	Performance	General model	All	All
2.1	Method	Individual models	Wearable	All
2.2	Method	Individual models	Computer	All
2.3	Method	General models	Wearable	All
2.4	Method	General models	Computer	All
3.1	Country	General models	All	Mexican
3.2	Country	General models	All	Italian
3.3	Country	General models	All	Training set: Mexican Testing set: Italian
3.4	Country	General models	All	Training set: Italian Testing set: Mexican
4.1	Method and country	General models	Wearable	Mexican
4.2	Method and country	General models	Computer	Mexican
4.3	Method and country	General models	Wearable	Italian
4.4	Method and country	General models	Computer	Italian
4.5	Method and country	General models	Wearable	Training set: Mexican Testing set: Italian
4.6	Method and country	General models	Computer	Training set: Mexican Testing set: Italian
4.7	Method and country	General models	Wearable	Training set: Italian Testing set: Mexican
4.8	Method and country	General models	Computer	Training set: Italian Testing set: Mexican

We have applied six classifier methods in each experiment: Zero Rule as a baseline (ZR), *k*-Nearest Neighbor (*k*-NN), Naïve Bayes (NB), Random Forest (RF), C4.5 (C4.5), and AdaBoost (AB). We used the workbench Weka. We used ten times 10-fold cross-validation technique. We used accuracy, precision, recall, and F-Measure as evaluation metrics.

Performance approach

Experiment 1.1 – Individual model scheme

We have built models for each subject using their own data. The training sets were created by splitting each dataset so that we have data only from one subject. We obtained five training sets multiplied by six classifier methods multiplied by 20 subjects resulting in 600 models. Table 5-4 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-4 Best model of each subject – Individual model scheme

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	1	1	1	1	All datasets	NB, RF, C4.5, AB
I06	0.75	0.77	0.75	0.74	DS05	NB, RF
I07	0.93	0.95	0.94	0.94	DS05	NB
I08	0.91	0.93	0.92	0.91	DS05	RF
I09	0.81	0.85	0.81	0.80	DS05	*k*-NN
I10	0.86	0.82	0.87	0.83	DS05	C4.5
I11	0.96	0.97	0.97	0.96	DS05	RF
I13	0.95	0.96	0.95	0.95	DS05	RF
M01	0.76	0.78	0.76	0.75	DS01, DS05	RF, AB
M02	0.78	0.80	0.78	0.77	DS04, DS05	RF, AB
M03	0.60	0.63	0.60	0.69	DS05	NB
M04	0.97	0.98	0.98	0.98	DS01, DS04	RF
M05	0.91	0.93	0.91	0.91	DS05	NB
M06	0.90	0.91	0.90	0.90	DS03	RF
M07	1	1	1	1	All datasets	*k*-NN, NB, RF, C4.5, AB
M08	0.84	0.86	0.85	0.84	DS05	*k*-NN
M09	0.88	0.91	0.89	0.89	DS05	NB
M10	0.98	0.99	0.99	0.99	DS01, DS03, DS04, DS05	NB, RF
M11	0.73	0.97	0.73	0.73	DS05	*k*-NN
M13	1	1	1	1	DS01, DS03, DS04, DS05	NB, RF

The average of all individual predictive models' performance is an accuracy of 0.88; a precision of 0.9; a recall of 0.88; and an F-Measure of 0.88. The best attribute selection method was Wrappers for Feature Subset Selection method (DS05), and the best classifier method was Random Forest.

Experiment 1.2 – Similar subject scheme

We have built a model for a subject using data from a set of subjects with similar behavior. The behavior of each subject was represented by a single vector of size *a* x *c* where *a* is the number of attributes (60 attributes) and *c* is the number of every possible combination of stress level (3 combinations: low-medium, low-high, and medium-high). Therefore, the behavior vector of each subject has a size of 180. Subjects I05, I07, M07, M09, and M13 did not present data of high-stress level. The possible combination of only two stress levels was one, so the vectors of these subjects have a size of 60.

For each attribute, we analyzed how does the median value changes between the different pairs of stress levels. For example, for one subject the difference of the medians of an attribute between low and high-stress levels may be positive but for another subject, it may be negative.

The behavior vector is built by computing the difference of the medians between every pair of stress level for each attribute.

We used *k*-means clustering to define the set of subjects with similar behavior. The *k*-means algorithm requires to specify a number *k* of desired groups. To find a good approximation of *k* we used the silhouette index [187] which is a measure of the quality of the resulting groups. A value near to +1 indicates that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster. The *k*-means algorithm was run for $k = 2$ to 10, and the *k* that maximizes the silhouette index was chosen as the final number of groups. We have defined to which group each subject belongs by calculating the distance from the subject to the centroids and choosing the minor one. Table 5-5 shows the resulting *k*, the Silhouette score, and the set of similar subjects.

Table 5-5 Set of similar subjects of each subject

ID	*k*	Silhouette index	Set of similar subjects
I05	2	0.179	{I07, I10, I11, M01, M02, M07, M09, M11}
I06	2	0.240	{I08, I09, I13, M01, M02, M03, M04, M06, M08, M10, M11}
I07	4	0.201	{I08, I10, M01, M02, M03, M09, M11}
I08	3	0.250	{I06, I09, I13, M01, M02, M03, M04, M06, M10, M11}
I09	2	0.241	{I06, I08, I13, M01, M02, M03, M04, M06, M08, M10, M11}
I10	2	0.328	{I06, I08, I09, I11, I13, M01, M02, M03, M04, M06, M08, M10, M11}
I11	2	0.332	{I06, I08, I09, I10, I13, M01, M02, M03, M04, M06, M08, M10, M11}
I13	2	0.246	{I06, I08, I09, M01, M02, M03, M04, M06, M08, M10, M11}
M01	2	0.240	{I06, I08, I09, I13, M02, M03, M04, M06, M08, M10, M11}
M02	2	0.299	{I06, I08, I09, I10, I11, I13, M01, M03, M04, M06, M08, M10, M11}
M03	2	0.234	{I06, I08, I09, I13, M01, M02, M04, M06, M08, M10, M11}
M04	2	0.314	{I06, I08, I09, I10, I11, I13, M01, M02, M03, M06, M08, M10, M11}
M05	2	0.203	{I10, I11}
M06	2	0.238	{I06, I08, I09, I13, M01, M02, M03, M04, M08, M10, M11}
M07	3	0.204	{I07, I08, I10, M01, M02, M09, M11}
M08	2	0.269	{I06, I08, I09, I13, M01, M02, M03, M04, M06, M10, M11}
M09	3	0.271	{I06, I08, I09, I13, M01, M02, M03, M04, M05, M06, M08, M10, M13}
M10	2	0.244	{I06, I08, I09, I13, M01, M02, M03, M04, M06, M08, M11}
M11	2	0.254	{I06, I08, I09, I13, M01, M02, M03, M04, M06, M08, M10}
M13	3	0.201	{I06, I09, I13, M04, M05, M06, M08, M10}

We have built models for each subject using the leave-one-out technique. The training sets were created using data only from similar subjects. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by 20 subjects resulting in 600 models. Table 5-6 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-6 Best model of each subject – Similar subjects

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	0.88	0.89	0.89	0.88	DS04	RF
I06	0.44	0.42	0.44	0.42	DS01	AB
I07	0.63	0.76	0.64	0.60	DS01, DS04	RF
I08	0.57	0.58	0.58	0.57	DS01	RF
I09	0.45	0.49	0.46	0.44	DS02	C4.5
I10	0.45	0.63	0.46	0.41	DS03	*k*-NN
I11	0.32	0.35	0.32	0.33	DS01	RF
I13	0.59	0.58	0.6	0.58	DS02	RF
M01	0.47	0.45	0.48	0.46	DS04	AB
M02	0.33	0.33	0.34	0.33	DS02	*k*-NN
M03	0.47	0.48	0.48	0.47	DS02	*k*-NN
M04	0.48	0.46	0.48	0.45	DS01, DS04	*k*-NN
M05	0.36	0.30	0.37	0.30	DS01, DS04	C4.5
M06	0.49	0.49	0.50	0.49	DS02	*k*-NN
M07	0.66	0.89	0.66	0.69	DS02	NB
M08	0.60	0.57	0.60	0.56	DS03	AB
M09	0.57	0.68	0.57	0.62	DS02	*k*-NN
M10	0.47	0.48	0.48	0.46	DS01	AB
M11	0.49	0.49	0.49	0.49	DS03	AB
M13	0.38	0.46	0.38	0.83	DS02	NB

The average of all individual predictive models' performance is an accuracy of 0.50; a precision of 0.53; a recall of 0.51; and an F-Measure of 0.51. The best attribute selection method was PCA (DS02), and the best classifier method was *k*-NN.

Experiment 1.3 – General model scheme

We have built models for each subject using the leave-one-out technique. The training sets were created by removing the instances from subject *i* of each dataset. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by 20 subjects resulting in 600 models. Table 5-7 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-7 Best model of each subject – General model scheme

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	0.95	0.96	0.96	0.96	DS01, DS04	RF
I06	0.47	0.47	0.48	0.46	DS04	C4.5
I07	0.80	0.81	0.81	0.81	DS01	RF
I08	0.45	0.44	0.45	0.44	DS01, DS02, DS04, DS05	*k*-NN, NB, C4.5, AB
I09	0.47	0.48	0.48	0.42	DS02	C4.5
I10	0.52	0.54	0.53	0.52	DS03	*k*-NN
I11	0.75	0.79	0.75	0.71	DS01, DS02	RF
I13	0.77	0.78	0.78	0.77	DS04	RF
M01	0.55	0.56	0.56	0.55	DS04	AB
M02	0.42	0.41	0.42	0.41	DS03	C4.5
M03	0.44	0.49	0.45	0.37	DS02, DS04	*k*-NN, RF
M04	0.44	0.45	0.44	0.44	DS01, DS04	C4.5
M05	0.60	0.59	0.60	0.58	DS02	AB
M06	0.57	0.57	0.58	0.57	DS02	*k*-NN
M07	0.89	0.92	0.89	0.90	DS03	AB
M08	0.72	0.78	0.73	0.69	DS02	NB
M09	0.57	0.66	0.57	0.61	DS04	C4.5
M10	0.44	0.47	0.45	0.43	DS04	C4.5
M11	0.44	0.49	0.44	0.40	DS01, DS04	*k*-NN, AB
M13	0.61	0.65	0.61	0.60	DS03	AB

The average of all individual predictive models' performance is an accuracy of 0.59; a precision of 0.61; a recall of 0.59; and an F-Measure of 0.58. The best attribute selection method was Information Gain method (DS04), and the best classifier method was C4.5.

Summary

This section presents a comparison of the performance obtained by model schemes. Table 5-8 shows the summary of the results of performance approach.

Table 5-8 Summary of results of performance approach

Experiment	Model scheme	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
1.1	Individual	0.88	0.9	0.88	0.88	DS05	RF
1.2	Similar subject	0.50	0.53	0.51	0.51	DS02	*k*-NN
1.3	General	0.59	0.61	0.59	0.58	DS04	C4.5

Individual model scheme showed the best result with an F-measure of 0.88. This result is expected because the models are built and tested with the data of the same subject. General model scheme obtained an F-measure of 0.58, higher than similar subject scheme which obtained an F-measure of 0.51.

