NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENT OF ELECTRICAL AND COMPUTER ENGINEERING

João Luis de Andrade Vaz

BSc in Electrical and Computer Engineering

LEVERAGING WEARABLE DEVICE DATA FOR CIRCADIAN RHYTHM ANALYSIS IN ONCOLOGY PATIENT FOLLOW-UP CARE

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon

September, 2023

Examination Committee:

Chair: Fernando José Vieira do Coito,
Associate Professor, NOVA University Lisbon

Adviser: João Paulo Pimentão,
Assistant Professor, NOVA University Lisbon

Arguer: José Barata Oliveira,
Full Professor, NOVA University Lisbon

Adviser: João Paulo Pimentão
Assistant Professor, NOVA University Lisbon

Co-advisers: Pedro Alexandre Sousa
Associate Professor, NOVA University Lisbon

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
September, 2023

DEPARTAMENT OF
ELETRICAL AND COMPUTER ENGINEERING

LEVERAGING WEARABLE DEVICE DATA FOR CIRCADIAN RHYTHM ANALYSIS IN ONCOLOGY PATIENT FOLLOW-UP CARE

JOÃO LUIS DE ANDRADE VAZ

BSc in Electrical and Computer Engineering

Leveraging Wearable Device Data for Circadian Rhythm Analysis in Oncology Patient Follow-up Care

To my late Grandma.

ACKNOWLEDGMENTS

I would like to start by giving my sincere gratitude to my advisers,
Pedro Sousa and João Paulo Pimentão, for accepting me in this project that has brought a lot of technical proficiency and personal gratification to me. The fight against cancer really is an area that speaks to me and my motivation. Thank you also for the continuous guidance that both of you provided throughout the entire process of my dissertation. I truly appreciate your patience with me and enthusiasm with the project.
I also want to give my thanks to HOLOS, a place I´ve been to almost everyday in the past year, and where I was so well received and mentored.
To my family and friends, who are such an important part of my life; no words can make justice of the weight of your support.

"Be ashamed to die until you have won some victory for humanity"
- Horace Mann

ABSTRACT

The circadian rhythm is a useful set of parameters to evaluate the physiological and behavioral patterns of patients. As patients that overcame oncology treatment suffer the lasting marks of deteriorated quality of life and health, the act of measuring the circadian profile of these patients can be useful for their follow-up care process. With the use of an ever-growing technology such as wearable sensor devices, the process of collecting circadian information about a patient becomes feasible.

This dissertation focuses on the research and development of features and tools for the CLARIFY (Cancer Long Survivors Artificial Intelligence Follow-up) web application. CLARIFY is an EU funded project that aims to ease the follow-up process of patients past their oncology treatments. This dissertation focuses on processes of dynamically integrating the data collected from wearable devices (provided by an IT team named Kronohealth) into the necessary data repositories of the app. Backend and user interface developments are also implemented to provide a tool for clinicians and medical professionals to analyze each patient's circadian variables with those of a control population. The information gained from a concrete and descriptive circadian profile of a patient is used by clinicians to help with their decision making regarding suggestions/interventions/medications to prescribe.

A study was also made using artificial intelligence techniques to evaluate the viability of producing a model that could predict which medical intervention a patient should be prescribed using a set of the patient's general and oncology data, as well as a set of their circadian rhythm data.

This dissertation was done in coordination with the Hospital Universitario Puerta de Hierro-Majadahond in Madrid, Kronohealth and Holos.

Key Words: Circadian Rhythm, Wearable Devices, Post Oncological Treatment Follow-up, ETL, Machine Learning, User Interface.

RESUMO

O ritmo circadiano é um conjunto útil de parâmetros para avaliar os padrões fisiológicos e comportamentais dos pacientes. Como os pacientes que superam o tratamento oncológico sofrem as marcas duradouras da deterioração da qualidade de vida e saúde, o ato de medir o perfil circadiano desses pacientes pode ser útil para o seu processo de acompanhamento. Com o uso de uma tecnologia cada vez maior, como dispositivos sensores vestíveis, o processo de recolha de informações circadianas sobre um paciente torna-se viável.

Esta dissertação centra-se na investigação e desenvolvimento de funcionalidades e ferramentas para a aplicação web CLARIFY (Cancer Long Survivors Artificial Intelligence Follow-up). CLARIFY é um projeto financiado pela UE que visa facilitar o processo de acompanhamento de pacientes após tratamentos oncológicos. Esta dissertação foca-se nos processos de integração dinâmica dos dados recolhidos de dispositivos vestíveis (fornecidos pela Kronohealth) nos repositórios de dados necessários da aplicação. Além disso, desenvolvimentos de *backend* e interface de utilizador também são implementados para fornecer uma ferramenta para médicos e profissionais de saúde analisarem as variáveis circadianas de cada paciente comparando com uma população de controlo. A informação obtida a partir de um perfil circadiano concreto e descritivo de um paciente é utilizada por médicos para ajudar na tomada de decisão relativamente a sugestões/intervenções/medicamentos a prescrever.

Também foi feito um estudo utilizando técnicas de inteligência artificial para avaliar a viabilidade de um modelo que pudesse prever que intervenção médica seria prescrita a um paciente usando um conjunto de dados gerais e oncológicos do paciente, bem como um conjunto de dados de seu ritmo circadiano.

Esta dissertação foi realizada em coordenação com o Hospital Universitário Puerta de Hierro-Majadahond de Madrid, Kronohealth e Holos.

Palavras chave: Ritmo Circadiano, Dispositivos Vestíveis, Acompanhamento Pós Tratamento Oncológico, ETL, Machine Learning, Interface de Usuário.

Contents

List of Figures

List of Tables

GLOSSARY

ETL	All of the necessary processes to Extract data from a source, Transform it into more usable data and Load it into a storage component such a Database.
JavaScript	Popular programming language, commonly used for web development.
JS Framework	Modern approaches to user interface development usually use a pre-written collection of code and libraries that provides a structured foundation for developing web applications.
Circadian Rhythm	Set of physiological and behavioral processes in living organisms that contribute to wellbeing and health.
Medical Wearable Devices	Portable and lightweight electronic devices designed to be worn by patients for monitoring and managing various health-related variables.
Back-end	Server-side components of an application. Usually responsible for processing data and interacting with databases.
Front-end	Client-side refers to the user interface and the components of an application or system that users interact with directly
Cloud	Refers to services of remote servers that are hosted on the internet and used to store, manage, and process data, rather than using a local server or a personal computer. Very common nowadays in the IT development space.

Post Oncology Follow-up	Medical care and monitoring that patients receive after they have completed their primary treatment for cancer.
Machine Learning	Algorithms based on statistical mathematics that allow computers to learn from datasets and perform specific tests. Branch of Artificial Intelligence.
Neural Networks	Set of Artificial Intelligence techniques that aim to mimic the workings of the human brain, and be able to perform elaborate tasks.
Classification Problems	Common use case for artificial intelligence techniques where the objective is to categorize or classify data points into predefined categories or classes.

Acronyms

CLARIFY	Cancer Long Survivor Artificial Intelligence Follow-up
ETL	Extract Transform Load
QoL	Quality of Life
EHR	Electronic Health Records
PRO	Patient-Reported Outcome
HIE	Health Information Exchange
SQL	Structured Query Language
IoT	Internet of Things
MIoT	Medical Internet of Things
GCP	Google Cloud Platform
AWS	Amazon Web Services
SVM	Support Vector Machine
MRI	Magnetic Resonance Imaging
DTR	Dynamic Treatment Regimes
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
FaaS	Function as a Service
BaaS	Backend as a Service
HTML	HyperText Markup Language

HTTP	HyperText Transfer Protocol
JS	JavaScript
UI	User Interface
FDA	Food and Drug Administration
FTP	File Transfer Protocol
SMTP	Simple Mail Transfer Protocol
POP3	Post Office Protocol
SFTP	SSH File Transfer Protocol
FTPS	File Transfer Protocol Secure
TLS	Transport Layer Security
SSL	Secure Sockets Layer
GCF	Google Cloud Functions
AWS	Amazon Web Services
API	Application Programming Interface
ANN	Artificial Neural Network
MLP	Multilayer Perceptron
RBF	Radial Basis Function
RNN	Recurrent Neural Networks
MNN	Modular Neural Networks
CNN	Convolutional Neural Networks
FCN	Fully Convolutional Networks
GAE	Google App Engine
TAP	Thermometry, Actimetry and Body Position
GCE	Google Compute Engine
AI	Artificial Intelligence

TRANSLATIONS

Tiempomov Time of Movement Activity (Spanish)

Intensidad Intensity of Movement (Spanish)

SYMBOLS

σ	Sigmoid activation function.

1

INTRODUCTION

1.1 Motivation

Cancer is one of the leading causes of death worldwide, and according to the World Health Organization, lung cancer is the variant that accounts for the most deaths [1], while sharing the top two most common causes with breast cancer [40]. Despite advances in prevention, diagnosis, and treatment the survival rates for lung cancer are still relatively low with over half the patients diagnosed not surviving more than one year [1]. For breast and lymphoma cancers, the survival rates depend greatly on the stage of cancer and type of lymphoma cancer, but in general 5-year survival rates seem to be more generous than in their lung counterparts.

Obviously, there is still a pressing need to develop new ways and strategies of battling this disease in order to improve survival rates and indeed oncology has been one of the areas with the most amount of research and scientific literature published. According to the Clinicaltrials(dot)gov[1], oncology is the area of medicine with the biggest volume of clinical trials [41] [65]. This field has experienced innovations at a remarkable pace in the last decade. From new treatments such as immunotherapy and modern non-invasive diagnostic techniques like liquid biopsy to the use of artificial intelligence [37], it seems many areas come together in the fight against cancer. This combined effort has shown to have increased the survival rates in recent years when compared to the 1990s. However, the follow up models for the surviving patients have not evolved in parallel and are inadequate to meet the needs that these patients report. This is a critical issue given that studies show that cancer survivors show symptoms of fatigue, anxiety, and depression, as well as functional limitations and decreased quality of life in general [3]. Moreover, these survivors are at high risk for developing new primary malignancies, second primary malignancies, and other chronic conditions such as cardiovascular and pulmonary diseases [4]. This highlights the need for adequate follow-up care to address the unmet needs of these patients.

[1] https://clinicaltrials.gov/

1.2 Proposed Work and Objectives

CLARIFY [1] or Cancer Long Survivors Artificial Intelligence Follow up is a EU-Funded project and a web application developed by HOLOS[2] in partnership with the oncology department of the Hospital Universitario Puerta de Hierro-Majadahonda that serves medicine professionals with their cancer patients. An objective of this project is to identify risk factors for deterioration of quality of life in a patient after oncological treatment and mitigate their symptoms based on their circadian profile.

The CLARIFY app has multiple purposes, such as individual and population analysis, predictive models and many others. Currently supports lung, breast, and lymphoma cancers.

Some CLARIFY patients undergo a week with a wearable device on their wrist, to track their circadian status in hopes that it reveals information about their lifestyle. The device measures variables such as daylight exposure, physical activity levels, sleeping patterns, among others. This circadian description of a patient is then compared against a cancer free population of 10 000 people. The main premise is that this concrete data can help their oncology doctor to recommend interventions or changes to improve the patient's quality of life. These wearable devices are provided by Kronohealth[3], a technology based company aimed at healthcare and especially the circadian system. The growing popularity of fields such as MIoT (Medical Internet of Things) and the increased interest in data driven solutions in medicine, have made these types of devices very appealing for the remote monitoring of patients.

CLARIFY's data repository integrates data from comprehensive Electronic Health Records, or EHR, genomic data, data from open sources such as PubMed and DrugBank and data from wearable devices.

The research and implementations in this work are fully aimed to develop and improve the CLARIFY web application. Some of the main functionalities that need to be implemented are:

- Automatic import and load of Kronohealth data to the data repository (ETL)
- Development and setting of the required cloud computing resources
- Development of the user interface components (comparison graphs and interventions)
- Machine learning algorithms to suggest areas of interventions

To meet the needs and understand the ways in which clinicians consult and use the application, meetings with Dra Maria Torrente from the Hospital Universitario Puerta de

[2] www.holos.pt
[3] www.kronohealth.com

Hierro-Majadahonda and Professor Manuel Campos, the CTO of Kronohealth were done frequently, so that feedback on potential updates and new features can be considered.

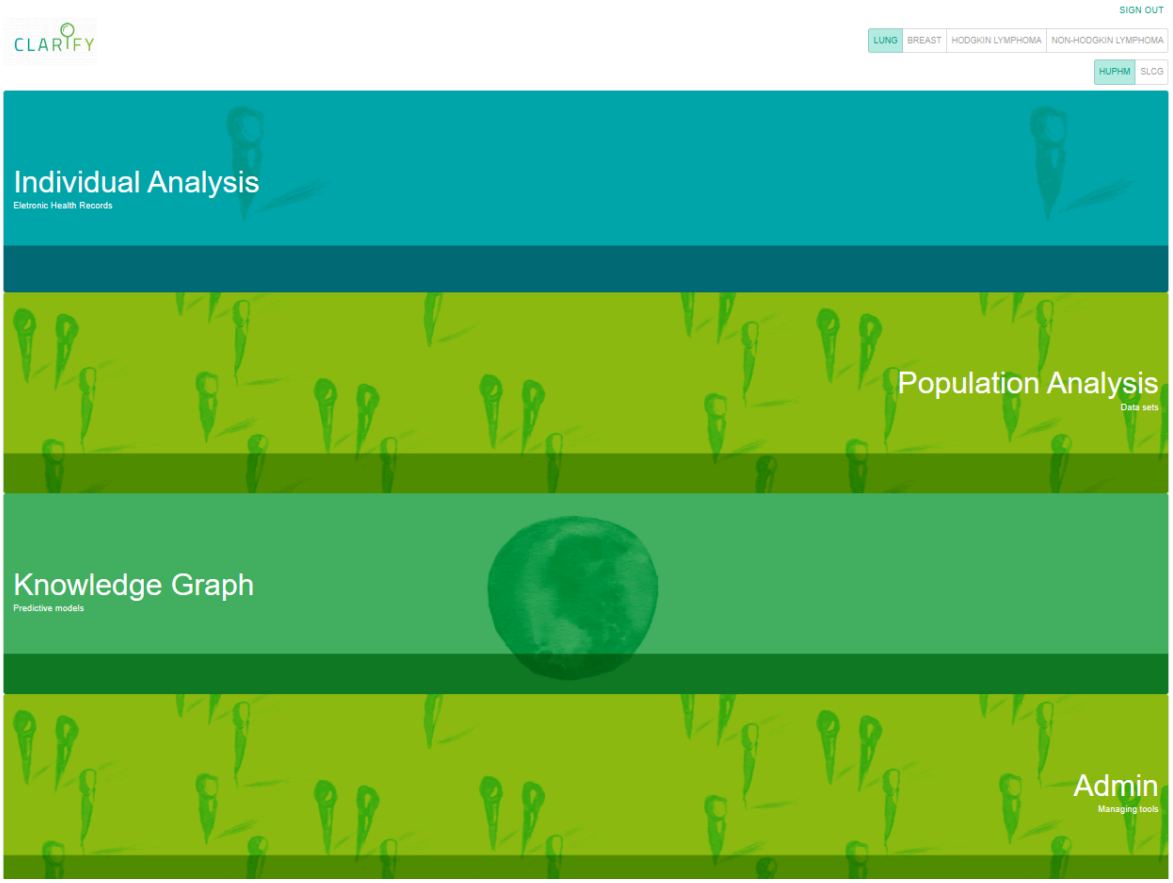

Figure 1 - User Interface of CLARIFY, initial Dashboard

1.3 Approach and Document Organization

The remaining sections of this document are structured in the following way: section 2, or state of the art, describes all of the research done in the topics deemed relevant to this work. First there is an emphasis on papers and projects that show different ways that data science is used across different areas of healthcare and medicine. A deeper analysis is done particularly on the circadian rhythm and the significance of chronobiology and how it is related to a patient's quality of life and potential symptoms. Thereafter, a few examples of other uses for medical wearable devices are explored in hopes that relevant similarities to our project could be worth noting. Some ethical considerations about the privacy and use of data in this area are also briefly mentioned.