The best attribute selection method for individual scheme was Wrapper (DS05). The ten most frequently selected attributes (in descending order) were: number of minutes asleep, number of delete keystrokes, average of heart rate, time spent in No category window filter, standard deviation of traversed pixels, number of scrolls down wheeling, number of switching between windows, standard deviation of heart rate, time spent in Communication window filter, and standard deviation of distance.

The best attribute selection method for the similar subject scheme was PCA (DS02). We cannot see the selected attributes because PCA internally transforms the attributes in a combination of attributes.

The best attribute selection method for the general scheme was Info Gain (DS04). The first ten ranked attributes (in descending order) were: standard deviation of scrolls up wheeling, standard deviation of total scrolling, number of switching between windows, average of heart rate, standard deviation of scrolls down wheeling, time spent in No category window filter, average of switching between windows, number of steps, number of total scrolling, and average of total scrolling.

We have also compared the model schemes by analyzing the performance of each classifier. Figure 5.3 shows the average of the F-measure of the best models of each classifier. The performance of the classifiers was very similar. The classifiers obtained an F-measure around 0.50, except for NB which obtained a slightly lower F-measure of 0.42.

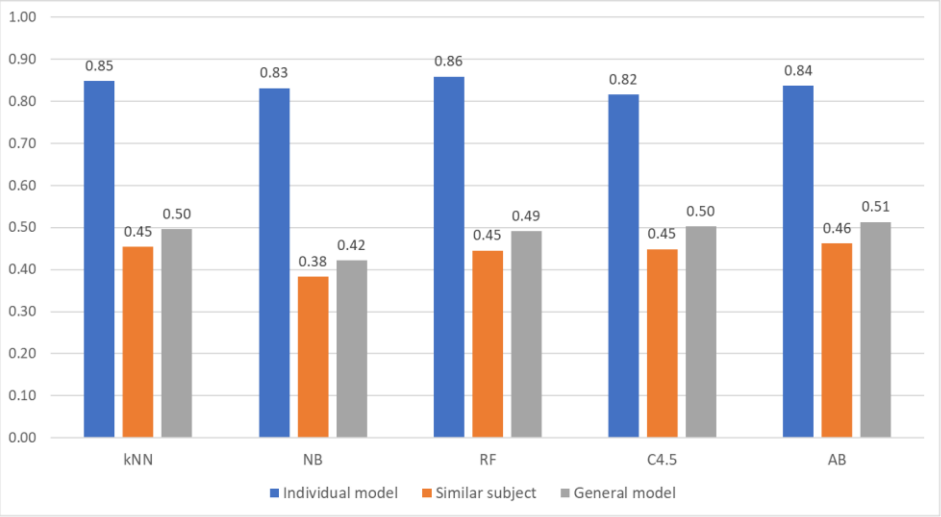


Figure 5.3 Comparison between model schemes

Method approach

Experiment 2.1 – Wearable method – Individual model scheme

We have built models for each subject using their own wearable data. The training sets were created by splitting each dataset so that we have data only from one subject, and removing the computer attributes. We obtained five training sets multiplied by six classifier methods multiplied by 20 subjects resulting in 600 models. Table 5-9 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-9 Best model of each subject – Wearable method

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	1	1	1	1	DS01, DS02, DS03, DS04	*k*-NN, NB, RF
I06	0.66	0.70	0.66	0.65	DS02	RF
I07	0.78	0.81	0.78	0.78	DS04	RF
I08	0.69	0.71	0.70	0.69	DS01, DS02, DS04	RF
I09	0.67	0.71	0.68	0.66	DS04	AB
I10	0.57	0.47	0.58	0.51	DS01, DS04	*k*-NN
I11	0.91	0.93	0.92	0.91	DS01	RF
I13	0.84	0.85	0.84	0.91	DS04	RF
M01	0.64	0.65	0.64	0.63	DS02	NB
M02	0.63	0.65	0.64	0.63	DS01	RF
M03	0.51	0.55	0.52	0.50	DS01	C4.5
M04	0.89	0.90	0.90	0.89	DS01	RF
M05	0.76	0.78	0.76	0.76	DS02	NB
M06	0.76	0.77	0.76	0.76	DS01	RF
M07	1	1	1	1	DS01, DS02, DS03, DS04	NB
M08	0.72	0.73	0.73	0.72	DS03	*k*-NN
M09	0.77	0.81	0.78	0.77	DS03	*k*-NN
M10	0.97	0.97	0.97	0.97	DS01, DS04	RF
M11	0.57	0.59	0.57	0.56	DS04	RF
M13	0.99	0.99	0.99	0.99	DS01, DS04	RF

The average of all individual predictive models' performance is an accuracy of 0.77; a precision of 0.77; a recall of 0.77; and an F-Measure of 0.76. The best attribute selection method was Information Gain method (DS04) but we also obtained the same results using all attributes (DS01). The best classifier method was Random Forest.

Experiment 2.2 – Computer method – Individual model scheme

We have built models for each subject using their own computer data. The training sets were created by splitting each dataset so that we have data only from one subject, and removing the wearable attributes. We obtained five training sets multiplied by six classifier methods multiplied by 20 subjects resulting in 600 models. Table 5-10 shows the best predictive model of each subject by means of the evaluation metrics.

The average of all individual predictive models' performance is an accuracy of 0.83; a precision of 0.84; a recall of 0.83; and an F-Measure of 0.82. The best attribute selection method was the Information Gain method (DS04), and the best classifier method was Random Forest.

Table 5-10 Best model of each subject – Computer method

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	1	1	1	1	DS01, DS02, DS03, DS04	NB, RF
I06	0.74	0.77	0.74	0.73	DS04	RF
I07	0.92	0.93	0.92	0.92	DS02, DS03	NB
I08	0.87	0.89	0.87	0.87	DS04	RF
I09	0.70	0.72	0.70	0.68	DS01, DS04	*k*-NN
I10	0.72	0.64	0.72	0.66	DS04	AB
I11	0.94	0.95	0.94	0.94	DS02	RF
I13	0.91	0.92	0.92	0.92	DS01, DS02	RF
M01	0.70	0.73	0.71	0.69	DS02	*k*-NN
M02	0.70	0.73	0.70	0.69	DS04	RF
M03	0.62	0.65	0.63	0.62	DS01	RF
M04	0.96	0.97	0.97	0.97	DS01, DS04	RF
M05	0.85	0.88	0.86	0.86	DS04	RF
M06	0.86	0.88	0.89	0.86	DS02	RF
M07	1	1	1	1	DS01, DS02, DS03, DS04	*k*-NN, NB, RF
M08	0.77	0.78	0.77	0.77	DS03	RF
M09	0.80	0.85	0.81	0.80	DS02	*k*-NN
M10	0.97	0.98	0.98	0.97	DS04	RF
M11	0.64	0.67	0.65	0.64	DS04	RF
M13	0.99	0.99	0.99	0.99	DS01, DS02, DS03, DS04	NB, RF

Experiment 2.3 – Wearable method – General model scheme

We have built models for each subject using only wearable data and the leave-one-out technique. The training sets were created by removing the instances from subject *i* of each dataset and removing the computer attributes. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by 20 subjects resulting in 600 models. Table 5-11 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-11 Best model of each subject – Wearable method

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	0.85	0.91	0.86	0.87	DS04	AB
I06	0.41	0.43	0.42	0.40	DS01, DS04	*k*-NN
I07	0.61	0.68	0.61	0.62	DS01	RF
I08	0.41	0.42	0.41	0.41	DS02	*k*-NN
I09	0.47	0.47	0.48	0.47	DS02	*k*-NN
I10	0.58	0.50	0.59	0.50	DS03	C4.5
I11	0.71	0.70	0.72	0.70	DS04	RF
I13	0.63	0.69	0.64	0.64	DS01	RF
M01	0.48	0.48	0.49	0.47	DS01, DS04	NB
M02	0.38	0.38	0.39	0.37	DS01, DS04	*k*-NN
M03	0.44	0.44	0.44	0.41	DS04	RF
M04	0.37	0.35	0.37	0.36	DS03	*k*-NN
M05	0.56	0.57	0.57	0.53	DS02	*k*-NN
M06	0.38	0.39	0.38	0.37	DS03	AB
M07	0.93	0.95	0.94	0.94	DS04	RF
M08	0.98	0.99	0.98	0.99	DS01	RF
M09	0.47	0.74	0.48	0.56	DS03	NB
M10	0.41	0.55	0.41	0.39	DS04	C4.5
M11	0.38	0.38	0.38	0.38	DS02	*k*-NN
M13	0.55	0.61	0.56	0.55	DS04	RF

The average of all individual predictive models' performance is an accuracy of 0.55; a precision of 0.58; a recall of 0.55; and an F-Measure of 0.54. The best attribute selection method was Information Gain method (DS04), and the best classifier methods were *k*-Nearest Neighbor and Random Forest.

Experiment 2.4 – Computer method – General model scheme

We have built models for each subject using only computer data and the leave-one-out technique. The training sets were created by removing the instances from subject *i* of each dataset and removing the wearable attributes. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by 20 subjects resulting in 600 models. Table 5-12 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-12 Best model of each subject – Computer method

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	0.91	0.93	0.91	0.91	DS01	RF
I06	0.47	0.48	0.48	0.48	DS02	*k*-NN
I07	0.80	0.81	0.81	0.80	DS01	RF
I08	0.45	0.47	0.46	0.45	DS02	C4.5
I09	0.67	0.66	0.67	0.67	DS02	C4.5
I10	0.61	0.62	0.62	0.60	DS02	C4.5
I11	0.73	0.73	0.73	0.72	DS02	C4.5, AB
I13	0.75	0.75	0.75	0.74	DS04	RF
M01	0.45	0.47	0.46	0.45	DS02	AB
M02	0.55	0.66	0.55	0.49	DS02	RF
M03	0.43	0.45	0.43	0.43	DS03	C4.5
M04	0.42	0.44	0.43	0.41	DS04	AB
M05	0.58	0.38	0.58	0.46	DS03	NB
M06	0.56	0.56	0.57	0.55	DS02	*k*-NN
M07	0.83	0.89	0.84	0.84	DS04	RF
M08	0.73	0.76	0.73	0.71	DS03	NB
M09	0.51	0.59	0.51	0.55	DS01, DS04	*k*-NN
M10	0.40	0.44	0.40	0.40	DS03	AB
M11	0.44	0.48	0.44	0.43	DS03	AB
M13	0.60	0.65	0.60	0.62	DS01	AB

The average of all individual predictive models' performance is an accuracy of 0.59; a precision of 0.61; a recall of 0.59; and an F-Measure of 0.58. The best attribute selection method was PCA (DS02), and the best classifier methods were C4.5 and AdaBoost.

Summary

This section presents the comparison of the stress recognition methods: wearable and computer. Table 5-13 shows the summary of the results of the method approach.

Table 5-13 Summary of results of the method approach

Exp	Model scheme	Method	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
2.1	Individual	Wearable	0.77	0.77	0.77	0.76	DS01, DS04	RF
2.2	Individual	Computer	0.83	0.84	0.83	0.82	DS04	RF
2.3	General	Wearable	0.55	0.58	0.55	0.54	DS04	*k*-NN, RF
2.4	General	Computer	0.59	0.61	.059	0.58	DS02	C4.5, AB

Individual models obtained better results than general models. This result is expected because the models are built and tested with the data of the same subject. In both individual and general model schemes, computer method showed better results than wearable methods. In the individual models, the computer method obtained an F-measure of 0.82, 6% higher than wearable method. In the general models, computer method obtained an F-measure of 0.58, 4% higher than wearable method.

The best attribute selection method for individual scheme and wearable method was Info Gain (DS04). The ten first ranked attributes (in descending order) were: average of heart rate, number of steps, the standard deviation of heart rate, the standard deviation of steps, number of minutes asleep, average of steps, distance traversed, the standard deviation of distance traversed, average of distance traversed, and number of floors.

The best attribute selection method for individual scheme and computer method was Info Gain (DS04). The ten first ranked attributes (in descending order) were: standard deviation of enter keystrokes, standard deviation of keyboard strokes, time spent in Mail window filter, standard deviation of traversed pixels, time spent in No category window filter, time spent in Communication window filter, number of scrolls down wheeling, number of spacebar keystrokes, number of enter keystrokes, and average of switching between windows.

The best attribute selection method for the general scheme and wearable method was Info Gain (DS04). The ten first ranked attributes (in descending order) were: average of heart rate, standard deviation of steps, standard deviation of distance traversed, distance traversed, number of steps, average of distance traversed, average of steps, number of minutes asleep, standard deviation of heart rate, and standard deviation of floors.

The best attribute selection method for the general scheme and computer method was PCA (DS02). We cannot see the selected attributes because PCA internally transforms the attributes in a combination of attributes.

Individual computer model obtained a better performance than the individual wearable model. But individual model using all the attributes obtained the best of the three models, with an F-measure of 0.88 (Experiment 1.1), 5% higher than the individual computer model, and 11% higher than the individual wearable model. Table 5-14 shows the results of individual models.

Table 5-14 Summary of results of individual models

Exp	Attributes	Accuracy	Precision	Recall	F-Measure
1.1	All	0.88	0.9	0.88	0.88
2.1	Wearable	0.77	0.77	0.77	0.76
2.2	Computer	0.83	0.84	0.83	0.82

General computer model obtained a better performance than the general wearable model and the same performance than the general model using all the attributes (Experiment 1.3). Table 5-15 shows the results of individual models.

Table 5-15 Summary of results of general models

Exp	Attributes	Accuracy	Precision	Recall	F-Measure
1.3	All	0.59	0.61	0.59	0.58
2.3	Wearable	0.55	0.58	0.55	0.54
2.4	Computer	0.59	0.61	.059	0.58

Country approach

Experiment 3.1 – Mexican

We have built models for each subject using only wearable data from Mexican subjects, and the leave-one-out technique. The training sets were created by removing the instances from subject *i* of each dataset and removing the computer attributes. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by 12 Mexican subjects resulting in 360 models. Table 5-16 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-16 Best model of each subject – All attributes – Mexican

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
M01	0.49	0.50	0.49	0.48	DS01	AB
M02	0.40	0.43	0.40	0.39	DS01	C4.5
M03	0.42	0.42	0.43	0.41	DS02	C4.5
M04	0.52	0.53	0.53	0.52	DS02	*k*-NN
M05	0.69	0.67	0.69	0.67	DS02	C4.5
M06	0.43	0.44	0.44	0.41	DS04	C4.5
M07	0.91	0.95	0.92	0.93	DS03	RF
M08	0.64	0.63	0.64	0.63	DS02	*k*-NN
M09	0.53	0.66	0.54	0.59	DS02	*k*-NN
M10	0.48	0.5	0.48	0.48	DS01	*k*-NN
M11	0.45	0.44	0.45	0.43	DS03	*k*-NN
M13	0.71	0.76	0.72	0.72	DS03	RF

The average of all individual predictive models' performance is an accuracy of 0.55; a precision of 0.57; a recall of 0.56; and an F-Measure of 0.55. The best attribute selection method was PCA (DS02). The best classifier method was *k*-Nearest Neighbor.

Experiment 3.2 – Italian

We have built models for each subject using only wearable data from Italian subjects and the leave-one-out technique. The training sets were created by removing the instances from subject *i* of each dataset and removing the computer attributes. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by eight Italian subjects resulting in 240 models. Table 5-17 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-17 Best model of each subject – All attributes – Italian

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	0.85	0.89	0.85	0.85	DS03	C4.5
I06	0.45	0.45	0.46	0.44	DS01	C4.5
I07	0.72	0.72	0.72	0.72	DS03	AB
I08	0.47	0.53	0.48	0.43	DS03	C4.5
I09	0.43	0.46	0.43	0.4	DS02	*k*-NN
I10	0.56	0.64	0.57	0.57	DS03	NB
I11	0.53	0.68	0.54	0.5	DS01	NB
I13	0.54	0.53	0.55	0.53	DS02	C4.5

The average of all individual predictive models' performance is an accuracy of 0.56; a precision of 0.61; a recall of 0.57; and an F-Measure of 0.55. The best attribute selection method was Correlation (DS03), and the best classifier method was C4.5.

Experiment 3.3 – Italian subjects in Mexican models

We have built models using only data from Mexican subjects. The training sets were created by removing the instances from Italian subjects of each dataset. Each testing set included the instances from one Italian subject. We obtained five training sets multiplied by six classifier methods multiplied by eight Italian subjects resulting in 240 models. Table 5-18 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-18 Best models built using Mexican data and tested in each Italian subject

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	0.69	0.87	0.70	0.71	DS02	*k*-NN
I06	0.52	0.51	0.53	0.51	DS01, DS04	*k*-NN
I07	0.79	0.85	0.8	0.82	DS03	RF
I08	0.47	0.48	0.47	0.46	DS04	RF
I09	0.39	0.50	0.40	0.36	DS04	AB
I10	0.54	0.62	0.54	0.53	DS01, DS04	*k*-NN
I11	0.85	0.86	0.86	0.85	DS04	RF
I13	0.65	0.66	0.66	0.63	DS03	RF

The average of all individual predictive models' performance is an accuracy of 0.61; a precision of 0.66; a recall of 0.62; and an F-Measure of 0.6. The best attribute selection method was Info Gain (DS04), and the best classifier method was Random Forest.

Experiment 3.4 – Mexican subjects in Italian models

We have built models using only data from Italian subjects. The training sets were created by removing the instances from Mexican subjects of each dataset. Each testing set included the instances from one Mexican subject. We obtained five training sets multiplied by six classifier methods multiplied by 12 Mexican subjects resulting in 360 models. Table 5-19 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-19 Best models built using Italian data and tested in each Mexican subject

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
M01	0.46	0.65	0.46	0.37	DS01	AB
M02	0.35	0.38	0.36	0.33	DS01, DS04	*k*-NN
M03	0.41	0.42	0.42	0.40	DS01	RF
M04	0.55	0.53	0.56	0.49	DS02	C4.5
M05	0.73	0.72	0.73	0.72	DS02	*k*-NN
M06	0.48	0.48	0.49	0.48	DS02	*k*-NN
M07	0.76	0.84	0.76	0.75	DS03	AB
M08	0.38	0.35	0.39	0.33	DS03	C4.5
M09	0.60	0.67	0.60	0.59	DS02	*k*-NN
M10	0.69	0.68	0.69	0.68	DS03	NB
M11	0.45	0.46	0.46	0.46	DS04	C4.5
M13	0.75	0.82	0.75	0.75	DS03	*k*-NN

The average of all individual predictive models' performance is an accuracy of 0.55; a precision of 0.58; a recall of 0.55; and an F-Measure of 0.52. The best attribute selection methods were PCA (DS02) and Correlation (DS03), and the best classifier method was *k*-Nearest-Neighbor.

Summary

This section presents the comparison of stress recognition by country. Table 5-20 shows the summary of the results of country approach.

Table 5-20 Summary of results of the country approach

Exp	Training set	Testing set	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
3.1	Mexican	Mexican	0.55	0.57	0.56	0.55	DS02	*k*-NN
3.2	Italian	Italian	0.56	0.61	0.57	0.55	DS03	C4.5
3.3	Mexican	Italian	0.61	0.66	0.62	0.60	DS04	RF
3.4	Italian	Mexican	0.55	0.58	0.55	0.52	DS02, DS03	*k*-NN

Mexican model tested in Mexican subjects obtained an F-measure of 0.55, and Italian model tested in Italian subjects also obtained an F-measure of 0.55. Mexican model tested in Italian subjects obtained an F-measure of 0.60, whereas Italian model tested in Mexican subjects obtained an F-measure of 0.52. The Mexican model performance was slightly higher than the Italian. The average of these four results is 0.555, which is very similar to the results obtained by general models (Experiment 1.3, F-measure = 0.58).

Method and country approach

Experiment 4.1 – Wearable method – Mexican

We have built models for each subject using only wearable data from Mexican subjects and the leave-one-out technique. The training sets were created by removing the instances from subject *i* of each dataset and removing the computer attributes. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by 12 Mexican subjects resulting in 360 models. Table 5-21 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-21 Best model of each subject – Wearable method – Mexican

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
M01	0.45	0.52	0.45	0.45	DS01, DS04	NB
M02	0.33	0.32	0.34	0.32	DS01, DS04	*k*-NN
M03	0.44	0.45	0.44	0.44	DS01, DS04	*k*-NN
M04	0.61	0.64	0.62	0.62	DS01	RF
M05	0.63	0.63	0.63	0.60	DS02	RF, C4.5
M06	0.48	0.53	0.48	0.45	DS04	RF
M07	0.97	0.99	0.98	0.98	DS04	RF
M08	0.69	0.69	0.70	0.69	DS02	AB
M09	0.61	0.83	0.62	0.60	DS03	NB
M10	0.75	0.76	0.76	0.75	DS01	AB
M11	0.41	0.40	0.41	0.39	DS03	C4.5
M13	0.84	0.91	0.85	0.86	DS01	RF

The average of all individual predictive models' performance is an accuracy of 0.60; a precision of 0.63; a recall of 0.6; and an F-Measure of 0.59. The best attribute selection method was Info Gain (DS04) but we also obtained the same results using all attributes (DS01). The best classifier method was Random Forest.

Experiment 4.2 – Computer method – Mexican

We have built models for each subject using only computer data from Mexican subjects and the leave-one-out technique. The training sets were created by removing the instances from subject *i* of each dataset and removing the wearable attributes. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by 12 Mexican subjects resulting in 360 models. Table 5-22 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-22 Best model of each subject – Computer method – Mexican

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
M01	0.57	0.58	0.57	0.52	DS01	RF
M02	0.44	0.47	0.45	0.43	DS02	*k*-NN
M03	0.51	0.53	0.52	0.49	DS04	AB
M04	0.70	0.75	0.71	0.69	DS01	RF
M05	0.78	0.86	0.78	0.77	DS03	RF
M06	0.68	0.70	0.68	0.67	DS01, DS04	*k*-NN
M07	0.96	0.97	0.97	0.97	DS04	RF
M08	0.78	0.84	0.78	0.76	DS03	NB
M09	0.56	0.60	0.56	0.56	DS01	RF
M10	0.89	0.90	0.90	0.90	DS01, DS03	RF
M11	0.50	0.62	0.51	0.48	DS02	AB
M13	0.95	0.96	0.96	0.96	DS04	RF

The average of all individual predictive models' performance is an accuracy of 0.69; a precision of 0.73; a recall of 0.69; and an F-Measure of 0.68. The best attribute selection method was Info Gain (DS04) but we also obtained the same results using all attributes (DS01). The best classifier method was Random Forest.