Right after that, still in the same section, comes the technical research part. With the ultimate objective of implementing the necessary features to the web application, a more indepth and thorough understanding of the underlying technologies and concepts was necessary. Research was done in the fields of data transfer protocols, ETL, user interface development and JS frameworks, serverless or cloud computing and machine learning. All of these fields had some relevance in the project implementations. Examples were drawn from the existing body of literature where these components have been successfully studied or deployed to understand some of the important considerations and best practices of each field.

The section 3 presents the choices and processes involved in the necessary developments done in the CLARIFY web application. Topics such as the architecture chosen to transfer and import data, the cloud services and user interface development and the machine learning algorithms done using patients' data are discussed in depth.

Finally, section 4 shows the results of the implementations, as well as a conclusion on how important the circadian variables obtained through wearable devices are, to determine patient interventions and recommended changes.

2

STATE OF THE ART

2.1 Data in Healthcare and Medicine

Data plays a crucial role in healthcare, particularly in the area of cancer research and treatment. A very large amount of information is generated by electronic health records, genomic data, imaging technologies, wearable devices and others. This data has the potential to enhance our understanding of a disease and improve patient outcomes and quality of life. Big data and artificial intelligence techniques have been applied to predict cancer outcomes [63] and treatment responses.

A 2019 paper [42] discusses the ways that big data is used in the field of healthcare. The paper points out the multiple views on this topic, ranging from optimists, who tend to emphasize the potential benefits and advantages of using big data, but also pessimists, who have raised concerns and highlighted potential negative aspects, such as privacy issues or data security challenges. Overall, the authors conclude that while big data technologies are very promising, certain parameters must be met to maximize its value. The authors elaborate and provide guiding principles for using it effectively to generate meaningful insights and evidence for healthcare decision-making. These guidelines are based upon the Bradford Hill criteria [64], which is a set of criteria in the field of epidemiology to determine whether or not causation can be concluded between a presumed cause and an observed effect. The guiding principles proposed by the authors are:

- Generate Evidence by Scientific Methods: When working with big data, researchers must follow rigorous scientific methods to make sure their results are reliable and accurate.

- Assure Validity and Significance: It's crucial to ensure that the data analysis produces valid and statistically significant results.
- Use Multi-Step Analysis: Big data analysis often involves complex processes that usually encompass data cleaning, preprocessing, modeling, and validation to extract meaningful insights.
- Consider Sensitivity and Specificity: In medical contexts, these parameters are used to describe the accuracy of a diagnostic test that reports whether a condition is present or absent.

In order to obtain relevant information from big data or data in general, techniques such as population analysis based on statistical methods or artificial intelligence algorithms are commonly done.

Studies have shown that machine learning algorithms with deep learning approaches can be effectively used to predict the survival of patients with cancer based on their clinical data with very high accuracy and could be implemented in clinical decision-making. [5] A 2019 study from the University of Malaya Medical Centre, used 4900 patient records to predict breast cancer survival. The researchers used various machine learning methods such as random forest, decision trees and support vector machines on the tested samples and managed to achieve very decent results, however it was the multilayer perceptron, which is a neural network algorithm, that got the best results by a considerable margin, being able to predict with an accuracy of 88% compared to only 83% for the best non deep learning method. The authors also mentioned that data transformations and parameter configurations impact severely the outcome of these algorithms and that proper parameterization is necessary.

Another 2019 study [43] showed how neural networks and other algorithms can be optimized when used in the context of medical healthcare. The case study chosen was prediction of apnea in neonatal children. A big emphasis was given to the optimization of a multilayer perceptron and the authors propose a novel framework for tuning of this algorithm. Succinctly the framework should follow the steps:

- Select the best performing activation function.
- Select the best performing gradient descent algorithm given the previous activation function selected.
- Select the best performing depth of network given the previous optimal activation function and gradient descent algorithm.
- Select the best performing regularization algorithm given the previous optimal activation function, gradient descent algorithm and depth of network.

- Select the best performing updater algorithm maintaining all other previously chosen parameters.

The approach revolves around maintaining the hyperparameters that were identified as optimal in the previous step, and in the case of the multilayer perceptron (MLP), this is the order that the paper argues for. The researchers then compare an MLP tuned with their novel approach with other neural networks and machine learning methods. The data used was notoriously complex and noisy as this represents realistic applications of healthcare data. They conclude that indeed a MLP can be significantly optimized and improved for classification and prediction purposes and it can be as accurate as deep learning approaches such as deep belief networks and deep auto-encoders and a lot more accurate than the more traditional and simpler machine learning algorithms such as decision trees, support vector machines, K Nearest Neighbor, etc.

While the positive and promising uses of data in healthcare keeps on getting more undisputed, there are also some concerns surrounding the ethics of those practices. The most recurring concerns are centered around the privacy and the security of the data. While the privacy concerns are usually focused around data ownership and anonymity, breaches in healthcare data can have severe consequences such as identity theft and insurance fraud. A 2007 article [44] claims that while most research endeavors that rely on sensitive healthcare data undergo independent ethical review, some audit activities that systematically collect patient data are exempt from review. In order to address ethical data use, some guidelines, regulations and laws have been created such as the Health Insurance Portability and Accountability Act of 1996 (HIPAA)[45] in the United States, and the General Data Protection Regulation (GDPR) in Europe [46]. These regulations make sure healthcare data usage is consensual and private meaning the patients have the right to know how their data is being used and have the ability to give or withdraw their consent. Organizations or researchers that have access to data are required to act with full transparency and to follow data minimization principles which means to only extract and use the necessary information regarding their concrete use case. In most cases, failure to comply with the regulations can result in penalties and fines, and entities that fail to protect patient data or violate privacy rights will likely face legal consequences.

As mentioned briefly in the introduction, medical data is usually very complex and noisy, given that is gathered in a variety of ways [6]:

- Electronic health records (EHRs), which allow for the collection, storage, and analysis of clinical data.
- Patient-reported outcome (PRO) data which can be collected through surveys or other means of patient reports.

- Health information exchange (HIE) systems, which allow for the sharing of clinical data across different providers to ensure access to the most up-to-date information.
- Clinical trials, meaning research studies that involve human participants and are used to evaluate the safety and efficacy of new treatments or drugs.
- Medical wearable devices that collect a set of biological parameters such as heart rates, sleep schedule among others in order to monitor a patient during possibly many days.

2.2 Medical Wearable Devices

Medical wearable devices have become a very promising field that took advantage of the advances in technologies like mobile medicine and smart sensing [17]. These are reasonably small electronic devices that are worn by patients to monitor various physiological parameters and collect data about them. The parameters to measure vary based on the area of medicine that they supplement and the patients themselves. These can be vital signs like blood pressure and heart rate, or circadian rhythms and sleep schedule, or even exposure to light and levels of physical exercise [18].

The data collected by these devices is key for clinicians to have objective and fact based feedback about a patient's health status and symptoms when they are not in a clinical appointment and living their daily lives. This information helps clinicians determine the reasons for poor health status and enables detailed custom suggestions and adjustments to patients routines or treatment regimens to improve their lives.

A common use of these devices is in patients who are receiving chemotherapy, as these treatment regimens can be complex and difficult to manage. The device can provide assistance with self managing tasks such as reminders to take medication, information about a drug side effect and the tracking of the patient's response to the treatment. Studies have shown that the use of wearable devices and mobile health applications helped clinicians significantly improve and relieve patients symptoms like fatigue, nausea and cardiorespiratory endurance during active chemotherapy [19].

A 2020 review [21] discussing the state of the art of this devices used 960 papers as research and concluded that in literature the devices are usually classified in four distinct groups:

- Health and safety monitoring
- Chronic disease management

- Disease diagnosis and treatment
- Rehabilitation

The study concluded that indeed their use is very valuable but pointed out some problems and limitations the industry of wearable devices faced that could prevent their use in a medical setting. These problems concerned mostly with security and privacy of the data, the lack of industry standards and in some cases, issues of non user friendly devices that bothered the wearer and influenced his behavior and normal activities.

Recent developments in the area of wearable devices include improvements in the devices, making them smaller, more comfortable, with longer lasting batteries and more reliable in their measurements. Additionally, there is a growing focus on the MIoT areas like the integration of the medical devices and the data they generate with mobile health applications, telemedicine and electronic health records [20].
The importance and benefits that these devices offer seem to be unanimous. A 2022 paper [22] however, argues that the biggest problem is patients' adherence and willingness to use the wearable devices. The already fragile condition of a disease and side effects of medication are reasons that can make the patients decline another burden and discomfort of wearing a device constantly. This scoping review used a set of multiple papers that covered the use of wearable devices in patients that suffered from various types of cancer, to determine factors that dictate adherence to the devices. The study concluded that the type of interventions and treatments had a big effect on the adherence. The highest adherence was reported in 12 weeks studies and there was a positive correlation between patient reports and wearable outcomes. The authors finish by stating that a "better understanding in intervention standards in terms of the clinical outcomes" would lead to higher wearable adherence and urging for its importance given the positive results that their use provides.

2.3 Data Transfer Protocols

The first steps in the data pipeline are usually related to the collection of data. A common case for data collection is data transferring or sharing of data between two or more entities.
Data transfer protocols can be a very loose term, that can refer to a number of different technologies and protocols such as: HTTP, FTP, SMTP, POP3, Web Sockets, Bluetooth, etc. All of the above mentioned have some data, but are in essence completely different in their area and use cases. In this section we will be looking at protocols related to communication

between a client and a server on a network, commonly used for file sharing, namely FTP and their modern more secure variants.

FTP or File Transfer Protocol is a data transfer method used across the internet, useful for large files. It was one of the earliest protocols intended for this purpose [48] being created in 1971. While it is still being used, it's considered to be not secure. Among the many issues with FTP are:

- Authentication credentials such as password and username are transferred in plain text and the authentication process only relies on these parameters.
- No mechanism to verify integrity of data, which means errors often go undetected.
- Lack of native firewall support can lead to problems when traversing firewalls and Network Address Translation (NAT) devices.

FTP is vulnerable to many cyber attacks [47] such as brute force attacks, anonymous authentication, directory traversal attack, cross-site scripting, etc. To address the limitations and flaws of the FTP, some alternatives were developed. The two most notorious ones are FTPS and SFTP.

[50]FTPS stands for file transfer protocol secure, formerly known as FTP over TLS/SSL, is an upgrade that uses a TLS/SSL layer below the original FTP to encrypt data channels. A session key protects the data in the channel and every message in a session is encrypted with its own key. In implicit FTPS the client connects to a dedicated port where SSL or TLS encryption is immediately enforced. FTPS also provides client/server authentication that can validate the sender or receiver in a data transfer. Client authentication is done using a standard username/password method while server authentication is done by a public key infrastructure such as X.509 digital certificates, that can be issued and signed by a trusted Certificate Authority.

[49]SFTP stands for ssh file transfer protocol and is based around the SSH protocol instead and it also provides secure encryption of data. In this protocol, server authentication is achieved by simply distributing the public keys to clients beforehand and client authentication is done with username/password. SFTP also has the option for multi factor authentication that can be achieved by combining some user information such as a passphrase with a private key and there is even the option to use biometric parameters like fingerprints.

While both options allow for secure transactions of data, and are clear improvements over the original FTP, their protocol implementations are very different. The choice between them should depend on more specific criteria and the concrete use case. Below are some of the relevant characteristics of each as well as some common areas of their implementations[66]:

	FTPS	SFTP
Security	SSL/TLS	SSH
Authentication	Username/password or certificate and public key infrastructure for server authentication	Supports public key, password, and multi-factor authentication for client, and server authentication is done by securely distributing the server's public key to clients
Requirements	Server X.509 certificate and private key	SSH servers usually come with SFTP support
Encrypted	Always if implicit FTPS or with a clear-text phase if explicit FTPS	Always
Configuration	Requires more configurations especially with firewalls and NAT devices	Ease of configuration and firewall friendly
Directory Manipulation	Limited/Require extra configurations	Various commands for manipulation, permissions locking
Usage	Very common due to direct replacement of "legacy" FTP	Common in more recent applications
Connections	2 or more, given that data and commands need different connections	1, data and commands use the same connection
Main Advantage	FTP applications are usually compatible with FTPS. Good when interoperability with a wide range of systems is needed	Wider cross-platform support, ease of setup and use

Table 1 - Comparison between SFTP and FTPS

2.4 ETL

ETL stands for Extract, Transform, and Load. It is usually associated with the beginning of the data pipeline and is composed of a set of processes used to integrate data from multiple sources into a unified data store like a data warehouse or a database [7].

As the name suggests the first step is to extract the data, this process is dependent on the field in question, given that data can come from very different types of sources [8][9]. In fact, modern digital technologies generate an abundance of data, and the challenge in a lot of cases is to know what data to use and to store. In this document and project, the main source of data is wearable devices worn by oncology patients, and in contrast to a lot of data driven applications, there is not an abundance of data.

The transform refers to the acts of cleaning, normalizing, aggregating and validating the data. Many of the choices made in this section must have in consideration the purpose and end goal of the consumer of the data. Perhaps in certain applications, errors or outliers must be corrected while in others the outliers could play a major role. In some data pipelines there may be the need to merge data from different sources or to convert data types or even check some of the observations for accuracy, etc. Any operation done to the data in between extraction from the source to the loading of it, is considered a transform and are typically performed using technologies such as SQL for data queries and Python or R for data manipulation, among many others.

Finally, the load step simply refers to the process of loading the now clean and good data to a data store which is a centralized location where data is saved, organized, and accessed by analysts, data scientists, business users, etc. The store can be a data warehouse, a data lake or even a simple database. Again, the amount of data and its purpose must be considered to make the most appropriate choice.

In big enough organizations it is usual to make the distinction between the operational system and the data warehouse [10]. Data stored in the operational system is used in the daily core activities of the organization, therefore these systems have a transactional nature and prioritize real time access and performance. A data warehouse, on the other hand, is a system that is used to store and manage historical data for reporting and analysis purposes which means it must support large amounts of data and complex queries, since the data stored in this fashion is used for reporting and analysis of previous behavior of the organization's activities.

In literature it is common to make another distinction between two very popular ETL types: [11] [12]

- Batch ETL that refers to the act of processing and loading the data periodically into the data store. Usually done when data is accumulated over time and in large quantities.

- Streaming ETL is the process of real time continuous processing of data as it is being generated by its sources. Its use is appropriate when data is generated in high volume and velocity. Data produced by sensors, social media or IoT applications are the most common applications for this type of ETL or any other application that requires real time analyses or event driven triggers.

A more recent development in this field has been motivated by an increase in volume, velocity, and variety of data being generated which resulted in a shift towards cloud based data stores [13]. These solutions allow for a more scalable and cost efficient ETL process that doesn't require users to invest in their own data centers. The most popular cloud service providers in the industry are, in no particular order:

- Google Cloud Platform (GCP)
- Microsoft Azure
- Amazon Web Services (AWS)

Delegating storage responsibilities to a third party has benefits in scalability, given that most solutions allow to adjust capacity as needed and also in accessibility as everyone with the right privileges can access the data stored from anywhere. Another strength of cloud storage is the redundancy backups and data recovery mechanisms these providers usually offer that otherwise would have to be accounted for. On top of that, the before mentioned providers also invest heavily in security measures to mitigate the risk of data breaches, unauthorized access and cyberattacks in general. More on ETL cloud applications in the next section.