Experiment 4.3 – Wearable method – Italian

We have built models for each subject using only wearable data from Italian subjects and the leave-one-out technique. The training sets were created by removing the instances from subject *i* of each dataset and removing the computer attributes. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by eight Italian subjects resulting in 240 models. Table 5-23 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-23 Best model of each subject – Wearable method – Italian

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	0.95	0.95	0.95	0.95	DS01, DS03, DS04	RF, C4.5
I06	0.63	0.62	0.63	0.62	DS01	RF
I07	0.90	0.95	0.90	0.92	DS04	RF
I08	0.66	0.68	0.66	0.66	DS02	RF
I09	0.65	0.69	0.65	0.65	DS04	C4.5
I10	0.70	0.63	0.70	0.63	DS03	C4.5
I11	0.67	0.67	0.68	0.67	DS04	AB
I13	0.88	0.89	0.88	0.88	DS01	RF

The average of all individual predictive models' performance is an accuracy of 0.75; a precision of 0.76; a recall of 0.75; and an F-Measure of 0.74. The best attribute selection method was Info Gain (DS04), and the best classifier method was Random Forest.

Experiment 4.4 – Computer method – Italian

We have built models for each subject using only computer data from Italian subjects and the leave-one-out technique. The training sets were created by removing the instances from subject *i* of each dataset and removing the wearable attributes. The testing set included the instances from subject *i*. We obtained five training sets multiplied by six classifier methods multiplied by eight Italian subjects resulting in 240 models. Table 5-24 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-24 Best model of each subject – Computer method – Italian

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	1	1	1	1	DS01, DS02, DS03, DS04	RF, C4.5, AB
I06	0.66	0.67	0.67	0.67	DS01, DS04	RF
I07	0.85	0.85	0.85	0.85	DS03	AB
I08	0.79	0.80	0.80	0.80	DS01, DS04	*k*-NN
I09	0.75	0.75	0.75	0.75	DS01, DS04	*k*-NN
I10	0.80	0.80	0.80	0.80	DS01	AB
I11	0.92	0.93	0.93	0.92	DS01, DS04	*k*-NN
I13	0.91	0.92	0.91	0.91	DS04	RF

The average of all individual predictive models' performance is an accuracy of 0.83; a precision of 0.84; a recall of 0.83; and an F-Measure of 0.83. The best attribute selection method was Info Gain (DS04) but we also obtained the same results using all attributes (DS01). The best classifier methods were k-Nearest Neighbor and Random Forest.

Experiment 4.5 – Wearable method – Italian subjects in Mexican models

We have built models using only wearable data from Mexican subjects. The training sets were created by removing the instances from Italian subjects and removing the computer data of each dataset. Each testing set included the instances from one Italian subject. We obtained five training sets multiplied by six classifier methods multiplied by eight Italian subjects resulting in 240 models. Table 5-25 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-25 Best model built using Mexican wearable data and tested in each Italian subject

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	0.76	0.86	0.76	0.76	DS02	C4.5
I06	0.44	0.46	0.44	0.43	DS03	*k*-NN
I07	0.60	0.63	0.61	0.61	DS03	C4.5
I08	0.45	0.47	0.46	0.46	DS02	*k*-NN
I09	0.46	0.60	0.47	0.42	DS03	C4.5
I10	0.54	0.55	0.54	0.54	DS02	NB
I11	0.72	0.72	0.73	0.72	DS02	RF
I13	0.60	0.63	0.60	0.61	DS03	AB

The average of all individual predictive models' performance is an accuracy of 0.57; a precision of 0.61; a recall of 0.57; and an F-Measure of 0.56. The best attribute selection methods were PCA (DS02) and Correlation (DS03), and the best classifier methods were k-Nearest Neighbor and C4.5.

Experiment 4.6 – Computer method – Italian subjects in Mexican models

We have built models using only computer data from Mexican subjects. The training sets were created by removing the instances from Italian subjects and removing the wearable data of each dataset. Each testing set included the instances from one Italian subject. We obtained five training sets multiplied by six classifier methods multiplied by eight Italian subjects resulting in 240 models. Table 5-26 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-26 Best models built using Mexican computer data and tested in each Italian subject

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
I05	0.80	0.89	0.81	0.82	DS03	*k*-NN
I06	0.75	0.77	0.75	0.75	DS02	RF
I07	0.75	0.77	0.75	0.75	DS02	RF
I08	0.75	0.77	0.75	0.75	DS02	RF
I09	0.45	0.62	0.45	0.42	DS01	AB
I10	0.54	0.57	0.54	0.54	DS01, DS04	*k*-NN
I11	0.80	0.80	0.81	0.80	DS04	RF
I13	0.66	0.66	0.67	0.66	DS02	RF

The average of all individual predictive models' performance is an accuracy of 0.69; a precision of 0.73; a recall of 0.69; and an F-Measure of 0.68. The best attribute selection method was PCA (DS02), and the best classifier method was Random Forest.

Experiment 4.7 – Wearable method – Mexican subjects in Italian models

We have built models using only wearable data from Italian subjects. The training sets were created by removing the instances from Mexican subjects and removing the computer data of each dataset. Each testing set included the instances from one Mexican subject. We obtained five training sets multiplied by six classifier methods multiplied by 12 Mexican subjects resulting in 360 models. Table 5-27 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-27 Best models built using Italian wearable data and tested in each Mexican subject

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
M01	0.47	0.47	0.47	0.43	DS03	NB
M02	0.36	0.35	0.36	0.35	DS01, DS03, DS04	NB
M03	0.44	0.44	0.44	0.44	DS02	*k*-NN
M04	0.46	0.47	0.47	0.47	DS04	C4.5
M05	0.61	0.58	0.61	0.59	DS02	AB
M06	0.37	0.37	0.38	0.37	DS03	C4.5
M07	0.96	0.97	0.97	0.97	DS03, DS04	RF, C4.5
M08	0.38	0.43	0.39	0.33	DS04	AB
M09	0.39	0.64	0.40	0.49	DS02	AB
M10	0.74	0.75	0.75	0.74	DS04	AB
M11	0.42	0.42	0.43	0.40	DS04	AB
M13	0.86	0.90	0.87	0.88	DS04	RF

The average of all individual predictive models' performance is an accuracy of 0.54; a precision of 0.56; a recall of 0.54; and an F-Measure of 0.53. The best attribute selection method was Info Gain (DS04), and the best classifier method was AdaBoost.

Experiment 4.8 – Computer method – Mexican subjects in Italian models

We have built models using only computer data from Italian subjects. The training sets were created by removing the instances from Mexican subjects and removing the wearable data of each dataset. Each testing set included the instances from one Mexican subject. We obtained five training sets multiplied by six classifier methods multiplied by 12 Mexican subjects resulting in 360 models. Table 5-28 shows the best predictive model of each subject by means of the evaluation metrics.

Table 5-28 Best models built using Italian computer data and tested in each Mexican subject

ID	Accuracy	Precision	Recall	F-Measure	Dataset	Classifier
M01	0.41	0.42	0.41	0.39	DS03	C4.5
M02	0.37	0.38	0.38	0.38	DS01	AB
M03	0.42	0.45	0.43	0.41	DS01	AB
M04	0.43	0.43	0.43	0.40	DS01	C4.5
M05	0.69	0.68	0.70	0.69	DS02	*k*-NN
M06	0.58	0.58	0.58	0.55	DS02	*k*-NN
M07	0.73	0.83	0.74	0.72	DS03	RF
M08	0.36	0.33	0.37	0.32	DS01	AB
M09	0.62	0.67	0.63	0.61	DS02	*k*-NN
M10	0.63	0.46	0.63	0.52	DS03	RF
M11	0.41	0.51	0.42	0.33	DS04	RF
M13	0.74	0.82	0.74	0.72	DS03	*k*-NN

The average of all individual predictive models' performance is an accuracy of 0.53; a precision of 0.54; a recall of 0.53; and an F-Measure of 0.5. The best attribute selection method was Correlation (DS03) but we also obtained the same results using all attributes (DS01). The best classifier method was k-Nearest Neighbor.

Summary

This section presents the comparison of the stress recognition by the method and by country. Table 5-29 shows the summary of the results of method and country approach.

Table 5-29 Summary of results of method and country approach

Exp	Method	Training set	Testing set	Accuracy	Precision	Recall	F-Measure
4.1	Wearable	Mexican	Mexican	0.60	0.63	0.60	0.59
4.2	Computer	Mexican	Mexican	0.69	0.73	0.69	0.68
4.3	Wearable	Italian	Italian	0.75	0.76	0.75	0.74
4.4	Computer	Italian	Italian	0.83	0.84	0.83	0.83
4.5	Wearable	Mexican	Italian	0.57	0.61	0.57	0.56
4.6	Computer	Mexican	Italian	0.69	0.73	0.69	0.68
4.7	Wearable	Italian	Mexican	0.54	0.56	0.54	0.53
4.8	Computer	Italian	Mexican	0.53	0.54	0.53	0.50

In general, computer models obtained better results than wearable models. In the models trained and tested with Mexican subjects, the computer model obtained an F-measure of 0.68, 9% better than the wearable model. In the models trained and tested with Italian subjects, the computer model obtained an F-measure of 0.83, and also 9% better than the wearable model.

These experiments obtained the best performances of all experiments. Table 5-30 shows the results of models trained and tested with Mexican subjects.

Table 5-30 Summary of results of models trained and tested with Mexican

Exp	Attributes	Accuracy	Precision	Recall	F-Measure
3.1	All	0.55	0.57	0.56	0.55
4.1	Wearable	0.60	0.63	0.60	0.59
4.2	Computer	0.69	0.73	0.69	0.68

The computer model obtained the best performance with an F-measure of 0.68, 9% higher than the wearable model, and 13% higher than the model with all attributes.

Table 5-31 shows the results of models trained and tested with Italian subjects.

Table 5-31 Summary of results of models trained and tested with Italian

Exp	Attributes	Accuracy	Precision	Recall	F-Measure
3.2	All	0.56	0.61	0.57	0.55
4.3	Wearable	0.75	0.76	0.75	0.74
4.4	Computer	0.83	0.84	0.83	0.83

The computer model obtained the best performance with an F-measure of 0.83, 9% higher than the wearable model, and 28% higher than the model with all attributes.

In the models trained with Mexican subjects and tested with Italian subjects, the computer model obtained an F-measure of 0.68, 12% better than the wearable model. Whereas in the models trained with Italian subjects and tested with Mexican subjects, the wearable model obtained an F-measure of 0.53, 3% better than the computer model.

Chapter 6. Conclusions

In this research, we have dealt with the recognition of stress levels in a certain period of time in software developers using computational methods. We have defined a methodology for the developing and evaluation of predictive models and we have validated it through a case study on stress recognition. We have developed a data mining application using computational methods and following the proposed methodology. We have obtained the stress predictive models and it is feasible to recognize stress levels of a software developers in a certain period of time. We have extended the CRISP-DM methodology for data mining applications on stress recognition. We have compared the two methodologies and we found similarities between the methodologies' phases.

We have used different schemes to build the predictive models: individual model, similar subject, and general model. Individual model scheme showed the best result with an F-measure of 0.88. General model scheme obtained an F-measure of 0.58, higher than the similar subject scheme which obtained an F-measure of 0.51. The most frequent attributes were: average of heart rate, number of switching between windows, and time spent in No category window filter. We have also compared the model schemes by analyzing the performance of each classifier. The classifiers obtained an F-measure around 0.50, except for NB which obtained a slightly lower F-measure of 0.42.

We have built individual and general predictive models using wearable and computer data separately to find out which method was more convenient for stress recognition. In both individual and general model schemes, computer method showed better results than wearable methods. In the individual models, the computer method obtained an F-measure of 0.82, 6% higher than wearable method. But individual model using all the attributes obtained the best of the three models, with an F-measure of 0.88, 5% higher than the individual computer model, and 11% higher than the individual wearable model. The reason may be that in individual models, the more attributes, the better the performance. In the general models, computer method obtained an F-measure of 0.58, 4% higher than wearable method. General model using all attributes also obtained an F-measure of 0.58.

We have built predictive models by country, that is to say, we built a model using Mexican data, and another model using Italian data. We have tested the models with Italian and Mexican data separately. Mexican model tested in Mexican subjects obtained an F-measure of 0.55, and Italian model tested in Italian subjects also obtained an F-measure of 0.55. Mexican model tested in Italian subjects obtained an F-measure of 0.60, whereas Italian model tested in Mexican subjects obtained an F-measure of 0.52. The Mexican model performance was slightly higher than the Italian. The reason for this result may be because there were more Mexican subjects (12) than Italian subjects (8), so the Mexican model had more training data. The average of these four results is 0.555, which is very similar to the results obtained by the general model using all subjects (F-measure = 0.58).

Furthermore, we have built predictive models by the method and by country, that is to say, we built a model using wearable Mexican data and another model using computer Mexican data. We did the same with Italian data. We tested the models with Italian and Mexican data separately. In the models trained and tested with Mexican subjects, the computer model obtained an F-measure of 0.68, 9% better than the wearable model, and 13% higher than the model with all attributes. In the models trained and tested with Italian subjects, the computer model obtained an F-measure of 0.83, and also 9% better than the wearable model, and 28% higher than the model with all attributes. In the models trained with Mexican subjects and tested with Italian subjects, the computer model obtained an F-measure of 0.68, 12% better than the wearable model. Whereas in the models trained with Italian subjects and tested with Mexican subjects, the wearable model obtained an F-measure of 0.53, 3% better than the computer model.

Individual model scheme showed the best result with an F-measure of 0.88 which is expected because the models are trained and tested with the same data using cross-validation technique. Similar subject and general models obtained an F-measure of 0.51 and 0.58 respectively. The similar subject scheme did not present any improvement over the general model scheme. The reason for this lack of improvement may be the poor quality of the clusters whose Silhouette indexes were lower than 0.332.

Computer method showed to be more convenient for stress recognition than wearable method. Only in one experiment, the wearable model resulted better than the computer one. Besides, the model built using the only computer attributes obtained the same F-measure of 0.58 as the model built using all attributes. Hence, the computer method may be used for stress recognition without losing precision and recall. Computer attributes have many advantages for stress recognition in office jobs. They are non-intrusive, cheap, and easy to monitor. Many more computer attributes could be monitored, and several more could be calculated. However, wearables should not be dismissed. Wearable is becoming more intelligent and sophisticated, and its monitoring accuracy is improving constantly.

The predictive models built by country using all attributes obtained an average of the F-measure of 0.555. This performance is similar to the general model performance of 0.58. According to these results, it seems that models built with data from subjects of one country may be used for stress recognition in subjects from another country. But when the attributes were separated by the method, the results were different. Models built and tested with computer data from one country showed better performance. Mexican models built using wearable and computer attributes obtained an F-measure of 0.59 and 0.68 respectively. Italian models built using wearable and computer attributes obtained an F-measure of 0.74 and 0.83 respectively. According to these results, there is a difference in the behavior of subjects from different countries when it is analyzed by the method.

The models trained with Mexican data and tested in Italian subjects showed a high performance with an F-measure of 0.60 when using all attributes, and an F-measure of 0.68 when using computer attributes. The reason for this improvement may be that there were more Mexican subjects than Italian subjects, so the Mexican models had more training data.

6.1 Contributions

The main contribution of this research is the definition of a methodology for the development and evaluation of predictive models. The methodology is an extension of CRISP-DM methodology for data mining applications for stress recognition.

The second contribution is the evaluation of two novel methods for stress recognition: wearable-based method and computer-based method. These methods are convenient for stress recognition on software developers. It is very important to recognize stress at earlier steps in order to carry out interventions that allow the software developers to overcome this condition and improve their quality of life.

The third contribution is the evaluation of three different models' schemes: individual model, general model, and similar users' model. Most works only have done general models to recognize stress. In this research, we determine the best scheme suitable for software developers' domain.

Finally, the fourth contribution is the evaluation of models created using data from users of two countries. This evaluation allowed us to find out and to analyze the differences between the behavior of software developers from different countries.

6.2 Future work

Some possible research future directions are:

- To validate the proposed methodology in other domains.
- To explore more the similar subject model scheme by testing other clustering algorithms and other quality indexes in order to improve the quality of the clusters.
- To explore the use of other wearable devices in order to increase the precision of the monitored attributes.
- To collect data using other stress recognition methods in order to identify the best methods to recognize stress, not only on desk job but in other contexts.
- To collect data from people of other countries in order to explore what happens with models trained and tested with data from the same country, and to generalize our results.
- To extend the data collection duration.
- To explore the use of more classification algorithms such as neural networks and support vector machine.

6.3 Publications

As a result of the knowledge acquired during the development of this research, two articles were generated in indexed journals, one article in an international congress, and three in national congresses. The articles focused on ambient intelligence and predictive modeling for stress recognition. In addition, two more articles were generated in indexed journals, one in an indexed journal by Scopus and CONACYT, and one in a national congress. These articles focused on other researches such as loneliness and social isolation recognition in older adults, and emotions recognition.

Stress recognition

1. W. Sanchez, A. Martinez, Y. Hernandez, H. Estrada, and M. Gonzalez-Mendoza, *A predictive model for stress recognition in desk jobs*, Journal of Ambient Intelligence Humanized Computing, p. 1-13, 2018, IF: 1.423 https://doi.org/10.1007/s12652-018-1149-9
2. Martínez, A., Sánchez, W., Benítez, R., González, Y., Mejía, M., & Ortiz, J. (2018). *A Job Stress Predictive Model Evaluation Through Classifiers Algorithms.* IEEE Latin American Transactions, vol. 16, no. 1, pp. 178–185, IF: 0.631. http://doi.org/10.1109/TLA.2018.8291471
3. Sánchez, W., Martínez, A., & González M. (2017). *Towards job stress recognition based on behavior and physiological features*. In: Ochoa S., Singh P., Bravo J. (eds) Ubiquitous Computing and Ambient Intelligence. UCAmI 2017. Lecture Notes in Computer Science, vol. 10586, pp. 311–322. Springer, Cham. https://doi.org/10.1007/978-3-319-67585-5_33
4. Camaños M., Sánchez, W., Martínez A., & Mejía-Lavalle M. (2016). *LaborCheck: Sistema de Monitoreo Automático de Variables Usuario-Computadora y de Interacción Social.* Encuentro Nacional de Ciencias de la Computación – Taller de Computación Clínica e Informática Médica. Chihuahua, México.
5. Martínez, A., Sánchez, W., González, Y., Estrada, H., & Zavala, C. (2015). *Correlación del estrés laboral con la actividad e inactividad física de los trabajadores.* Encuentro Nacional de Ciencias de la Computación – Taller de Computación Clínica e Informática Médica. Ensenada, México.
6. Sánchez, W., & Martínez, A. (2015). *Prevención del estrés laboral a partir de las interacciones humano-computadora y factores fisiológicos*. Encuentro Nacional de Ciencias de la Computación – Consorcio de Posgrado. Ensenada, México.

Other researches

1. Campos, W., Martínez, A., Sánchez, W., Estrada, H., Castro-Sánchez, N., & Mújica, D. (2016). A Systematic Review of Proposals for the Social Integration of Elderly People Using Ambient Intelligence and Social Networking Sites. Cognitive Computation, 8(3), 529–542, IF: 1.933. http://doi.org/10.1007/s12559-016-9382-z

2. Campos, W., Martínez, A., Sánchez, W., Estrada, H., Favela, J., & Pérez, J. (2016). Inferring social isolation in older adults through Ambient Intelligence and Social Networking Sites. Computing and Systems, 20(1), 143–152, IF: 0.1204. http://doi.org/10.13053/CyS-20-1-2193
3. Sánchez, W., Martínez, A., Campos, W., Estrada, H., & Pelechano, V. (2015). Inferring loneliness levels in older adults from smartphones. Journal of Ambient Intelligence and Smart Environments, 7(1), 85–98, IF: 1.063. http://doi.org/10.3233/AIS-140297
4. Molina, A., Martínez, A., & Sánchez, W. (2014). Emotion-Bracelet: Una interfaz emocional. Encuentro Nacional de Ciencias de la Computación – Taller de Computación Clínica e Informática Médica. Ocotlán, México.

References

[1] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY: Springer New York, 2013.

[2] S. Geisser, *Predictive Inference: An Introduction*. Chapman and Hall, 1993.

[3] A. H. Al-Fatlawi, H. K. Fatlawi, and S. H. Ling, “Recognition physical activities with optimal number of wearable sensors using data mining algorithms and deep belief network,” in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, pp. 2871–2874.

[4] S. Leka, A. Griffiths, and T. Cox, “La organización del trabajo y el estrés: estrategias sistemáticas de solución de problemas para empleadores, personal directivo y representantes sindicales,” *Paris, Fr. Organ. Mundal la Salud*, 2004.

[5] OMS, “La organización del trabajo y el estrés,” 2004.

[6] European Agency for Safety and Health at Work (EU-OSHA), “Calculating the cost of work-related stress and psychosocial risks,” 2014.

[7] “European foundation for the improvement of living and working conditions,” *Fourth European Working conditions Survey*, 2005. [Online]. Available: https://www.eurofound.europa.eu/surveys/european-working-conditions-surveys/fourth-european-working-conditions-survey-2005.

[8] American Psychological Association, “Stress in America,” 2012. [Online]. Available: http://www.apa.org/news/press/releases/stress/index.aspx?tab=5.

[9] IEESA. Instituto de Estudios Educativos y Sindicales de América, “El estrés laboral en los docentes de educación básica: factores desencadenantes y consecuencias.” 2013.

[10] A. Maxhuni, P. Hernandez-Leal, L. E. Sucar, V. Osmani, E. F. Morales, and O. Mayora, “Stress modelling and prediction in presence of scarce data,” *J. Biomed. Inform.*, vol. 63, pp. 344–356, 2016.

[11] D. Carneiro, P. Novais, J. C. Augusto, and N. Payne, “New Methods for Stress Assessment and Monitoring at the Workplace,” *IEEE Trans. Affect. Comput.*, vol. 14, no. 8, pp. 1–1, 2017.