A modern trend in the area of ETL is the emergence of artificial intelligence and machine learning to automate processes [14]. Profiling data, error correction, outlier detection, or any kind of data operation can now be done using these algorithms as well as the pattern recognition and post ETL analyzes to find patterns within data. Automating the ETL step has clear benefits of not relying on human manual intervention and coordination between operators that is typically required in data warehouses. In this case study example [14] the author's purpose an automated ETL process that can manage the very large quantity and variety of data in the fields of marketing, retail and financial services by using various machine learning based methods to preprocess data before loading it into a data warehouse resulting in a near real time delivery of the required reports.

Another trend in business driven use of data is the use of self-service ETL tools that allow non-technical users to perform ETL tasks without the need to code or the need to work with IT or data engineers. Some of these tools rely on user interfaces that can be navigated easily by anyone. This separation between technical knowledge and the business decision

making is a feature desired by some who argue that the quick access to information by business personnel in big organizations is key to taking the correct action and decision in real time [15].

In some cases, such as the healthcare and medicine fields, the data can have a sensitive nature and some level of privacy is required. Common ETL workflows don't usually provide or can be integrated with anonymization tools. A project conducted in 2019, [16] tried to implement expert level anonymization in ETL workflows.

Expert level anonymization is a method used to protect individuals and mask personal identifying information while still maintaining the data usable.

The team in question was successful in creating a plugin for the Pentaho Data Integration tool that "enables integrating data anonymization and re-identification risk analyses directly into ETL workflows".

Indeed, when working with data, ethical considerations and legislations regarding privacy and ownership of information play a big role. As mentioned previously, the use of data for research and healthcare purposes must follow strict guidelines and regulations. Usually the patients must consent to have their data stored and studied, and in the case of this work the patients in question have an understanding about the process that ultimately serves to improve their quality of life.

2.5 Cloud and Serverless Computing

In the last 15 years the use of the cloud has become increasingly popular and the COVID-19 pandemic even accelerated this process as remote work became necessary. Many businesses are reporting huge shifts in their approach to use more cloud services and this has made the market cap for cloud providers to be over 450000000000$ (four hundred and fifty thousand million) as of 2022[51].

"The cloud" simply refers to the use of computing services through the internet without the need for a dedicated infrastructure (from the perspective of the user, as the provider obviously has huge infrastructures). This technology is particularly useful in scenarios that require high scalability and unpredictable resource usage.

In 2010 [32], the cloud was mostly aimed at system administration. It made computing infrastructure easier to configure and manage through the use of virtual servers and networks made from massive and numerous data centers.

More recently, a new trend has emerged that provides high level abstractions for web and app developers to simplify the cloud environment by "hiding" the server. This trend is called serverless computing, even though the remote servers are still very much in place. An example of serverless computing is cloud functions that allow functional pieces of code in a multitude of languages to be called in response to web requests or the triggering of events. Nowadays cloud services provide a huge amount of different services, such as computing power, data storage, container orchestration, dev ops tools, artificial intelligence integrations and many more.

Cloud services can be generally divided into three categories, depending on the service they provide:

- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)

In IaaS models [52], the provider lends hardware capacity such as memory, storage or processing to a client that usually pays in a pay-as-you-use system. This is done mostly through big data centers with remote servers. This model is very useful for scalability purposes given that no hardware improvements are required once higher capacities and volume increase, since the provider can adapt to it and charge accordingly. Businesses commonly use these virtualized computing assets to host websites or applications and to manage development, testing and production environments. Some examples of this type of model are Microsoft Azure and Google Compute Engine.

PaaS [53][54] alludes to a model where the provider hosts application development platforms or tools on its own servers and rents them to developers. The big benefit is that the development team only needs to worry about their own code regarding applications, while updating and maintenance of the infrastructure are responsibilities of the provider. On top of that, some platforms can also have useful built in tools that developers can use on their apps. This type of solution is very useful for web and app development and for API hosting. Examples of this type of model are AWS Elastic Beanstalk, Heroku and Google App Engine.

Finally, SaaS [55] providers offer distribution models that host complete software applications, making them accessible via the internet. Users can access fully developed services without needing to manage or be aware of the underlying infrastructure. This model is prevalent in modern large-scale applications and websites. Notable examples of SaaS providers include Salesforce, Netsuite, Netflix.

Given the success and high adoption rates of this technology, the number of Cloud providers and serverless platforms has increased. More recently in literature the terms FaaS (Function as a service) or BaaS (Backend as a service) have been commonly used.

BaaS [56] refers to services that act and substitute traditional backends. These can store and query data, deal with authentication processes, notifications and basically everything that a regular backend server could be implemented to do. A well known example is Google Firebase. This service makes processes such as user management from an admin perspective much easier to do.

FaaS just refers to services that execute code in response to events. These cloud functions can also be used as backend services and act like API points that are triggered by the client's incoming HTTP requests.

Some of the concepts mentioned above may have overlapping definitions; however, the renowned cloud providers offer comprehensive solutions that encompass all of these aspects. It's a testament to the versatility and interconnectedness of cloud services that businesses often find themselves utilizing multiple offerings from these providers. These services are designed to work cohesively, forming an integrated ecosystem that empowers organizations to build, scale, secure, and innovate with remarkable efficiency. Whether it's managing infrastructure, developing applications, analyzing data, or implementing advanced AI capabilities, the interconnected web of cloud services provides the flexibility and agility that modern businesses require in today's dynamic digital landscape. [57]

[4]

Figure 2 - Overview of some Cloud Solutions provided by Google

[4] Figure 4 provided by cloud.google.com

2.5.1 Cloud computing in ETL processes

As briefly mentioned before, the use of cloud and serverless technologies applied to ETL processes are a modern and viable alternative to traditional ETL. This approach offers several advantages that businesses find compelling such as the scalability given that the resources used can be dynamically allocated, guaranteeing good performance even when data volume drastically increases. Most of the time the payment plans provided by these cloud solutions are also dependent on the amount of traffic or usage they receive and commonly follow a pay-as-you-go model that allows an entry cost that can be more reasonable as it eliminates big upfront costs that would be associated with the infrastructure for more typical models. High speeds and flexibility are other benefits of cloud ETL solutions. Most of the available tools easily manage a wide range of data sources and destinations at high-speed. These ETL services offered by cloud providers also reduce the operational burden on IT teams given that most of them offer some degree of automatization.

A 2018 paper [33] research, argued that cloud solutions are desirable for big data ETL processes, given that the variety, volume and velocity of data characteristic of big data would require "unlimited" computing resources that can be provided by cloud solutions. However, the authors also pointed out that the pay-per-use nature of the cloud could be a problem as the volume of data increases and that new ETL strategies are needed to face big data.

Today, the reality is that cloud computing and serverless computing, already offer scalable and non cost prohibitive solutions to ETL processes. Some examples, among many more, are:

- Hevo Data[5]
- Google Cloud Data Fusion[6]
- AWS Glue[7]
- Skyvia[8]
- Microsoft Azure Data Factory[9]
- Fivetran[10]
- Integrate.io[11]

[5] https://hevodata.com/
[6] https://cloud.google.com/data-fusion
[7] https://aws.amazon.com/glue/
[8] https://skyvia.com/
[9] https://azure.microsoft.com/en-us/products/data-factory
[10] https://www.fivetran.com/
[11] https://www.integrate.io/

Some of these are SaaS solutions which means users only need to have a decent internet connection, while others don't even require to code and are full data pipelines as well.

2.6 User Interface in Web Applications

With Cloud Computing being a part of modern solutions regarding applications' back end infrastructures, there is still a need to develop the application interface to interact with the user. This type of paradigm is usually referred to as front-end development. Front-end development involves designing and creating the pages of an application using a language known as HTML (Hypertext Markup Language) and implementing the functionality with a programming language running directly in the user's browser (in the case with web applications). While numerous programming languages can be employed for this purpose, JavaScript stands out as the most prevalent choice. However, contemporary front-end development rarely relies solely on vanilla JavaScript and instead, developers often opt for a JavaScript framework or library. These frameworks are really different from one another and it is important to make an educated choice, given that migrating an entire application to a different framework may not be feasible.

A study made in 2020 [36] compared the most common JavaScript frameworks:

- React JS[12]
- Angular JS[13]
- Vue JS[14]
- Svelte JS[15]

on a variety of different parameters like loading times, typescript compatibility, depth and ease of use of its online documentation, among many others. In the end, the author ranked React as the number one framework to use, mostly due to its popularity and indeed the CLARIFY web application mentioned in this document is written in React with Redux. However, the paper was written in 2020, and every framework in question has been updated and improved since. In practical terms and in agreement with the mentioned paper, the performances of the most popular frameworks are really comparable and the choice should come down to preference as the landscape of Javascript frameworks and front-end in general is notorious for the rapid

[12] https://react.dev/
[13] https://angularjs.org/
[14] https://vuejs.org/
[15] https://svelte.dev/

development of new frameworks and updates to the existing ones which makes it difficult to keep up with every novel trend.

An important and fundamental concept in most of these frameworks is state management. This refers to the process of maintaining the knowledge and data of an application´s inputs and state across multiple related data flows across a session in order to better understand the condition of the app and its components at any given moment. As modern web applications grow in complexity, the need for efficient state management becomes increasingly pronounced. There are many libraries that implement state management in a simple and compatible way with the frameworks mentioned before and the choice of a state management tool often aligns with the framework being used. Among the most popular choices are Redux, MobX, Zustland, etc. Without state management the different components of the interface would need to send their props (data) to each other. This can quickly become very complicated once an application reaches an appreciable size, and state management provides the ability for each component to access the needed resources from a centralized source.

The CLARIFY web application uses Redux[16] as the state management tool. It provides a centralized (on the client browser) store to manage the application's data and state changes. The main benefit is allowing developers to maintain a clear and structured flow of data, making it easier to debug and maintain complex applications and it significantly enhances the scalability of the front-end architecture.

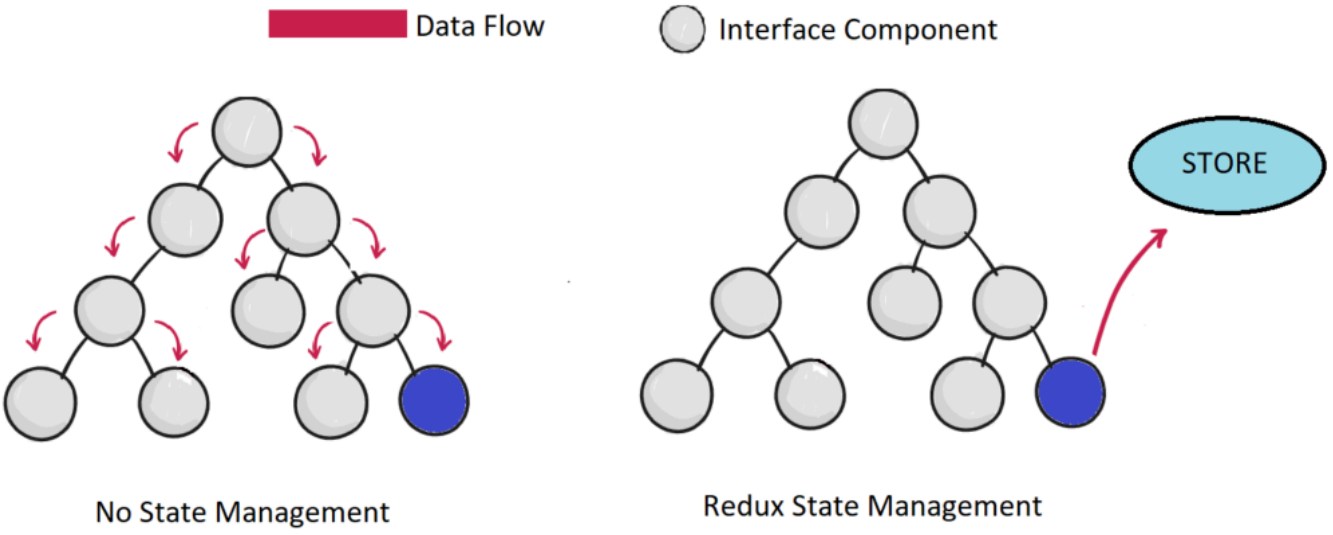

Figure 3 - Comparison between data flows of an UI without state management and with Redux state management

[16] https://redux.js.org/

2.7 Artificial intelligence and Data Visualization

2.7.1 Machine Learning

Machine learning algorithms are artificial intelligence computational techniques that enable computers to learn and make predictions or support decisions without being explicitly coded but instead by finding patterns in data. Machine learning can basically be categorized as supervised or unsupervised although there are some nuances that will be discussed.

This process is usually the last in the data pipeline and requires clean data, usually provided by the ETL step. In traditional programming, the computer is provided with a set of data or inputs and the definition of the algorithm that informs the computer on the operations to do with the input, to then generate another set of data, known as the output. In supervised machine learning there's a shift in paradigm: the computer has initial access to the dataset and their associated outputs and generates the algorithm that best describes the relationship between the two [23].

This constantly evolving technology is viable in a variety of different fields and is able to perform very complex tasks. From mastering the game of chess, to self driving cars or even learning the laws of physics from experimental data, there seems to be a presence of this new technology across all areas with very impressive results. Medicine and healthcare are no exception, and there are a lot of problems in these fields that can be tackled using artificial intelligence approaches. In the literature, machine learning techniques are split into various categories. The most prevalent are [24]:

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Reinforced learning

Understanding the distinction between supervised and unsupervised learning is crucial. Supervised learning involves the use of labeled data, where the algorithm is provided with the correct answers for each instance. In contrast, unsupervised learning deals with

unlabeled data and focuses on discovering hidden patterns through techniques like data clustering.

Supervised learning finds extensive application in predictive tasks, such as classification and regression. Classification problems try to assign test data into categories and learn what factors and with what weight they matter for classification. An example would be to predict if an image has a cat or a dog or if an email is spam or not. Given that the data is labeled the algorithm has direct access to the answer of the class and can find the stronger correlations to other variables. It is possible to then determine the accuracy of the algorithm. As with other types, the learning and testing data should not be mixed for a fair assessment of accuracy. Regression problems, another form of supervised learning, are useful for predicting numerical values and understanding relationships between dependent and independent variables. An example would be to forecast sales revenue in a company. There are many different supervised learning algorithms. Some of the most common are:

- Decision tree
- Naive Bayes Classification
- Support vector machine for classification
- Linear and logistic regression
- Random forest for classification
- K-nearest neighbor for classification

In healthcare, supervised learning algorithms can be used to predict diseases such as cancer relapses. A study done in the University of Sydney [25] identified a large number of papers that used supervised learning in medicine for disease prediction and compared the accuracy of each algorithm. The authors found that Support Vector Machines (SVM) were the most used algorithm followed by Naive Bayes, however the Random forest algorithm showed the highest accuracy in most of the papers.

To contrast, in unsupervised learning there are no outputs or classes to predict. The algorithm tries to find natural patterns and clusters within the data points. This is useful in a lot of scenarios where subject matter experts are unsure of properties within the data set. Semi-supervised algorithms are used when only part of the data is labeled. Both this and fully unsupervised approaches tend to be more time efficient given that they dont require labeling data. Unsupervised learning can be divided into two types of problems: clustering and association. Clustering refers to grouping data points into clusters such that points in a given cluster share commonalities with other points in the same cluster and don't share with points from different clusters. Association clustering is used to find the relationships between variables in the dataset and determine the set of items that occur together. Real cases of unsupervised learning use include customer segmentation, content recommendation systems,

medical imaging, anomaly detection, among others. Some of the most common algorithms of this type are:

- K-means clustering
- Principal component analysis
- Hierarchical clustering
- Neural networks

These algorithms are also very popular in the many fields of medicine: A study [26] found use for unsupervised learning algorithms to group and classify multiple sclerosis subtypes based on pathological features according to data provided by brain MRI scans. Another example of a completely different use, was a study done in Echocardiography [27] that used unsupervised learning for a "diastolic function classification and risk stratification using the left ventricular diastolic function". In oncology a popular use of this technology is to identify gene signature for specific types of cancer. Genomic profiling can give an upper hand in predicting cancer. In a 2022 project [28] a team used a support vector machine for clustering based on "expression levels" of 32-gene signature for gastric cancer. The team managed to identify four molecular subtypes that predicted survival.

Reinforcement learning refers to algorithms that interact with the environment and are encouraged to take some desired behavior and punished to take an undesired one. Common examples are natural language processing, image processing, finance applications and many others. In healthcare these algorithms are used in dynamic treatment regimes (DTR) for chronic illnesses to create optimal treatment recommendations for individual patients [29]. Other healthcare applications involve automated medical diagnosis [30] , drug discoveries and interactions or even resource scheduling and allocation.

2.7.2 Neural Networks

Neural networks [58], also referred to as artificial neural networks (ANNs), are a class of artificial intelligence models inspired by the structure and function of the brain. These computational systems consist of interconnected nodes, known as neurons or perceptrons, organized into layers. There is a flow of data that starts with an input layer and passes through the hidden layers culminating in an output layer. The number of neurons in each layer is very dependent on the use case and specific ANN implementation. The input layer has as many nodes as the input variables while the number of nodes in the output layer depends on the task that the ANN is trying to solve. In classification problems, it is common to have as many output nodes as the number of possible classes, however in other applications that may

not be the case, for example in regression tasks it is possible to have only one node with the predicted output. The number of hidden layers and the number of nodes in each hidden layer are the essential architectural decisions that need to be made when designing a neural network, as they can significantly impact the network's performance.

The number of hidden layers in a neural network is referred to as its depth. It is usual to refer to ANN with a low amount of hidden layers as shallow networks and those are usually preferred to solve simpler problems. If an ANN has a large number of hidden layers it is referred to as a deep neural network [59]. These are capable of learning complex, hierarchical representations and are well suited for tasks involving intricate patterns, such as image recognition, natural language processing, playing high level chess, and much more.

The number of nodes in a single hidden layer is often referred to as the width of the layer. The choice of the number of nodes in a hidden layer, also known as the layer's width or size, is influenced by several factors such as the complexity of the problem (more complex problems require a higher width), the amount of available training data and the computational resources available. It is important to note that more width is not necessarily better, as if the network has too many nodes in the hidden layers for the amount of available data, it may overfit the training data and not behave as well in unseen novel data (which is ultimately the goal).

Neural networks arrive at a desired solution by adjusting the strengths of connections between the nodes (or neurons), called weights, during the training process. There is also a non linear process in each node called an activation function. These are crucial component neural networks that serve as mathematical operations applied to the input of each neuron to determine its output. There are many possible activation functions to use and all of them are non-linear which enables the ANN to model complex relationships in data. The choice of an activation function is another parameter to determine the architecture of the neural network. Among the most common activation functions are the Sigmoid, the Hyperbolic Tangent, the Rectified Linear Unit, the Scaled Exponential Linear Unit, and many more.

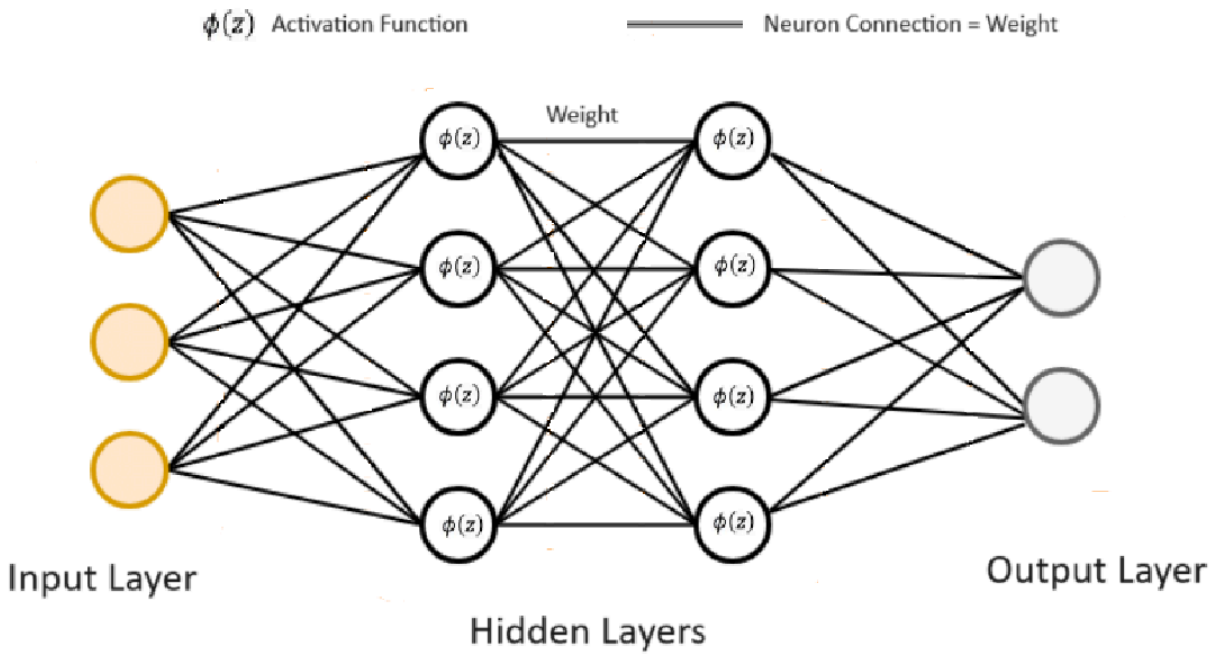

Figure 4 - Typical neural network with a sigmoid activation function

The figure 4 is a visual representation of a typical artificial neural network. In this example the input layer has 3 nodes (or neurons) which means that the model only takes 3 variables into consideration. There are also 2 hidden layers with 4 nodes each and each node has an activation function (the Sigmoid is represented) and finally an output layer with 2 nodes. The final model, once trained, is just the set of weights that represent each node connection.

There are many types of different neural networks that vary on their applications and inner workings. Some of them are a lot more nuanced and complex than the simpler Feedforward Artificial Neural Network described above. To name a few:

- Perceptron and Multilayer Perceptron (MLP) Neural Networks
- Radial Basis Functions (RBF) Artificial Neural Networks
- Recurrent Neural Networks (RNNs)
- Modular Neural Networks
- Convolutional Neural Networks (CNNs)

The detailed study and differentiation of each of these examples is outside the scope of this work, however it is worth mentioning that the Multilayer Perceptron is a very valid and common choice in classification problems in the healthcare domain. As mentioned previously in the section 2.1 of this document, a 2019 paper [43] showed how a MLP could be optimized to achieve very interesting results in healthcare prediction and classification.

Indeed, literature shows [60] that ANN are widely used in healthcare and biomedicine for various purposes, including medical image analysis, disease prediction, drug discovery, genomics research, and more. ANN have proven to be effective tools in handling and analyzing the complex nature of biomedical data.

Another example of its usage can be found in a 2021 research paper [61] that applied deep neural networks for the classification of protein localization in various cellular subcompartments with the objective of accurately identify the locations of proteins within these subcompartments, as the authors say this is a crucial crucial task in understanding protein function and interactions. For this task, the researchers tested convolutional neural networks (CNN) and fully convolutional networks (FCN) on a data set with 20,000 confocal microscopy images with annotations for 13 cellular subcompartments where target proteins are expected to reside. It was found that both neural network algorithms perform well in classifying major cellular organelles, however, FCN outperformed CNN when it came to images with multiple simultaneous protein localizations. Among other difficulties, the authors noted that class imbalance in the dataset can affect classification accuracy, with rare classes being more prone to misclassifications. This is usually the case with most Artificial Intelligence algorithms. In the end, the study concluded that these types of ANN are useful for their classification requirements and they also suggest the potential benefits of ensemble models combining the strengths of CNN and FCN for improved classification accuracy. This composite model would ideally combine the classification capabilities of CNN and the localization capabilities of FCN to enhance overall performance as this approach would take advantage of the unique strengths of each model. The researchers did not implement such a model in the mentioned paper.

2.7.3 Data Visualization

At times, data utilization focuses on addressing simpler objectives where the application of artificial intelligence methods may not be necessary, and the use of data visualization techniques alone can suffice to meet the desired goals. Data visualization algorithms are also widely used in computer science, engineering, business decision making and healthcare. The ability to access information in a timely manner on a very large set of data is one of the main requirements in most data driven applications and is crucial for scientific research and statistical understanding of a given data set.

The most common data visualization techniques are: [31]

- Line Graph
- Bar chart

- Scatter plot
- Pie chart
- Radar chart
- Tables

Figure 5 - More Examples of data visualization techniques

All of which can be implemented using high level programming languages such as Python with packages and modules such as matplotlib or Seaborn commonly used to build business

and research reports, or JavaScript with modules such as ChartJs or FusionCharts used to develop web applications.

2.8 Other Related Projects

In order to prepare to implement the features in the application in question, it's important to learn and research previous projects that can have some amount of similarity to ours.

As far as tools or platforms to aid patients in the follow up of surviving cancer, an online search can quickly find web sites and communities where cancer survivors and their potential caretakers are free to join and connect with other people in similar circumstances. "Cancer Connect[17]" and "Cancer Support Community[18]" are good examples that seem to have a big and active community of members. This type of offering is aimed at the patients themselves and not at clinicians, and it has little to no resemblance to CLARIFY.

Another project named "Cancer Care Point[19]" offers free consultations to cancer patients and cancer survivors for the purposes of assistance, counseling and education to the patient's condition. Again, this is totally different from the continuous and data based approach meant for clinicians that CLARIFY has.

A 2021 project [34] compared social relationships between a control population and young adult cancer survivors. The authors conducted a series of online surveys and concluded, unsurprisingly, that cancer survivors had the worst patient-reported outcomes and experienced greater feelings of loneliness than the control population. The authors finish by adding that social integration is a real issue for cancer survivors that warrants intervention but currently no real practical solution is implemented. Indeed, the follow-up oriented nature of CLARIFY, that uses patients data to continuously monitor and improve their lives, seems to be rare.

The most similar and relevant project to mention is FAITH[20] [39]. This project is also EU-funded and has close links with CLARIFY. FAITH uses federated learning, which is an artificial intelligence technique, on data shared by hospitals or other relevant entities to monitor depression levels on cancer survivors and inform the patients point of care in case they decline. Similarly to CLARIFY, FAITH is applying data mining and machine learning to

[17] https://news.cancerconnect.com/
[18] https://www.cancersupportcommunity.org/
[19] https://www.cancercarepoint.org/
[20] https://www.h2020-faith.eu/

better the follow-up process of survivors. This app focuses on what their authors call targeted depression markers, while CLARIFY follow-up is more based around circadian rhythm and other stats such as data obtained through medical wearable devices. The development, technologies and motivations behind both these projects share a lot of similarities: The improvement of post cancer treatment follow-up, the ETL process of data, the application of AI techniques, the presentation of results to the clinicians that need the information and even the privacy and ethical considerations regarding the use of private and sensitive data.

Even if projects with the same goal as CLARIFY or FAITH are hard to come by in literature, certain parts of their implementations can readily be found. For example, tools that collect some sort of patient data to build learning models about a disease are a lot more common. An example is a project called "Patient Crossroads[21]", that collects patient registry data on orphan diseases into a single data repository to serve as research with the purpose of finding non-obvious relationships between genes and diseases. Unfortunately, the project web page doesn't elaborate more and no meaningful references were found there for further analyses.

As far as medical wearable devices, their use is already a staple in healthcare and many examples can be found: a famous example is the use of glucose trackers to monitor diabetes that also include automated insulin delivery systems. "Eversense" is the first FDA approved implantable device that accomplishes just that. Other features present in CLARIFY, such as drug interaction tool, cancer relapse predictor and cancer population analyses are more common in literature and it's easier to find examples of existing tools.

While this research that aimed at finding similar projects to the one described in this document is valuable to understand the landscape and state of the art on the post cancer follow-up area, it wasn't particularly helpful in the developments and implementations made in CLARIFY addressed in the following section of this document.

[21] https://treat-nmd.org/organization/patient-crossroads/

3

DEVELOPMENT

3.1 Overview and Architecture

As mentioned in the introduction, every development done and mentioned in this document aims at improving or creating new features to the CLARIFY web application more specifically regarding the analysis of data produced by the Kronohealth wearable. The main objective is to compare a set of variables relating to the circadian rhythm of a patient that went through oncology treatment, with a control population of 10 000 healthy individuals, also provided by Kronohealth. This information is made available in the web application to be consulted by the patient's doctor/clinician in order to help decide on interventions and provide suggestions to the patient's lifestyle based on concrete data.

The Clarify web application is hosted in the Google App Engine (GAE). This is Google's Platform as a Service (PaaS) offering, that allows app deployment and hosting without the need to manage the underlying infrastructure.

The proposed implementation can be divided in a few separate, although connected and coherent, steps:

- Develop a mechanism to import and prepare all the data related to Kronohealth into the respective storage methods in order to be accessed in the Clarify application.
- Develop the backend functionality to access the necessary data and make it available to the user interface.
- Develop the required user interface components to show the Kronohealth wearable device data

- Using Classification methods, study how certain variables can be used to predict recommendation of the clinical interventions of a patient.

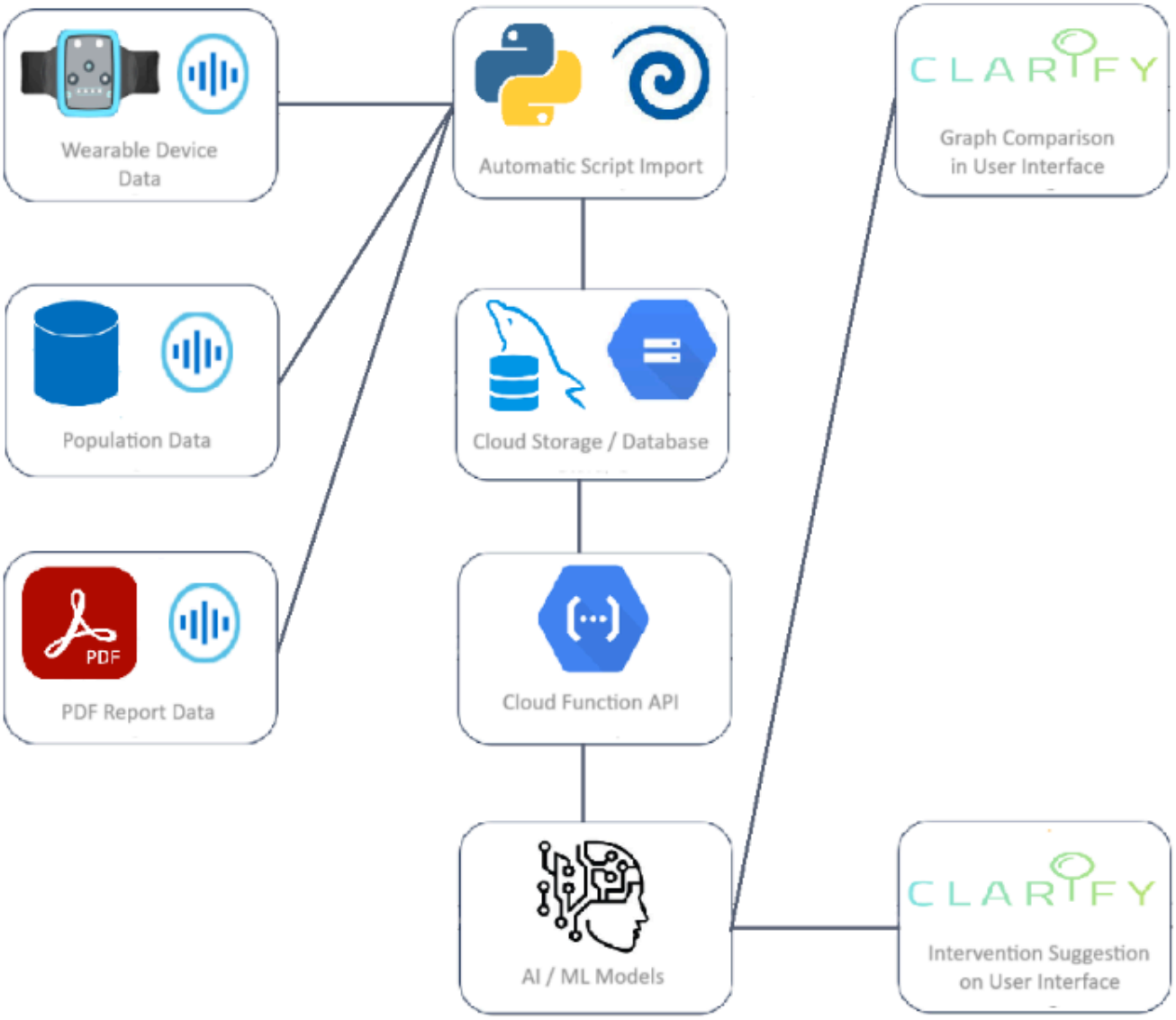

Figure 6 - General Architecture of the proposed developments

The figure 6 shows a graphic representation of architecture related to the Kronohealth section in the CLARIFY application. The process of collecting data from the wearable devices, the patient's PDF reports and the data from the control population were all handled by Kronohealth.