[12] World Health Organization, “Sensibilizando sobre el Estrés Laboral en los Países en Desarrollo.”

[13] R. Blaug, A. Kenyon, and R. Lekhi, “Stress at Work: A report prepared for The Work Foundation’s Principal Partners,” London, 2007.

[14] A. Alberdi, A. Aztiria, and A. Basarab, “Towards an automatic early stress recognition system for office environments based on multimodal measurements: A review,” *J. Biomed. Inform.*, vol. 59, pp. 49–75, 2016.

[15] Secretaría de Salud de México, “Guía sobre el manejo y prevención del estrés laboral,” México, DF., 2010.

[16] K. Glanz and M. Schwartz, “Stress, coping and health behavior,” in *Health behavior and health education: Theory, research, and practice*, 4th ed., San Francisco, California: Jossey-Bass, 2008, pp. 211–236.

[17] C. Maslach, W. B. Schaufeli, and M. P. Leiter, “Job Burnout,” *Annu. Rev. Psychol.*, vol. 52, no. 1, pp. 397–422, Feb. 2001.

[18] M. Sysoev, U. Sedlar, A. Kos, and M. Pogačnik, “Stress-sensors classification and stress-analysis algorithms,” *Stress*, vol. 81, no. 5, pp. 263–271, 2014.

[19] A. Bogomolov, B. Lepri, M. Ferron, F. Pianesi, and A. S. Pentland, “Pervasive Stress Recognition for Sustainable Living,” in *Pervasive Computing and Communication Workshops*, 2014, pp. 345–350.

[20] A. Kołakowska, “Towards detecting programmers’ stress on the basis of keystroke dynamics,” *2016 Fed. Conf. Comput. Sci. Inf. Syst.*, vol. 8, pp. 1621–1626, 2016.

[21] V. Padma, N. Anand, S. M. G. S. Gurukul, S. M. A. S. M. Javid, A. Prasad, and S. Arun, “Health problems and stress in Information Technology and Business Process Outsourcing employees,” *J. Pharm. Bioallied Sci.*, vol. 7, no. 5, p. 9, 2015.

[22] G. Fischer, “Cognitive view of reuse and redesign,” *IEEE Softw.*, vol. 4, no. 4, pp. 60–72, 1987.

[23] I. A. Khan, W.-P. Brinkman, and R. M. Hierons, “Do moods affect programmers’ debug performance?,” *Cogn. Technol. Work*, vol. 13, no. 4, pp. 245–258, Nov. 2011.

[24] L. G. Wallgren and J. J. Hanse, “Job characteristics, motivators and stress among information technology consultants: A structural equation modeling approach,” *Int. J. Ind. Ergon.*, vol. 37, no. 1, pp. 51–59, 2007.

[25] D. Graziotin, X. Wang, and P. Abrahamsson, “Software developers, moods, emotions, and performance,” *IEEE Softw.*, vol. 31, no. 4, pp. 24–27, 2014.

[26] S. C. de B. Sampaio *et al.*, “A review of productivity factors and strategies on software development,” in *2010 Fifth International Conference on Software Engineering Advances*, 2010, pp. 196–204.

[27] T. Dybå, “Improvisation in small software organizations,” *IEEE Software*, vol. 17, no. 5. pp. 82–87, 2000.

[28] S. Maitlis and S. Sonenshein, “Sensemaking in crisis and change: Inspiration and insights from weick (1988),” *J. Manag. Stud.*, vol. 47, no. 3, pp. 551–580, 2010.

[29] M. R. Wrobel, “Emotions in the software development process,” *2013 6th Int. Conf. Hum. Syst. Interact.*, pp. 518–523, 2013.

[30] V. Sreecharan and M. S. Reddy, “A study on individual and interpersonal stress levels among software employees,” *Int. J. Inf. Technol. Comput. Sci. Perspect.*, vol. 2, no. 4, pp. 711–716, 2013.

[31] E. D. Chowdary, K. A. Devi, D. Mounika, S. Venkatramaphanikumar, and K. V. K. Kishore, “Ensemble Classification technique to detect Stress in IT- Professionals,” *2016 Int. Conf. Inven. Comput. Technol. (Icict), Vol 3*, pp. 525–529, 2015.

[32] A. D. Carswell and I. Bojanova, “E-learning for IT professionals: The UMUC experience,” *IT Prof.*, vol. 13, no. 6, pp. 16–21, 2011.

[33] J. Shropshire and C. Kadlec, “I’m leaving the IT field: The impact of stress, job insecurity, and burnout on IT professionals,” *Int. J. Inf.*, vol. 2, no. 1, pp. 6–16, 2012.

[34] A. J. Ko, R. DeLine, and G. Venolia, “Information Needs in Collocated Software Development Teams,” *Icse 2007*, pp. 344–353, 2007.

[35] B. Vasilescu *et al.*, “The Sky Is Not the Limit: Multitasking Across GitHub Projects,” *Proc. 38th Int. Conf. Softw. Eng. - ICSE ’16*, pp. 994–1005, 2016.

[36] R. Wirth and R. Wirth, “CRISP-DM: Towards a standard process model for data mining,” *Proc. FOURTH Int. Conf. Pract. Appl. Knowl. Discov. DATA Min.*, pp. 29--39, 2000.

[37] S. Cohen, “Perceived Stress Scale,” *Psychology*, pp. 1–3, 1994.

[38] R. Karasek, C. Brisson, N. Kawakami, I. Houtman, P. Bongers, and B. Amick, “The Job Content Questionnaire (JCQ): An instrument for internationally comparative assessments of psychosocial job characteristics,” *J. Occup. Health Psychol.*, vol. 3, no. 4, pp. 322–355, 1998.

[39] T. Haratani, “Psychometric evaluation of the NIOSH job stress questionnaire and the job content questionnaire,” *Sangyo Eiseigaku Zasshi*, vol. 40, 1998.

[40] T. Haratani, N. Kawakami, and S. Araki, “Reliability and validity of Chinese version of the NIOSH generic job stress questionnaire,” *Jpn J Ind Heal.*, vol. 35, 1993.

[41] S. Kazronian, S. A. Zakerian, J. N. Saraji, and M. Hosseini, “Reliability and Validity study of the NIOSH Generic Job Stress Questionnaire (GJSQ) among Firefighters in

Tehran city," *Heal. Saf. Work*, vol. 3, no. 3, pp. 25–34, Dec. 2013.

[42] M. Nübling, U. Stößel, H.-M. Hasselhorn, M. Michaelis, and F. Hofmann, "Measuring psychological stress and strain at work - Evaluation of the COPSOQ Questionnaire in Germany.," *Psychosoc. Med.*, vol. 3, p. Doc05, Oct. 2006.

[43] The American Institute of Stress, "Workplace Stress Survey." p. 100.

[44] S. Ackroyd, *Data collection in context*. Longman Group United Kingdom, 1992.

[45] J. Milne, "Questionnaires: Advantages and disadvantages," *Eval. Cookb.*, 1999.

[46] D. H. Hellhammer, S. Wüst, and B. M. Kudielka, "Salivary cortisol as a biomarker in stress research," *Psychoneuroendocrinology*, vol. 34, no. 2, pp. 163–171, Feb. 2009.

[47] E. V. Goldfarb, M. I. Froböse, R. Cools, and E. A. Phelps, "Stress and Cognitive Flexibility: Cortisol Increases Are Associated with Enhanced Updating but Impaired Switching," *J. Cogn. Neurosci.*, vol. 29, no. 1, pp. 14–24, Jan. 2017.

[48] L. Luo, L. Xiao, D. Miao, and X. Luo, "The Relationship between Mental Stress Induced Changes in Cortisol Levels and Vascular Responses Quantified by Waveform Analysis: Investigating Stress-Dependent Indices of Vascular Changes," in *2012 International Conference on Biomedical Engineering and Biotechnology*, 2012, pp. 929–933.

[49] W. Wu, S. Pirbhulal, H. Zhang, and S. C. Mukhopadhyay, "Quantitative Assessment for Self-Tracking of Acute Stress based on Triangulation Principle in a Wearable Sensor System," *IEEE J. Biomed. Heal. Informatics*, pp. 1–1, 2018.

[50] J. A. Healey and R. W. Picard, "Detecting Stress During Real-World Dring Tasks Using Physiological Sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 156–166, 2005.

[51] A. Barreto, J. Zhai, and M. Adjouadi, "Non-intrusive Physiological Monitoring for Automated Stress Detection in Human-Computer Interaction," *Human–Computer Interact.*, pp. 29–38, 2007.

[52] A. O. Akmandor and N. K. Jha, "Keep the Stress Away with SoDA: Stress Detection and Alleviation System," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 3, no. 4, pp. 269–282, Oct. 2017.

[53] S. Yoon, J. K. Sim, and Y. H. Cho, "A Flexible and Wearable Human Stress Monitoring Patch," *Sci. Rep.*, vol. 6, no. August 2015, pp. 1–11, 2016.

[54] N. Sharma and T. Gedeon, "Modeling Stress Recognition in Typical Virtual Environments," in *Proceedings of the ICTs for improving Patients Rehabilitation Research Techniques*, 2013, pp. 17–24.

[55] A. Berina, D. Sejdinovic, L. Gurbeta, and A. Badnjevic, "Classification of Stress Recognition using Artificial Neural Network," in *Mediterranean Conference on Embedded Computing*, 2016, pp. 297–300.

[56] M. Chauhan, S. V Vora, and D. Dabhi, "Effective Stress Detection using Physiological Parameters," in *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017.

[57] M. Salai, I. Vassányi, and I. Kósa, "Stress detection using low cost heart rate sensors," *J. Healthc. Eng.*, vol. 2016, no. i, 2016.

[58] B. Zhang, Y. Morère, L. Sieler, C. Langlet, B. Bolmont, and G. Bourhis, "Reaction time and physiological signals for stress recognition," *Biomed. Signal Process. Control*, vol. 38, pp. 100–107, 2017.

[59] D. Mcduff, S. Gontarek, and R. Picard, "Remote Measurement of Cognitive Stress via Heart Rate Variability," pp. 2957–2960, 2014.

[60] D. J. McDuff, J. Hernandez, S. Gontarek, and R. W. Picard, "COGCAM: Contact-free Measurement of Cognitive Stress During Computer Tasks with a Digital Camera," *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst.*, pp. 4000–4004, 2016.

[61] S. Betti *et al.*, "Evaluation of an integrated system of wearable physiological sensors for stress monitoring in working environments by using biological markers," *IEEE Trans. Biomed. Eng.*, vol. 9294, no. c, 2017.

[62] N. Keshan, P. V. Parimi, and I. Bichindaritz, "Machine learning for stress detection from ECG signals in automobile drivers," *Proc. - 2015 IEEE Int. Conf. Big Data, IEEE Big Data 2015*, pp. 2661–2669, 2015.

[63] R. K. Nawasalkar, "EEG based Stress Recognition System based on Indian Classical Music," 2015.

[64] D. R. Chavan, M. S. Kumbhar, and R. R. Chavan, "The human stress recognition of brain, using music therapy," *2016 Int. Conf. Comput. Power, Energy, Inf. Commun. ICCPEIC 2016*, pp. 200–203, 2016.