3.2 Automatic Import of Data

The first task tackled was to create an automatic way to import data sent by Kronohealth into the appropriate locations. Previously, Kronohealth would send HOLOS and CLARIFY developers the data via email, which then had to be manually imported. The goal of this task was to be able to automatically receive the data and integrate with the application without the need for any extra human intervention. It is important to note the different kinds of data that Kronohealth provides in this regard:

- PDF with medical circadian report (PDF file)
- Circadian variables average values (JSON and CSV files)
- Time series circadian variables values for an average day (CSV file)

Every CLARIFY patient that underwent Kronohealth wearable device monitoring will generate the above mentioned files of data.

As the data from different files serve different purposes in the user interface, they are stored in different ways: The PDF report file along with the JSON and CSV files regarding the average values of the circadian variables are stored in the Google Cloud Platform's Bucket Storage section. There, every patient has its own subfolder with their respective information and files.

The time series variables are a larger set of data, given that every 10 minutes the device takes a measure of every circadian variable. These are stored in a MySql database. There is a table that maps each CLARIFY pseudonymised patient ID to the corresponding Kronohealth ID, and then a table for every circadian variable with every patient and their values.

The circadian variables are:

- TAP or Thermometry, Actimetry and Body Position
- Sleep
- Light
- Time of movement
- Intensity of movement
- Temperature

In order to automate the process, it made sense to suggest another way to transfer the files. As mentioned in the state of the art section of the document, the main contenders for this purpose are SFTP and FTPS.

Given that in this situation no clear benefits of FTPS were apparent, the choice for the more modern protocol in the SFTP was easy. For that purpose, an SFTP server was set up in a virtual machine in Google Compute Engine (GCE) with a corresponding user for the Kronohealth team. GCE is a cloud solution for Infrastructure as a Service (IaaS) and all the computing and logic regarding this aspect of automatic import of data was implemented there.

Once established a more convenient way to receive data, a cron job, running on the same virtual machine calls a script every 10 minutes that checks if a recent file has been received and in that case what type of file. Knowing what kind of file has been received is important as their destination is different.

Two types of scripts were tested: a Python script and a Pentaho Data Integration[22] script. For the required purpose of identifying the type and destination of the file to the actual treatment of the data and subsequent load, both options work well. In the end the Python script was the selected choice, given that it is easier to maintain and eventually update if other types of data are later required. This is simply because more developers are familiar with this technology and the documentation is more comprehensive.

The script uses the "os" and the "glob" libraries to access the files in the directory, and to compare their extensions and names. The files intended for the GCP bucket use the python "Google Cloud Storage" package. Authentication within the GCP project is done through a JSON key file with writing permissions for the Kronohealth bucket. For the file containing the time series variables, intended for the MySQL database, the "Pandas" library was used to access the files variables and values and load them into a Pandas dataframe. This allows for much easier manipulation. There was no need to make any major data transformations as the data sent by Kronohealth was of very good quality, meaning no missing values or obvious errors. The data was all in the same format, clean and ready to be used. To load the data to the desired MySQL database the python "mysql connector" library was used.

This combination of the SFTP server set up with a cron job to call the importing python script allows for the desired outcome of integrating novel data sent by Kronohealth directly into the storage solutions which make the data readily available in the web application.

The figures 7 and 8 show the result of the imported data. In figure 7 we can see a sample of the Kronohealth bucket in Google Cloud Platform on the left with 4 visible patients and the corresponding files for one of the patients on the right. Figure 8, displays a sample of the time

[22] https://www.hitachivantara.com/en-us/products/pentaho-platform/data-integration-analytics/download-pentaho.html

series variables stored in their respective table on the MySQL database. For this type of data, the population average values were also stored in the same fashion as the patients.

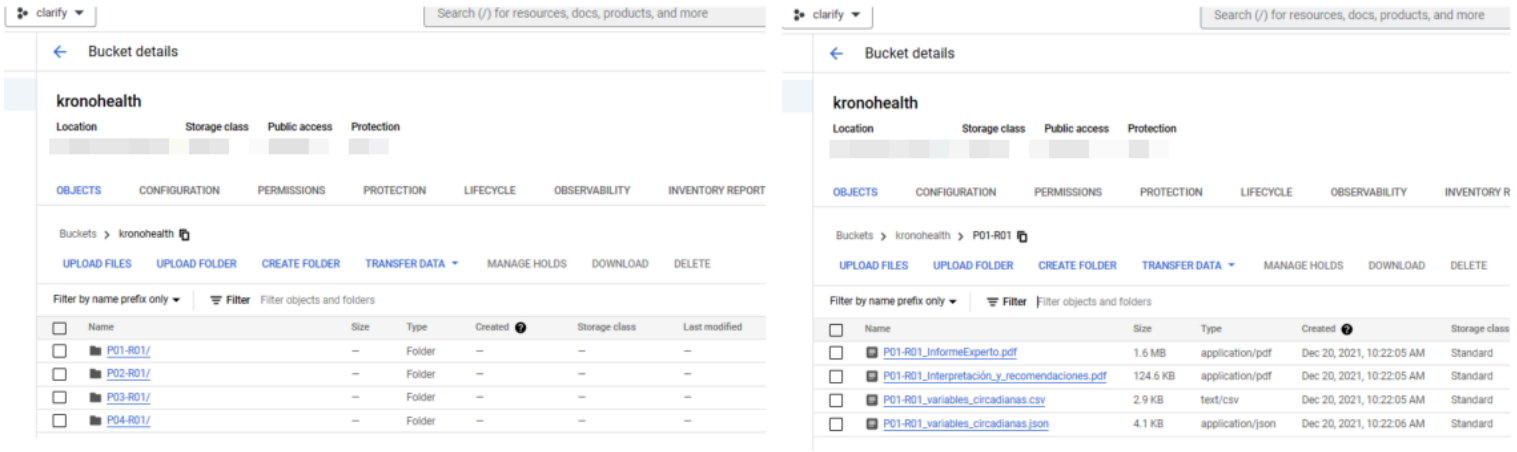

Figure 7 - Google Cloud Platforms Storage Bucket with a few patients as example

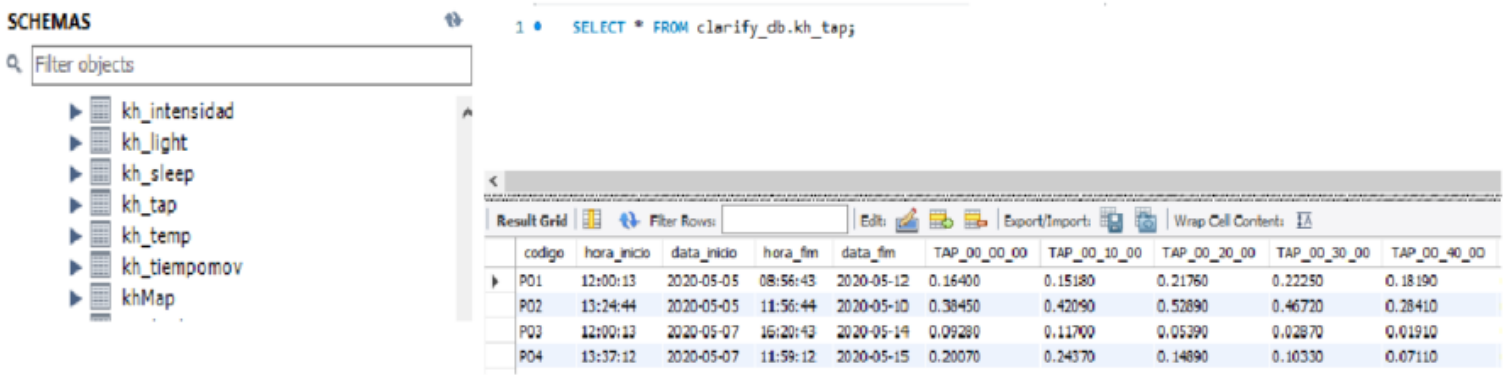

Figure 8 - MySQL tables with the time series values for the circadian variables

3.3 Backend development

The Backend (or server side) processes of the CLARIFY web application are all addressed using Google cloud solutions, namely FireBase and Google Cloud Platform (GCP).

In order to accomplish the proposed requirements of displaying interactive comparison graphs in the user interface of the application, there needs to be a way to get the data from the storage to the client side of the application. This was done using GCP cloud functions, which is Google's Function as a Service (FaaS) solution. These are a serverless computing service that allows to build and deploy individual functions and pieces of code that can be executed in response to events or triggers. In the CLARIFY web application architecture these functions act as the typical server-side portion that deals with requests from the client and communicates with the storage methods to fetch the necessary data.

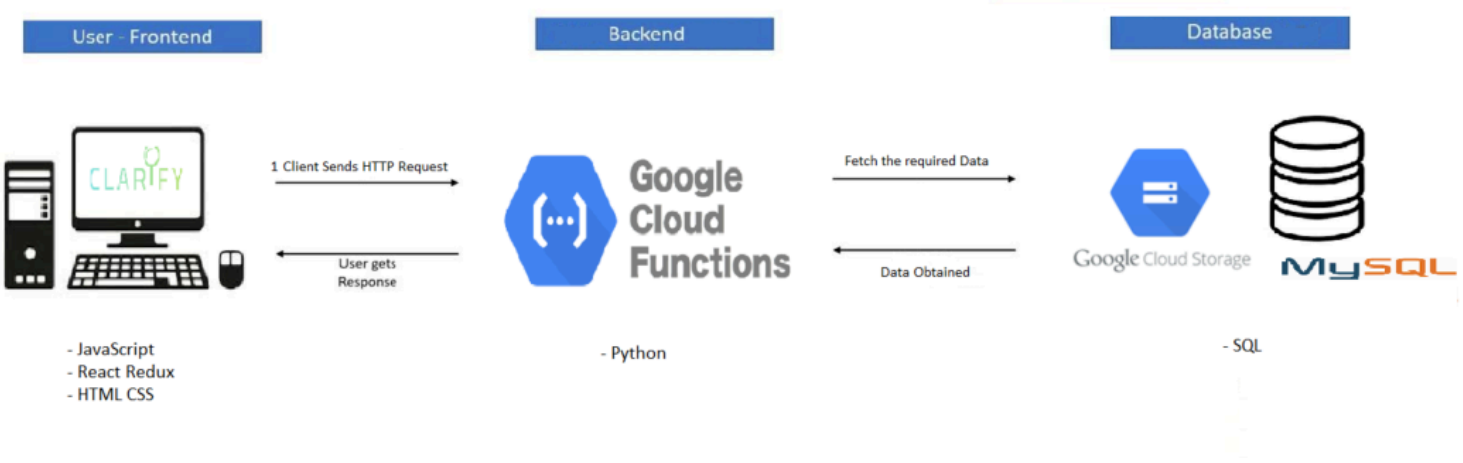

Figure 9 - Flow UI- Cloud Functions- Storage

The figure 9 shows a typical flow of data: the Kronohealth cloud function receives an HTTP request from the user client with the patient ID. The cloud function communicates with the database to map that patient identification to their respective Kronohealth ID and then fetches all of the patient's Kronohealth data from the database and the cloud storage bucket. It also fetches the control population data. Once the function has access to the data, it makes some light manipulations to ease the access and use in the React client side portion (Frontend), and responds to the original HTTP request with the data.

The Cloud Function is written in Python language and uses the standard Firebase packages to authenticate the request. This stops the function from being randomly accessed by anyone. In order to authenticate the function to access properties of the Google Cloud Platform

CLARIFY project such as the storage buckets, the standard “google.cloud” packages are also used. Finally, to connect and access the MySQL database the “SQLAlchemy” library was used.

Using cloud functions as the backend of the app has many advantages. As already mentioned, the serverless architecture delegates the management and infrastructure responsibilities to Google which frees the developers to simply write the necessary code. It is also ideal to follow the microservice architecture where the code is simple and limited to a single functionality. In this case every feature of the CLARIFY app that requires server side or database resources has its own independent cloud function.

Another key feature provided by this approach is the monitoring, logging, debugging and version control that each cloud function has. The figure 10 shows the dashboard of the Kronohealth function in the metrics section. Here it is possible to control the amount of times the function is being invoked, each of the execution durations, the memory usage and possible errors.

There is also a version control at the top that allows navigation back to previously implemented versions of the function. Every time a developer deploys or updates the code Google Cloud automatically creates a new version. This makes the development process much easier and safer for developers.

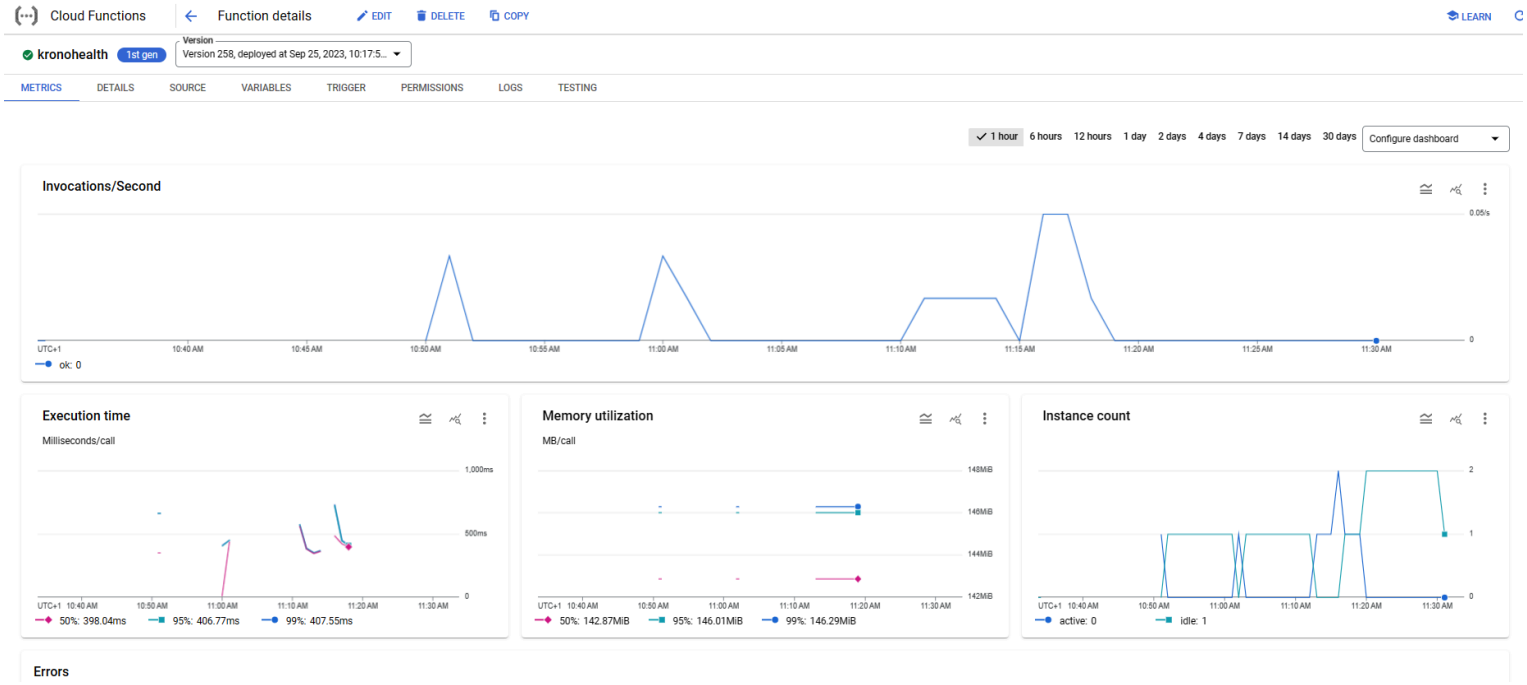

Figure 10 - Dashboard of a Google Cloud Function, Metrics tab

The debug process is also facilitated with the TESTING and LOGS sections of the dashboard. The figure 11 shows a small sample of logs generated by the Kronohealth function.

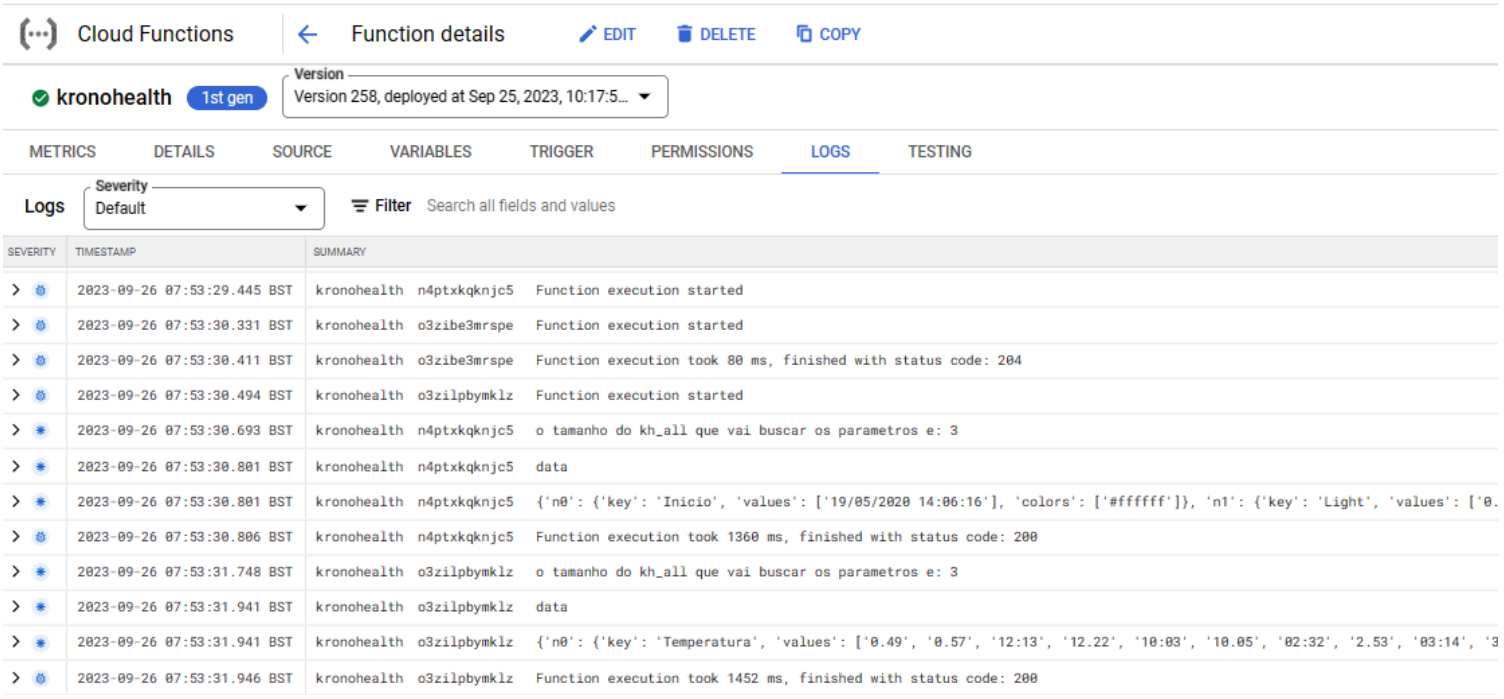

Figure 11 - Dashboard of a Google Cloud Function, LOGS tab

3.4 User Interface Implementation

Given that the Kronohealth cloud function could already supply the user interface with the patient data, all the front end components have access to the necessary resources by simply making an HTTP request to the function. This was done using the “Redux-Saga” library, a resource very useful for managing side effects in Redux applications that provides a structured and simple way to handle asynchronous operations.

The graph implementations mentioned in this document are part of the Kronohealth subsection in the individual analyses section of the CLARIFY application. The first task was to add another subsection to the Kronohealth dashboard. The subsection for the comparison graphs of the time series circadian variables was called “Mean Circadian Rhythms".

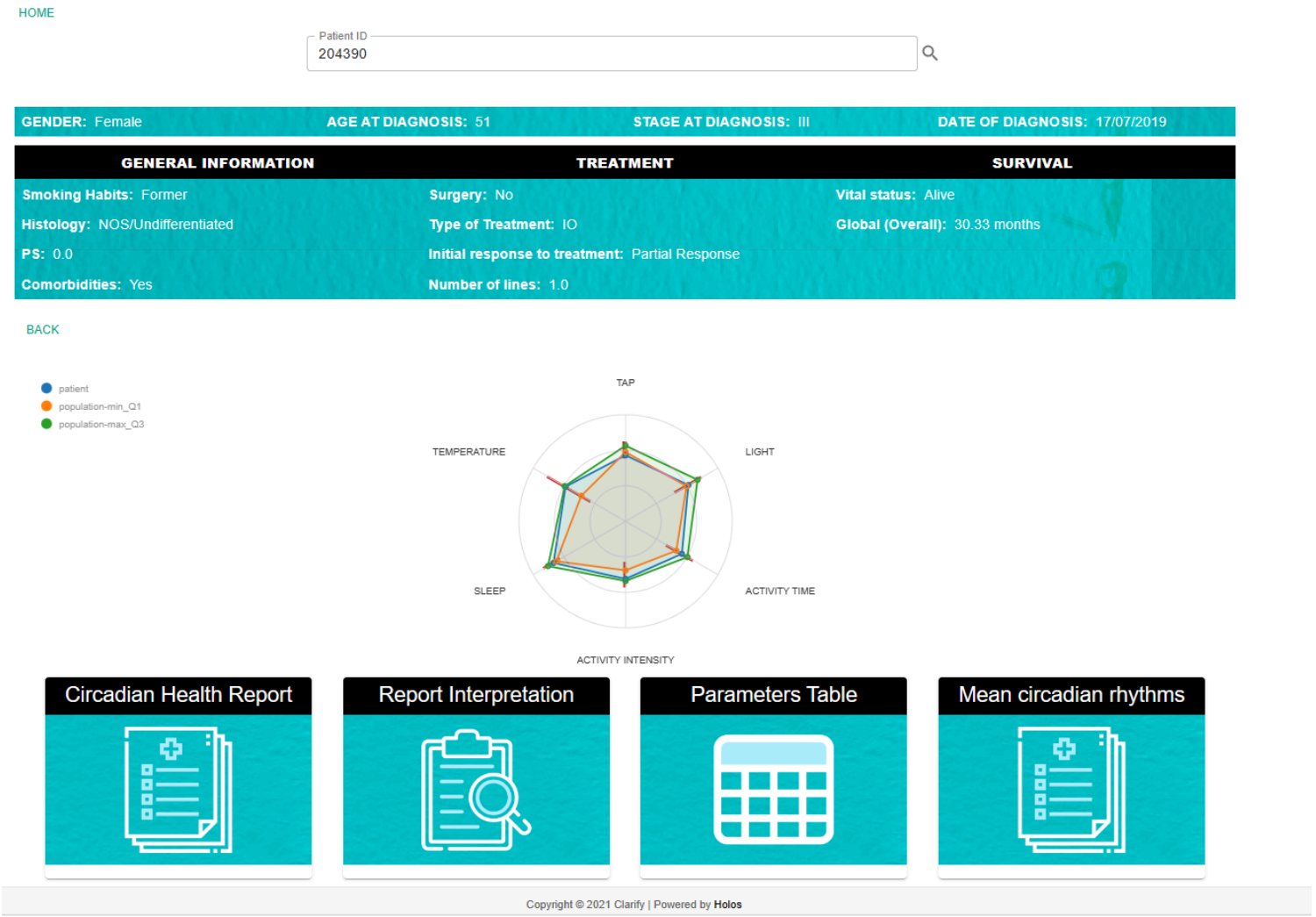

Figure 12 - User Interface of CLARIFY Kronohealth section

In figure 12, the dashboard for the Kronohealth section can be seen. The top header has general information about the patient, such as the gender, age, cancer type, the treatments that the patient underwent, etc.

It is worth noting that every subsection in the Kronohealth area has a displayed radar graph with average values for the circadian variables compared with the population first quartile value and third quartile value. These data values are obtained from the CSV and JSON files each patient has stored in the storage bucket. This radar graph was already implemented, while the more nuanced time series graphs are a novelty implementation. The radar can also be seen in figure 12. The blue hexagon line represents the patient's values, the orange line the first quartile value and the green line the third quartile value.

The user interface for these new graphs had to be intuitive and user friendly. The focus was on simplicity and ease of consultation as the end user, a clinician/doctor, likely has to compare the multiple circadian variables at the same time and on the same timestamp. In order to achieve that, the choice of design was to have a simple checkbox for every circadian variable that could toggle the display of the respective graph.

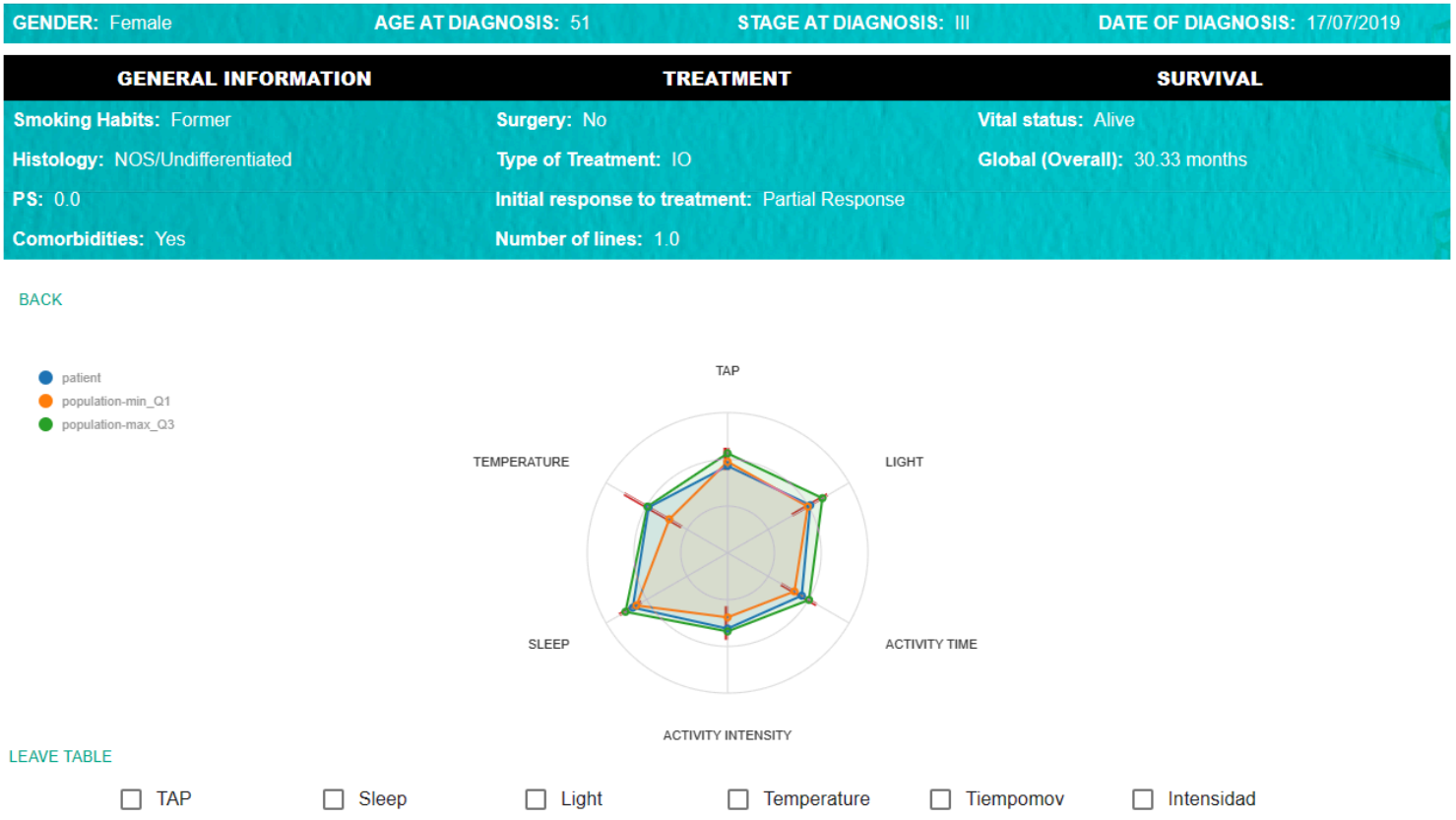

Figure 13 - User Interface of Mean Circadian Rhythms with no selected graph

Figure 13 shows the user interface with every checkbox unchecked which represents the initial state of this section when the user enters the "Mean Circadian Rhythms" option.

Once a checkbox is selected the corresponding graph is displayed and the screen dragged to center the graph on the screen.

The graphs use the same timestamp scale on the x axis and are all vertically aligned to facilitate the reading of values in a given timestamp. Hovering the mouse over the graph will display the variable value for the patient and the control population as shown in figure 14.

In order to compare the patient´s values with the control population average, a graph capable of displaying two types of lines in a single cartesian grid was needed. To achieve this, the library "recharts" was used. The area line always refers to the patient values while the simple orange line refers to the control population.