[65] X. Hou, Y. Liu, O. Sourina, and W. Mueller-Wittig, "CogniMeter: EEG-based Emotion, Mental Workload and Stress Visual Monitoring," *Proc. - 2015 Int. Conf. Cyberworlds, CW 2015*, pp. 153–160, 2016.

[66] W. L. Lim *et al.*, "EEG-based mental workload and stress monitoring of crew members in maritime virtual simulator," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10830 LNCS, pp. 15–28, 2018.

[67] P. Gaikwad and A. N. Paithane, "Novel Approach for Stress Recognition using EEG Signal by SVM Classifier," in *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, 2017, pp. 967–971.

[68] J. Aigrain, M. Spodenkiewicz, S. Dubuisson, M. Detyniecki, D. Cohen, and M. Chetouani, "Multimodal stress detection from multiple assessments," *IEEE Trans. Affect. Comput.*, vol. 14, no. 8, pp. 1–1, 2016.

[69] H. Yasufuku, T. Terada, and M. Tsukamoto, "A Lifelog System for Detecting Psychological Stress with Glass-equipped Temperature Sensors," *Proc. 7th Augment. Hum. Int. Conf. 2016 - AH '16*, pp. 1–8, 2016.

[70] A. Bogomolov, B. Lepri, F. B. Kessler, F. Pianesi, and A. S. Pentland, "Daily Stress Recognition from Mobile Phone Data , Weather Conditions and Individual Traits," in *ACM international conference on Multimedia*, 2014, pp. 477–486.

[71] E. Vildjiounaite, J. Kallio, V. Kyllönen, M. Nieminen, J. Mäntyjärvi, and G. Gimel'farb, "Unobtrusive stress detection on the basis of smartphone usage data," *Pers. Ubiquitous Comput.*, pp. 1–18, 2018.

[72] R. Ferdous, V. Osmani, and O. Mayora, "Smartphone app usage as a predictor of perceived stress levels at workplace," in *Conference on Pervasive Computing Technologies for Healthcare*, 2015.

[73] E. Garcia-Ceja, V. Osmani, and O. Mayora, "Automatic Stress Detection in Working Environments from Smartphones' Accelerometer Data: A First Step," *Biomed. Heal. Informatics, IEEE J.*, 2015.

[74] M. Ciman and K. Wac, "Individuals' stress assessment using human-smartphone interaction analysis," *IEEE Trans. Affect. Comput.*, vol. 9, no. 1, pp. 1–1, 2016.

[75] M. Ciman, K. Wac, and O. Gaggi, "iSenseStress: Assessing stress through human-smartphone interaction analysis," *Proc. 9th Int. Conf. Pervasive Comput. Technol. Healthc.*, pp. 84–91, 2015.

[76] M. Sysoev, A. Kos, and M. Pogačnik, "Noninvasive stress recognition considering the current activity," *Pers. Ubiquitous Comput.*, vol. 19, no. 7, pp. 1045–1052, 2015.

[77] A. Muaremi, B. Arnrich, and G. Tröster, "Towards Measuring Stress with Smartphones and Wearable Devices During Workday and Sleep," *Bionanoscience*, vol. 3, no. 2, pp. 172–183, May 2013.

[78] H. Sarker *et al.*, "Assessing the availability of users to engage in just-in-time intervention

in the natural environment," *Proc. 2014 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. - UbiComp '14 Adjun.*, pp. 909–920, 2014.

[79] M. Gjoreski, M. Luštrek, M. Gams, and H. Gjoreski, "Monitoring stress with a wrist device using context," *J. Biomed. Inform.*, vol. 73, no. August, pp. 159–170, 2017.

[80] V. Dibia, "FOQUS: A Smartwatch Application for Individuals with ADHD and Mental Health Challenges," *Proc. 18th Int. ACM SIGACCESS Conf. Comput. Access. - ASSETS '16*, pp. 311–312, 2016.

[81] M. Haescher, D. J. C. Matthies, J. Trimpop, and B. Urban, "SeismoTracker: Upgrade any Smart Wearable to enable a Sensing of Heart Rate, Respiration Rate, and Microvibrations," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*, 2016, pp. 2209–2216.

[82] F. Mokhayeri and M.-R. Akbarzadeh-T, "Mental Stress Detection Based on Soft Computing Techniques," in *2011 IEEE International Conference on Bioinformatics and Biomedicine*, 2011, pp. 430–433.

[83] L. A. Torres-Salomao, M. Mahfouf, and E. El-Samahy, "Pupil diameter size marker for incremental mental stress detection," in *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*, 2015, pp. 286–291.

[84] J. Zhai and A. Barreto, "Stress detection in computer users based on digital signal processing of noninvasive physiological variables.," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. 2006, EMBS*, pp. 1355–1358, 2006.

[85] F. Zhang, J. Su, L. Geng, and Z. Xiao, "Driver Fatigue Detection Based on Eye State Recognition," in *2017 International Conference on Machine Vision and Information Technology (CMVIT)*, 2017, pp. 105–110.

[86] K. Imoto, Y. Tsubota, A. Raux, T. Kawahara, and M. Dantsuji, "Modeling and automatic detection of English sentence stress for computer-assisted English prosody learning system," *Acoust. Sci. Technol.*, vol. 24, no. 3, pp. 159–160, 2003.

[87] Min Lai, Yining Chen, Min Chu, Yong Zhao, and Fangyu Hu, "A Hierarchical Approach to Automatic Stress Detection in English Sentences," in *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*, 2006, vol. 1.

[88] M. Ulinskas, M. Wo, M. Woźniak, and R. Damaševičius, "Analysis of Keystroke Dynamics for Fatigue Recognition," *Comput. Sci. Its Appl. – ICCSA 2017. ICCSA 2017. Lect. Notes Comput. Sci.*, vol. 10408, pp. 235–247, 2017.

[89] S. Koldijk, M. A. Neerincx, and W. Kraaij, "Detecting work stress in offices by combining unobtrusive sensors," *IEEE Trans. Affect. Comput.*, vol. 3045, no. c, pp. 1–1, 2016.

[90] J. Andrén and P. Funk, "A Case-Based Approach Using Behavioural Biometrics to Determine a User's Stress Level.," *ICCBR Work.*, vol. 5, no. October, 2005.

[91] A. Brouwer, "Assessing stress at the workplace: an explorative study on measuring emotion using unobtrusive sensor techniques," *TCP Scr.*, 2017.

[92] D. Carneiro, A. Pimenta, J. Neves, and P. Novais, "A multi-modal architecture for non-intrusive analysis of performance in the workplace," *Neurocomputing*, vol. 231, pp. 41–46, 2017.

[93] D. Carneiro, A. Pimenta, S. Gonçalves, J. Neves, and P. Novais, "Monitoring and improving performance in human-computer interaction," *Concurr. Comput. Pract. Exp.*, vol. 28, no. 4, pp. 1291–1309, Mar. 2016.

[94] H. Lu *et al.*, "StressSense: Detecting Stress in Unconstrained Acoustic Environments using Smartphones," in *UbiComp '12 Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012.

[95] KDnuggets, "CRISP-DM, still the top methodology for analytics, data mining, or data

science projects," 2014. [Online]. Available: https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html.

[96] J. M. Koolhaas *et al.*, "Stress revisited: A critical evaluation of the stress concept," *Neurosci. Biobehav. Rev.*, vol. 35, no. 5, pp. 1291–1301, 2011.

[97] H. Selye, *The stress of life*. New York, USA: McGraw-Hill, 1956.

[98] R. S. Lazarus and S. Folkman, "Stress, appraisal and coping." Springer, 1984.

[99] B. S. McEwen, "The neurobiology of stress: from serendipity to clinical relevance," *Brain Res.*, vol. 886, no. 1–2, pp. 172–189, Dec. 2000.

[100] T. W. Colligan and E. M. Higgins, "Workplace Stress," *J. Workplace Behav. Health*, vol. 21, no. 2, pp. 89–97, Jul. 2006.

[101] J. Bakker, M. Pechenizkiy, and N. Sidorova, "What's Your Current Stress Level? Detection of Stress Patterns from GSR Sensor Data," *2011 IEEE 11th Int. Conf. Data Min. Work.*, no. 1, pp. 573–580, Dec. 2011.

[102] N. Sharma, A. Dhall, T. Gedeon, and R. Goecke, "Thermal spatio-temporal data for stress recognition," *Eurasip J. Image Video Process.*, vol. 2014, no. 1, pp. 1–12, 2014.

[103] B. Mishra, S. Mehta, N. D. Sinha, S. K. Shukla, N. Ahmed, and A. Kawatra, "Evaluation of work place stress in health university workers: a study from rural India.," *Indian J. Community Med.*, vol. 36, no. 1, pp. 39–44, Jan. 2011.

[104] EU-OSHA — European Agency for Safety and Health at Work, "OSH in figures: stress at work — facts and figures," Luxembourg, 2009.

[105] A. Broughton, "Work-related stress." Dublin, 2010.

[106] B. H. W. Eijckelhof *et al.*, "Office workers' computer use patterns are associated with workplace stressors," *Appl. Ergon.*, vol. 45, no. 6, pp. 1660–1667, Nov. 2014.

[107] J. Wijsman, B. Grundlehner, H. Liu, J. Penders, and H. Hermens, "Wearable physiological sensors reflect mental stress state in office-like situations," in *Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, 2013, no. iii, pp. 600–605.

[108] J. Han, M. Kamber, and J. Pei, *Data Mining. Concepts and techniques*, Third. Morgan Kaufmann, 2012.

[109] Z. He, *Data Mining for Bioinformatics Applications*. Elsevier Ltd, 2015.

[110] K. Kalegele, K. Sasai, H. Takahashi, G. Kitagata, and T. Kinoshita, "Four Decades of Data Mining in Network and Systems Management," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2700–2716, 2015.

[111] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM J. Res. Dev.*, vol. 3, no. 3, pp. 210–229, Jul. 1959.

[112] M. Kang and N. J. Jameson, "Machine Learning: Fundamentals," in *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things*, M. G. Pecht and M. Kang, Eds. John Wiley & Sons, 2019, pp. 85–109.

[113] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explor.*, vol. 11, no. 1, 2009.

[114] K. Plarre *et al.*, "Continuous Inference of Psychological Stress from Sensory Measurements Collected in the Natural Environment," in *Information Processing in Sensor Networks (IPSN)*, 2011, pp. 97–108.

[115] A. Raij *et al.*, "mStress: Supporting Continuous Collection of Objective and Subjective Measures of Psychosocial Stress on Mobile Devices," Technical report No. CS-10-004, 2010.

[116] J. Hernandez, P. Paredes, A. Roseway, and M. Czerwinski, "Under Pressure: Sensing Stress of Computer Users," in *ACM SIGCHI Conference on Human Factors in*

Computing Systems, 2014, pp. 51–60.

[117] A. Sano and R. W. Picard, "Stress Recognition Using Wearable Sensors and Mobile Phones," in *Humaine Association Conference on Affective Computing and Intelligent Interaction*, 2013, pp. 671–676.

[118] J. Hernandez, R. R. Morris, and R. W. Picard, "Call Center Stress Recognition with Person-Specific Models," in *Affective Computing and Intelligent Interaction*, 2011, vol. 6974, pp. 125–134.