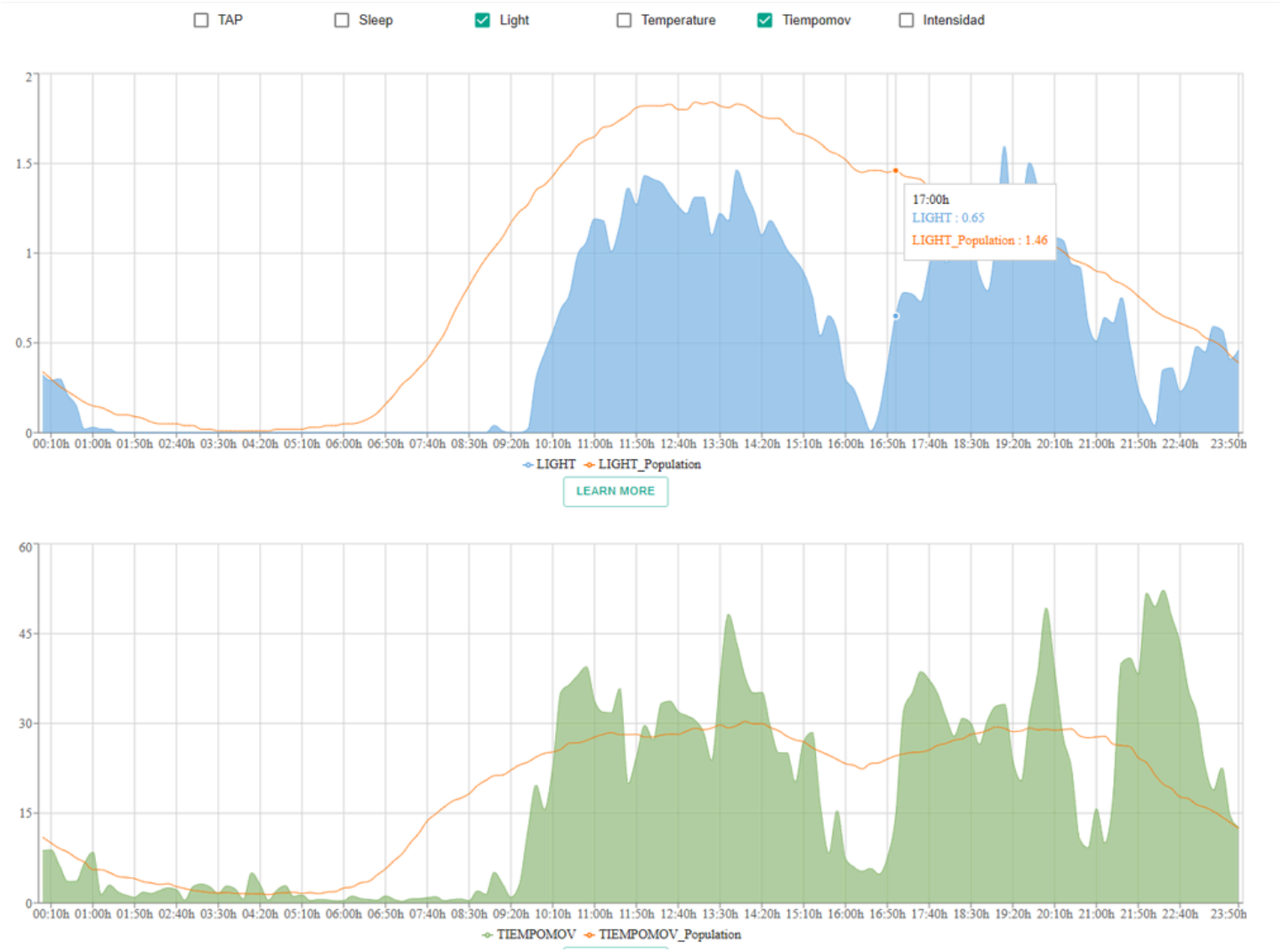

Figure 14 - User Interface of CLARIFY Kronohealth section with 2 graphs displayed

Figure 14, shows how the interface behaves when 2 graphs are displayed simultaneously. The behavior is similar for any other group of variables.

As an example, for the displayed patient, we can see that her values for the circadian variable "Light" are considerably below the average values of the control population. While the x axis only accounts for a day, the reality is that the patient wore the wearable device for an entire week and the values displayed are the average for every single measured value. This means that her low values for light are more representative of his overall routine rather than a single day, which wouldn't be as statistically significant. For the "TIEMPOMOV" or time of movement variable, the patient seems to have intervals where she outperforms the control population, and others where she falls short of it. This type of information is then used and analyzed by the clinicians.

Each graph also has a "Learn More" button that expands and displays a small text explaining the graph and what high and low values signify for the variable in question, as shown in figure 15. The texts were provided by Kronohealth and given the target user base, are written in spanish.

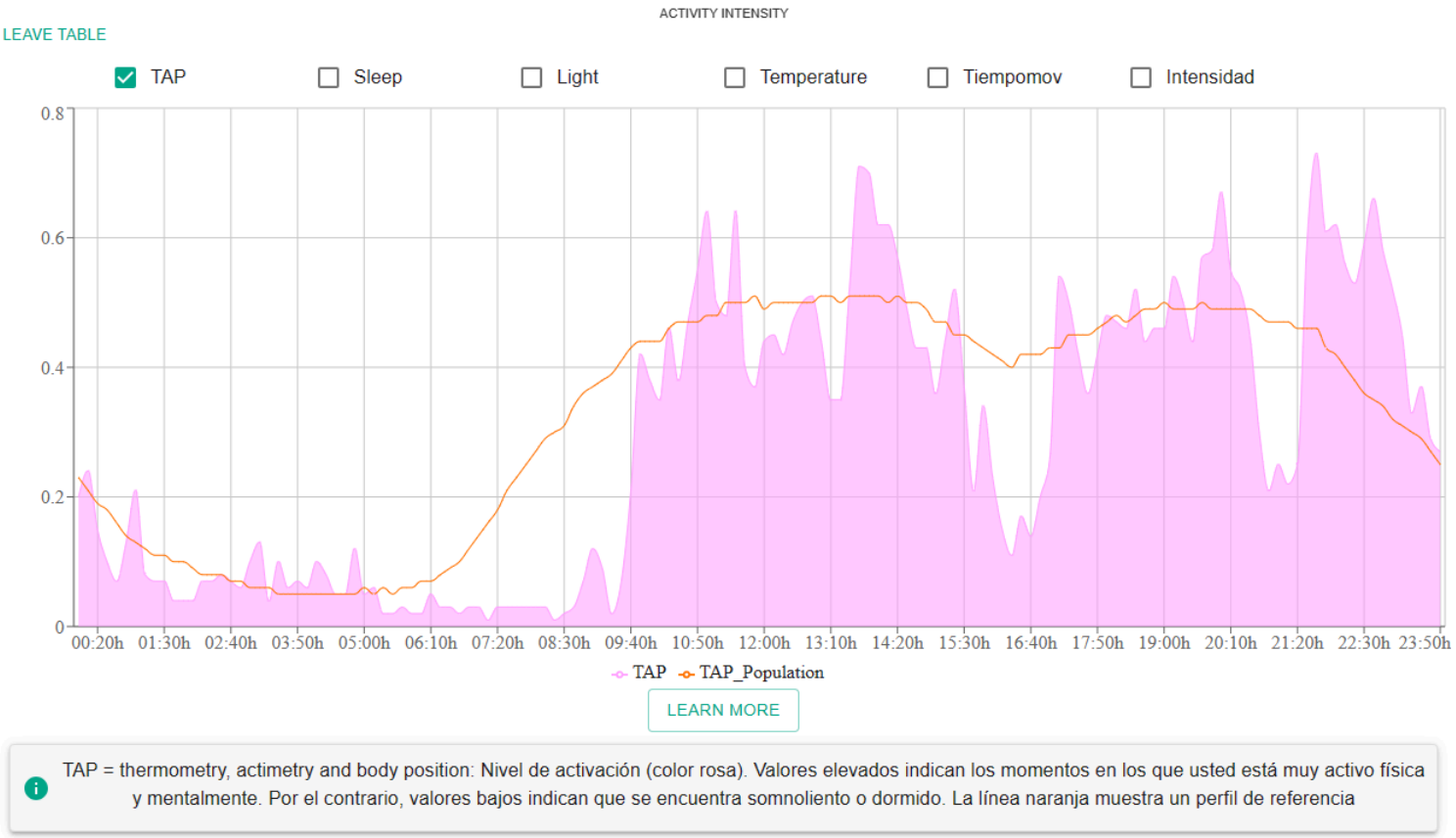

Figure 15 - Graph with the "Learn More" text displayed

The programming language used for the above mentioned implementations was naturally Javascript. To follow the same paradigm of the entire application the React library was also used coupled with the Redux state management library. Among the many node package modules used, it is worth mentioning “Material UI” (MUI) , an open-source library commonly used in react applications, as it provides a set of pre-designed, customizable and appealing UI components based on Google's Material Design guidelines. The MUI grid component was especially useful to align the multiple graphs and make them responsive and adaptable to different screen sizes.

The app also follows the React Atomic Design which is an approach to organizing and structuring user interface components in a hierarchical and modular manner. It is inspired by Brad Frost's Atomic Design methodology [62], which is a design system that breaks down UI elements into smaller, reusable components. In the context of React, the Atomic Design approach helps developers create scalable and maintainable UIs by categorizing components into distinct levels. This is particularly relevant as it allows clinicians to have access to the application on a computer, a Tablet or even a mobile phone as they move around the hospital. The levels identified by Atomic Design are:

- Atoms - the smallest UI components such as buttons and icons.
- Molecules - composed of one or more atoms, they represent simple UI patterns or combinations of atoms that work together to perform a specific function. An example would be an input field (atom) with a search button (atom) can be combined to create a search bar molecule.
- Organisms - larger functional units that combine molecules together and atoms.
- Template/Pages - highest-level components that represent individual views or routes within an application

In the implementations referred in this document we could identify as an example, the checkbox for each graph as an atom, each graph with their respective button and drop down text as a molecule, the combination of every checkbox with their respective graphs and texts put together in a react element called “KhGraphsContainer” as an organism and the entire kronohealth “Mean Circadian Rhythms” as a page.

3.5 Interventions Classification

Knowing that ultimately, the objective of describing the circadian rhythm of a patient is to suggest some interventions to their lifestyle in order to improve their quality of life, a study was conducted to verify if certain predictions could be made:

- Can the intervention that a doctor will recommend be predicted using Artificial intelligence techniques, if provided with a detailed description of the patient's circadian rhythm profile?
- Can the intervention that a doctor will recommend be predicted even without the circadian variables, and just using data related to the patient's cancer/treatment/history?

The motivation for this study was to build a decision support tool that could alert doctors/clinicians and notify them of potential areas or interventions that could be worthy of further investigation. Perhaps the second question, if answered affirmatively, could lead to a tool that ranks CLARIFY patients to choose which of them are more urgent to undergo the full week with the Kronohealth wearable device. In this case, a patient that the model would recommend one or more intervention could be accommodated with a wearable device with a higher priority than a patient that the model classifies as not needing interventions.

Such a thing, would not mean that the wearable devices and the circadian rhythm profiling of a patient are not necessary. Much to the contrary; an eventual AI classification model would work together with circadian variables to aid the medical professional in their decision making. It should be obvious that this study does not aim to replace medical professionals or their decisions, but simply to help them detect patterns in data more easily.

While the CLARIFY patient population is very vast, given that it integrates sources such as patients from the Puerta Hierro University Hospital and patients from the Spanish Lung Cancer Group, as of the writing of this document, only 355 patients in Clarify have used the Kronohealth wearable device. This is considered a small sample for machine learning and neural networks algorithms.

Furthermore, the Kronohealth patients are not classified, meaning there is no data that indicates which interventions each patient should be recommended for. This would be the job of the clinician/doctor. Indeed, the above mentioned implementations aim to help with that task.

In order to classify the patients, Dr. Maria Torrente provided guidelines that would indicate whether or not a patient would require an intervention in a given area based on their average values for the circadian variables:

- Light exposure - A patient would need an intervention to increase exposure if their values for the variable LIGHT were below a certain threshold.
- Physical activity - A patient would need an intervention to increase the amount of time engaged in physically demanding activities if the values for the variable TIEMPOMOV were below a certain threshold.
- Sleeping disorder - A patient would need an intervention to increase the amount and quality of sleep per day if the values for the variable SLEEP were below a certain threshold.
- Activity Intensity - A patient would need an intervention to increase the intensity of his activities if the values for the variable INTENSIDAD were below a certain threshold.

This would mean that a classification of 1 in any of the above categories would simply mean that an intervention in that respective area would likely be required and 0 would signify intervention likely not required. The exact intervention is part of medical decision making, and outside of the capabilities of the models studied. As an example given by Dra Maria Torrente, an intervention could simply be a suggestion to the patient to go outside more often, but could also mean prescription of medication such as antidepressants, analgesics, melatonin treatment, etc.

With the guidelines provided, a simple Python script appended four classification variables to a dataset containing every Kronohealth patient's data. Figure 16 shows a sample of 10 classified patients.

The remaining circadian variables provided by the Kronohealth wearable device were not used to make the classifications and were not target to any intervention. This is simply because no quick guide-line could be provided to satisfactorily indicate the need for an intervention, and in that case the in-depth analysis by a clinician using the time series graphics would be more appropriate.

	sleep_class	light_class	intensidad_class	tiempomov_class
P01	1	1	0	0
P02	0	0	0	0
P03	0	1	1	1
P04	0	1	0	0
P05	0	1	0	1
P06	0	0	0	1
P07	0	1	1	1
P08	0	1	0	0
P09	0	0	0	0
P10	0	0	1	1

Figure 16 - Sample of 10 patients and their classifications

3.5.1 Model using Circadian Variables

As proposed initially, the first case study involved predicting if a patient needs an intervention in any area, based on their circadian profile.

A series of Machine Learning algorithms were used, but the result of this task could be easily anticipated, as we are using circadian variables to make the prediction about a parameter (the class) that was calculated based on the patient's circadian profile itself.
Patients that have already passed, have been removed.

The dataset contains the value used in the hexagon radar graph of each of the 6 circadian variables already mentioned and is a mean representation of the variable.

The first algorithm implemented was a Random Forest Classifier, a popular machine learning algorithm used for both classification and regression tasks. It's based on the idea of combining multiple decision trees to make more accurate predictions. Each decision tree is trained independently on a random subset of the training data and a random subset of the features. This randomness introduces diversity among the trees. Then, when making predictions, each tree in the forest "votes" (in classification cases) or provides an output (in regression). For classification, the class that receives the most votes becomes the predicted class.

This algorithm was implemented individually to predict the 4 classes independently.

As machine learning good practices dictate, the data was separated into the training set and the test set to avoid overfitting, meaning that the model can´t be tested by predicting on the same data it learned from.

Ten-fold cross validation was also used to achieve more realistic results. This simply means that the entire process of training and testing the data is repeated 10 times, with different separations of the test and training data. This is done to avoid variance in the results, as the model could perform perhaps better in some cases of data splits than others. In this case the end result is the average of every iteration. In this case the entire data set contains 300 entries and the split was 80/20 meaning that for each iteration the training set has 240 instances and the test set 60 instances.

The implementation was done using the Python programming language, in Jupyter Notebook, using the packages "Pandas" and "Sklearn".

The table 2 shows the results for the class related to the sleep classification. Unsurprisingly, we can see a perfect score. Again, this is because the circadian profile of the patients was initially used for the classification of the data.

Sleep	**precision**	**recall**	**f1-score**
Class 0	1.00	1.00	1.00
Class 1	1.00	1.00	1.00
Macro avg	1.00	1.00	1.00
Weighted avg	1.00	1.00	1.00
Accuracy		1.00	

Table 2 - Results for Sleep Classification using Random Forest

The results for the classes related to the light, intensity and time of movement interventions can be seen in the tables 3, 4 ,5 respectively. As expected these classes also have perfect results.