[119] D. Majoe, P. Bonhof, T. Kaegi-trachsel, J. Gutknecht, and L. Widmer, "Stress and Sleep Quality Estimation from a Smart Wearable Sensor," in *Pervasive Computing and Applications (ICPCA)*, 2010, pp. 14–19.

[120] J. Hernandez, X. Benavides, P. Maes, D. McDuff, J. Amores, and R. W. Picard, "AutoEmotive: Bringing Empathy to the Driving Experience to Manage Stress," in *Designing interactive systems*, 2014, pp. 53–56.

[121] A. Pimenta, D. Carneiro, P. Novais, and J. Neves, "Monitoring mental fatigue through the analysis of keyboard and mouse interaction patterns," *Lect. Notes Comput. Sci.*, vol. 8073, pp. 222–231, 2013.

[122] A. Pimenta, D. Carneiro, P. Novais, and J. Neves, "Analysis of Human Performance as a Measure of Mental Fatigue," *Hybrid Artif. Intell. Syst. HAIS 2014. Lect. Notes Comput. Sci.*, vol. 8480, pp. 389–401, 2014.

[123] A. Pimenta, D. Carneiro, J. Neves, and P. Novais, "A neural network to classify fatigue from human-computer interaction," *Neurocomputing*, vol. 172, pp. 413–426, 2015.

[124] D. Carneiro *et al.*, "Multimodal behavioral analysis for non-invasive stress detection," *Expert Syst. Appl.*, vol. 39, no. 18, pp. 13376–13389, 2012.

[125] M. El-Abed, M. Dafer, and R. El Khayat, "RHU Keystroke: A mobile-based benchmark for keystroke dynamics systems," in *2014 International Carnahan Conference on Security Technology (ICCST)*, 2014, pp. 1–4.

[126] I. A. Khan, W. P. Brinkman, and R. Hierons, "Towards estimating computer users' mood from interaction behaviour with keyboard and mouse," *Front. Comput. Sci.*, vol. 7, no. 6, pp. 943–954, 2013.

[127] L. Vizer, "Different strokes for different folks: individual stress response as manifested in typed text," in *Extended Abstracts on Human Factors in Computing Systems*, 2013, pp. 2773–2778.

[128] W. Liao, W. Zhang, Z. Zhu, and Q. Ji, "A Real-Time Human Stress Monitoring System Using Dynamic Bayesian Network," *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Work.*, vol. 3, pp. 70–70, 2005.

[129] M. X. Huang, J. Li, G. Ngai, and H. V. Leong, "StressClick: Sensing Stress from Gaze-Click Patterns," *Proc. 2016 ACM Multimed. Conf. - MM '16*, pp. 1395–1404, 2016.

[130] I. Karunaratne, A. S. Atukorale, and H. Perera, "The Relationship Between Psychological Distress and Human Computer Interaction Parameters: Linear or Non-linear?," *Lect. Notes Electr. Eng.*, vol. 312, pp. 471–478, 2015.

[131] I. Karunaratne, A. S. Atukorale, and H. Perera, "Surveillance of Human-Computer Interactions: A Way Forward to Detection of Users' Psychological Distress," in *Humanities, Science and Engineering (CHUSER)*, 2011, pp. 491–496.

[132] A. Maxhuni, P. Hernandez-leal, E. Morales, L. Enrique, V. Osmani, and O. Mayora, "Using Intermediate Models and Knowledge Learning to Improve Stress Prediction," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2017, vol. 179, pp. 1–12.

[133] R. Ferdous, V. Osmani, J. Beltran, and O. Mayora, "Investigating correlation between verbal interactions and perceived stress," in *Engineering in Medicine and Biology*

Society (EMBS), 2015.

[134] P. Hernandez-Leal, A. Maxhuni, L. E. Sucar, V. Osmani, E. F. Morales, and O. Mayora, “Stress Modelling Using Transfer Learning in Presence of Scarce Data,” *Ambient Intell. Heal.*, pp. 1–12, 2015.

[135] M. Gjoreski, V. Janko, H. Gjoreski, B. Cvetkovć, M. Luštrek, and M. Gams, “Activity and stress monitoring using smartphone and wrist device,” in *7th Jožef Stefan International Postgraduate School Students’ Conference*, 2016, pp. 154–164.

[136] N. Condori-Fernandez, F. Suni Lopez, and A. Catala Bolos, “Towards Real-time Automatic Stress Detection for Office Workplaces,” in *Conference on Information Management and Big Data*, 2018, no. September.

[137] B. Egilmez, E. Poyraz, W. Zhou, G. Memik, P. Dinda, and N. Alshurafa, “UStress: Understanding college student subjective stress using wrist-based passive sensing,” in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017*, 2017, pp. 673–678.

[138] L. Han, Q. Zhang, X. Chen, Q. Zhan, T. Yang, and Z. Zhao, “Detecting work-related stress with a wearable device,” *Comput. Ind.*, vol. 90, pp. 42–49, 2017.

[139] B. Cvetkovic *et al.*, “Management of Physical, Mental and Environmental Stress at the Workplace,” in *2017 International Conference on Intelligent Environments (IE)*, 2017, pp. 76–83.

[140] M. Gjoreski and H. Gjoreski, “Continuous Stress Detection Using a Wrist Device – In Laboratory and Real Life,” in *UbiComp*, 2016, pp. 1185–1193.

[141] B. Cvetković *et al.*, “Real-time Physical Activity and Mental Stress Management with a Wristband and a Smartphone,” *Proc. 2017 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. Proc. 2017 ACM Int. Symp. Wearable Comput.*, pp. 225–228, 2017.

[142] G. Giannakakis *et al.*, “Stress and anxiety detection using facial cues from videos,” *Biomed. Signal Process. Control*, vol. 31, pp. 89–101, 2017.

[143] Y. Cho, N. Bianchi-Berthouze, and S. J. Julier, “DeepBreath: Deep Learning of Breathing Patterns for Automatic Stress Recognition using Low-Cost Thermal Imaging in Unconstrained Settings,” pp. 456–463, 2017.

[144] Y. Cho, “Automated mental stress recognition through mobile thermal imaging,” *2017 Seventh Int. Conf. Affect. Comput. Intell. Interact.*, pp. 596–600, 2017.

[145] I. Lefter, G. J. Burghouts, and L. J. M. Rothkrantz, “Recognizing Stress Using Semantics and Modulation of Speech and Gestures,” *IEEE Trans. Affect. Comput.*, vol. 7, no. 2, pp. 162–175, 2016.

[146] S. K. Thompson, *Sampling*, Third. Wiley, 2012.

[147] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[148] I. H. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Burlington: Morgan Kaufmann, 2011.

[149] IBM Corp., “IBM SPSS Statistics for Windows.” IBM Corp., Armonk, NY, 2017.

[150] R. C. Team, “R: A language and environment for statistical computing. R Foundation for Statistical Computing.” Vienna, Austria, 2013.

[151] J. Demsar *et al.*, “Orange: Data Mining Toolbox in Python,” *J. Mach. Learn. Res.*, vol. 14, p. 2349−2353, 2013.

[152] E. Szmidt and J. Kacprzyk, “The Spearman rank correlation coefficient between intuitionistic fuzzy sets,” *2010 5th IEEE Int. Conf. Intell. Syst.*, pp. 276–280, 2010.

[153] G. H. Dunteman, *Principal Components Analysis*. SAGE Publications, 1989.

[154] M. A. Hall, “Correlation-based Feature Selection for Machine Learning,” The University of Waikato, 1999.

[155] A. Sharma and S. Dey, "Performance Investigation of Feature Selection Methods and Sentiment Lexicons for Sentiment Analysis," *Spec. Issue Int. J. Comput. Appl.*, pp. 15–20, 2012.

[156] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, Dec. 1997.

[157] O. Loyola, M. A. Medina, and M. García, "Inducing Decision Trees based on a Cluster Quality Index," *IEEE Lat. Am. Trans.*, vol. 13, no. 4, pp. 1141–1147, 2015.

[158] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," *UAI'95 Proc. Elev. Conf. Uncertain. Artif. Intell.*, pp. 338–345, Aug. 1995.

[159] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[160] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. 2006.

[161] R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[162] J. R. Quinlan, "Induction of Decision Trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[163] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth International Group, 1984.

[164] Zhi-Song Pan, Song-Can Chen, Gen-Bao Hu, and Dao-Qiang Zhang, "Hybrid neural network and C4.5 for misuse detection," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, 2003, no. 4, pp. 2463–2467.

[165] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining," in *KDD*, 1998.

[166] P. Clark and T. Niblett, "The CN2 Induction algorithm," *Mach. Learn.*, vol. 3, no. 4, pp. 261–283, 1989.

[167] W. Cohen and Y. Singer, "Context-sensitive learning methods for text categorization," *ACM Trans. Inf. Syst.*, vol. 17, no. 2, pp. 141–173, 1999.

[168] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.

[169] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. MIT Press, 1999, pp. 185–208.

[170] I. Dagan, Y. Karov, and D. Roth, "Mistake-driven Learning in Text Categorization," in *Proceedings of EMNLP*, 1997.

[171] N. Littlestone, "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Mach. Learn.*, vol. 2, pp. 285–318, 1988.

[172] H. T. Ng, W. Goh, and K. Low, "Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization," in *ACM SIGIR Conference*, 1997.

[173] H. Schutze, D. Hull, and J. Pedersen, "A Comparison of Classifiers and Document Representations for the Routing Problem," in *ACM SIGIR Conference*, 1995.

[174] E. Wiener, J. O. Pedersen, and A. S. Weigend, "A neural network approach to topic spotting," in *SDAIR*, 1995, pp. 317–332.

[175] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[176] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," *Thirteen. Int. Conf. Mach. Learn.*, pp. 148–156, 1996.

[177] X. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, Dec. 2008.

[178] X. Wu and V. Kumar, *Top ten algorithms in data mining*, vol. 14, no. 1. Florida: Chapman & Hall/CRC, 2009.

[179] Q. YANG and X. WU, “10 Challenging Problems in Data Mining Research,” *Int. J. Inf. Technol. Decis. Mak.*, vol. 05, no. 04, pp. 597–604, 2006.

[180] R. C. Holte, “Very Simple Classification Rules Perform Well on Most Commonly Used Datasets,” *Mach. Learn.*, vol. 11, no. 1, pp. 63–90, Apr. 1993.

[181] P. C. Team, “Python: A dynamic, open source programming language. Python Software Foundation.” 2015.

[182] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, “Julia: A Fast Dynamic Language for Technical Computing,” Sep. 2012.

[183] K. Arnold, J. Gosling, and D. Holmes, *The Java programming language*. Addison-Wesley, 2000.

[184] M. Hofmann and R. Klinkenberg, *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC, 2013.

[185] M. R. Berthold *et al.*, “KNIME: The Konstanz Information Miner,” in *Studies in Classification, Data Analysis, and Knowledge Organization*, Springer, Berlin, Heidelberg, 2008, pp. 319–326.

[186] M. Camaños, W. Sánchez, A. Martínez, and M. Mejía-Lavalle, “LaborCheck: Sistema de Monitoreo Automático de Variables Usuario-Computadora y de Interacción Social,” in *Mexican International Conference on Computer Science - Taller de Computación Clínica e Informática Médica*, 2016.

[187] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.