Light	precision	recall	f1-score
Class 0	1.00	1.00	1.00
Class 1	1.00	1.00	1.00
Macro avg	1.00	1.00	1.00
Weighted avg	1.00	1.00	1.00
Accuracy		1.00	

Table 3 - Results for Light Classification using Random Forest

Intensidad	precision	recall	f1-score
Class 0	1.00	1.00	1.00
Class 1	1.00	1.00	1.00
Macro avg	1.00	1.00	1.00
Weighted avg	1.00	1.00	1.00
Accuracy		1.00	

Table 4 - Results for Intensidad Classification using Random Forest

Tiempomov	precision	recall	f1-score
Class 0	1.00	1.00	1.00
Class 1	1.00	1.00	1.00
Macro avg	1.00	1.00	1.00
Weighted avg	1.00	1.00	1.00
Accuracy		1.00	

Table 5 - Results for Tiempomov Classification using Random Forest

The accuracy is the model metric that evaluates the overall performance of the model across all classes. It is simply the percentage of correctly classified instances out of all instances in the dataset.

The metrics precision, recall, and F1-score are calculated for each individual class and are meant to evaluate the performance of the model for each class separately.

- Precision - It measures the percentage of true positive predictions out of all positive predictions for a specific class. It indicates how many of the predicted positive instances are actually correct.
- Recall - It measures the percentage of true positive predictions out of all actual positive instances for a specific class. It indicates how many of the actual positive instances were correctly predicted.
- F1-Score - It is the harmonic mean of precision and recall. It provides a balance between precision and recall. It's a useful metric when there is an imbalance between classes.

In classification reports the metrics “macro average” and “weighted average” are also commonly used. They provide an overall summary of the model's performance across all classes:

- Macro Average -This value is calculated by taking the average of the precision, recall, and F1-score across all classes without considering class imbalance. In other words, it treats all classes equally and calculates the average as if each class contributes equally to the overall performance.
- Weighted Average - This value is calculated by taking a weighted average of precision, recall, and F1-score across all classes, with the weight being the support (the number of true instances) for each class. It considers class imbalance, so classes with more instances have a greater impact on the weighted average.

It would be interesting to test this approach again, once real classifications done by clinicians with the time series graphs in order to see if the mean values for the circadian variables could have any predicting power for the interventions.

The next model studied had a more interesting premise: without using any information provided by the wearable devices, can the interventions be predicted. This model will work with variables that are not related to the circadian profile of the patient (and not used in the classification of the data).

3.5.2 Model without Circadian Variables

In order to attempt this model, the dataset had to contain only Kronohealth patients (as these are the only ones previously classified) but variables related to their oncological condition. The issue is, that among the 355 Kronohealth patients, three different types of cancer are present. As the oncological data from these patients is obtained from various sources such as the Puerta Hierro University Hospital and the Spanish Lung Cancer Group, the variables present in each type of cancer are mostly cancer specific, meaning that there are not many features present in all three of these cancer types databases.

The table 6 shows the distribution of Kronohealth patients based on their type of cancer.

Lung Cancer	**Lymphoma Cancer**	**Breast Cancer**
140 patients	34 patients	181 patients

Table 6 - Distribution of cancer on the Kronohealth patients

A choice had to be made: either use only the Lung Cancer patients (or any other cancer alone), which would be a smaller sample size, but with more features or use all of the available patients at a cost of reduced features, given that less variables are available to the three types of cancer simultaneously. As the choice to which would yield better results was not obvious, both scenarios were tested.

First the dataset was obtained from the database tables with the information on Lung Cancer Patients, as while being less than their Breast counterparts, they have less missing values. Again, patients that had already passed away and patients with missing values for the selected features were removed. This is a worrying scenario as the already small set of available data becomes even smaller, but most of the variables don't behave well if we try to fill the missing values with the population average.

The selected variables/features to use in the model were:

- Gender
- Age when diagnosed with cancer
- Stage of the cancer when diagnosed
- Smoking habits (current smoker, former smoker, or never smoked)
- Presence of comorbidities
- Family history of cancer
- Performance status, which a metric to determine a person's ability to carry out daily activities
- Survival months, since the time of diagnosis to the time the data was taken

After the customary data cleaning processes of deleting bad entries and encoding the features into usable values, the data set is left with 83 instances.

The first algorithm tried was again the Random Forest Classifier, tables 7, 8, 9 and 10 show the results for the classes Sleep, Light, Intensity (Intensidad) and Time of movement (Tiempomov) respectively. Like before, the value 1 means that the patient likely needs an intervention in that area, while 0 means the opposite.

Sleep	**precision**	**recall**	**f1-score**
Class 0	0.91	0.96	0.94
Class 1	0.25	0.12	0.17
Macro avg	0.58	0.54	0.55
Weighted avg	0.85	0.88	0.86
Accuracy		0.88	

Table 7 - Results for Sleep Classification using Random Forest

Light	**precision**	**recall**	**f1-score**
Class 0	0.21	0.12	0.16
Class 1	0.70	0.81	0.75
Macro avg	0.45	0.47	0.45
Weighted avg	0.56	0.61	0.58
Accuracy		0.61	

Table 8 - Results for Light Classification using Random Forest

Intensidad	precision	recall	f1-score
Class 0	0.43	0.31	0.36
Class 1	0.63	0.75	0.68
Macro avg	0.53	0.53	0.52
Weighted avg	0.56	0.58	0.56
Accuracy		0.58	

Table 9 - Results for Intensidad Classification using Random Forest

Tiempomov	precision	recall	f1-score
Class 0	0.46	0.51	0.48
Class 1	0.46	0.40	0.43
Macro avg	0.46	0.46	0.46
Weighted avg	0.46	0.46	0.46
Accuracy		0.46	

Table 10 - Results for Tiempomov Classification using Random Forest

The time of movement (Tiempomov) with 46% accuracy, is worse than flipping a coin which indicates that this class cannot be predicted with the selected features using the Random Forest classification method. The intensity (Intensidad) and Light classes have slightly better accuracy, but still very poor for the intended purposes. The sleep variable seems to have a more interesting accuracy, however with further analysis it can be seen that while the model is very good at predicting the class as 0, it is very poor at classifying the class as 1, with only 25% of instances with sleep class as 1 being classified as such.

The reason for this is likely imbalances in classes. In fact, out of the 83 available patients, 75 were originally classified with 0. A model that would always predict 0 for sleep class (0R) would have a similar result as this Random Forest.

An interesting observation is the classes with more balanced distributions perform worse. This indicates that the model indeed does not perform well, and high accuracy scores are only

caused by class imbalance and not by a desired statistical variable correlation. This model was deemed a failure and more models were tried.

The next model implemented was the Naive-Bayes algorithm for classification, a very popular option for classification problems that applies Bayes Network with independence assumptions between the features. The tables 11, 12, 13 and 14 show the results for the classes Sleep, Light, Intensity (Intensidad) and Time of movement (Tiempomov) respectively.

Sleep	**precision**	**recall**	**f1-score**
Class 0	0.89	0.88	0.89
Class 1	0.00	0.00	0.00
Macro avg	0.45	0.44	0.44
Weighted avg	0.81	0.80	0.81
Accuracy		0.80	

Table 11 - Results for Sleep Classification using Naive-Bayes

Light	**precision**	**recall**	**f1-score**
Class 0	0.30	0.28	0.29
Class 1	0.71	0.73	0.72
Macro avg	0.51	0.51	0.51
Weighted avg	0.59	0.60	0.59
Accuracy		0.60	

Table 12 - Results for Light Classification using Naive-Bayes

Intensidad	precision	recall	f1-score
Class 0	0.30	0.28	0.29
Class 1	0.58	0.60	0.59
Macro avg	0.44	0.44	0.44
Weighted avg	0.48	0.48	0.48
Accuracy		0.48	

Table 13 - Results for Intensidad Classification using Naive-Bayes

Tiempomov	precision	recall	f1-score
Class 0	0.52	0.39	0.44
Class 1	0.54	0.66	0.59
Macro avg	0.53	0.52	0.52
Weighted avg	0.53	0.53	0.52
Accuracy		0.53	

Table 14 - Results for Tiempomov Classification using Naive-Bayes

The results show a very similar behavior to the Random Forest classifier, but with even lower accuracy. Again, most of the target classes can´t be predicted at all, having accuracies of about 50%. The sleep class, while showing higher overall accuracy, still performs very poorly when predicting the value 1. This example is even more extreme, as the Naive-Bayes classifier in question failed to correctly predict a class 1 in the sleep target class even once.

The same reasons for failure apply; possible lack of correlation between the features and the target classes, and for the Sleep, a class imbalance coupled with small sample size make the model severely deficient in predictions.

Various other machine learning classification methods were implemented such as Regression Models, Support Vector Machines, K-nearest neighbors, etc. As these models vary significantly on how they work, there was hope that perhaps one could be efficient in these classifications, but the results for all of the mentioned models were the same: Low accuracy for

most of the target classes, with sleep having a higher accuracy but still failing to classify instances as 1.

As the realization that the most likely scenario of no existing correlation being the most likely outcome to this endeavor, there was still a few different attempts; as mentioned in the previous section there are instances in literature of researches having very good success in classification problems in medicine/healthcare areas using neural network approaches, more concretely the Multilayer Perceptron (MLP)[43]. The "sklearn" Python package made it easy to try different network architectures and tests were made with varying numbers of hidden layers, neurons per hidden layer and different activation functions. In the end every attempt followed the trend of the previously implemented models; unsatisfactory accuracy rates for most of the target classes, and inability to predict the value 1 for the sleep class.

It is possible to visually compare each of the predictions of the model with the actual class value for each instance. The figure 17 shows this comparison for an MLP with 2 hidden layers with 100 and 50 nodes respectively and with the Rectified Linear Unit activation function. It verifies that indeed the model only predicts 0 for sleep class. We can also see that for the intensity (intensidad) class and the light class it also only predicts one of the values.

```
==========================================================================================
Class: sleep_class                                  Class: intensidad_class
Sample 0: Predicted=0, Actual=0.0                   Sample 0: Predicted=1, Actual=1.0
Sample 1: Predicted=0, Actual=0.0                   Sample 1: Predicted=1, Actual=0.0
Sample 2: Predicted=0, Actual=1.0                   Sample 2: Predicted=1, Actual=1.0
Sample 3: Predicted=0, Actual=0.0                   Sample 3: Predicted=1, Actual=1.0
Sample 4: Predicted=0, Actual=1.0                   Sample 4: Predicted=1, Actual=1.0
Sample 5: Predicted=0, Actual=0.0                   Sample 5: Predicted=1, Actual=1.0
Sample 6: Predicted=0, Actual=0.0                   Sample 6: Predicted=1, Actual=1.0
Sample 7: Predicted=0, Actual=0.0                   Sample 7: Predicted=1, Actual=1.0
==========================================================================================
Class: light_class                                  Class: tiempomov_class
Sample 0: Predicted=1, Actual=1.0                   Sample 0: Predicted=0, Actual=1.0
Sample 1: Predicted=1, Actual=1.0                   Sample 1: Predicted=1, Actual=1.0
Sample 2: Predicted=1, Actual=1.0                   Sample 2: Predicted=1, Actual=1.0
Sample 3: Predicted=1, Actual=0.0                   Sample 3: Predicted=1, Actual=1.0
Sample 4: Predicted=1, Actual=1.0                   Sample 4: Predicted=0, Actual=0.0
Sample 5: Predicted=1, Actual=1.0                   Sample 5: Predicted=0, Actual=1.0
Sample 6: Predicted=1, Actual=0.0                   Sample 6: Predicted=0, Actual=1.0
Sample 7: Predicted=1, Actual=0.0                   Sample 7: Predicted=1, Actual=0.0
```

Figure 17 - Sample of predictions and correct answers for the 4 target classes

As the neural network approach failed as well, the last attempt was to reduce dimensionality of the data set. This can sometimes lead to better prediction outcomes as it may alleviate the model of useless features that cause noise and negatively affect the outcome. There is also the possibility of what is usually called "curse of dimensionality" in which a dataset with

many variables may experience data points becoming sparse making it harder to find meaningful patterns or separate classes effectively.

To select a fewer number of variables to the dataset, feature selection algorithms are commonly used. A common choice for classification problems is the Univariate feature selection using the chi-squared statistical test. This method, in order to select the best features, quantifies the degree of association or independence between the feature and the target class. It measures how much the observed frequency of each category in the feature differs from the expected frequency if the feature and target were independent.

The selected features are determined independently for each target variable based on their individual associations with the features. Given that there are 4 different target classes , we got 4 different sets of selected features for each target class. The choice was made to use the number of features as only 3. The table 15 shows what features were deemed more important for each target class:

Target Class	**Most Important Features**
Sleep	Gender, Performance status, Survival months
Light	Age of diagnosis, Performance status, Survival months
Intensidad	Age of diagnosis, Performance status, Survival months
Tiempomov	Age of diagnosis, Smoking habits, Survival months

Table 15 - Preferred 3 Features for each class

For a final attempt, another data set was considered, with the same amount of instances (83) and with only 3 features (the most common across all the classes in the table X):

- Age of diagnosis
- Performance status
- Survival months

With this new dataset the previously implemented algorithms were tested. The first was the Random Forest as it was the one with higher accuracies and the one with higher amount of correctly predicted 1 values for the sleep class.

The tables 16, 17, 18 and 19 display the results for the classes Sleep, Light, Intensity (Intensidad) and Time of movement (Tiempomov) respectively.

Sleep	precision	recall	f1-score
Class 0	0.90	0.94	0.92
Class 1	0.00	0.00	0.00
Macro avg	0.45	0.47	0.46
Weighted avg	0.82	0.85	0.83
Accuracy		0.85	

Table 16 - Results for Sleep Classification using Random Forest

Light	precision	recall	f1-score
Class 0	0.33	0.16	0.22
Class 1	0.71	0.87	0.78
Macro avg	0.52	0.51	0.50
Weighted avg	0.60	0.66	0.62
Accuracy		0.66	

Table 17 - Results for Light Classification using Random Forest

Intensidad	precision	recall	f1-score
Class 0	0.30	0.25	0.27
Class 1	0.59	0.64	0.61
Macro avg	0.44	0.45	0.44
Weighted avg	0.48	0.49	0.48
Accuracy		0.49	

Table 18 - Results for Intensidad Classification using Random Forest

Tiempomov	precision	recall	f1-score
Class 0	0.40	0.34	0.37
Class 1	0.46	0.52	0.49
Macro avg	0.43	0.43	0.43
Weighted avg	0.43	0.44	0.43
Accuracy		0.44	

Table 19 - Results for Tiempomov Classification using Random Forest

As can be seen in the tables the results kept the trend of very poor performance. The only class with a decent accuracy value was again the sleep class, but the same issue regarding inability to classify instances as 1 persisted.

The other models were also all tested on this less dimensional data set. While the accuracies and precision for each class slightly deviated from one model to another, the overarching conclusion was always the same.

Tests were also conducted using the selected features for each target class shown in table 15 to individually attempt to classify that class alone. Again, no improvement was achieved.

As discussed in the beginning of this classification study without the use of the circadian rhythm related variables, a choice between a smaller dataset with only Lung cancer patients with more features, or a larger dataset containing patients from the 3 types of cancer with a fewer

number of variables was made. The above implementations were all done with the more nuanced smaller Lung cancer patients only, however tests were also made with the larger dataset.

To build this dataset that comprised patients from different tables and sources, the features had to be present and common in all three of the types of cancer. The only features available were:

- Gender
- Age when diagnosed with cancer
- Cancer type
- Stage of the cancer when diagnosed
- Smoking habits (current smoker, former smoker, or never smoked)
- Survival months

There was also the feature of "Family history of cancer", but the amount of missing values was overwhelming, and a decision to drop this variable was made. The figure 18 shows a sample of 10 patients with their corresponding features and target variable values.

KH	gender	age_of_diagnosis	cancer	stage	smoker	survival_months	sleep_class	light_class	intensidad_class	tiempomov_class
P04	Female	51	Lung	III	Former	30.33	0	1	0	0
P04	Male	51	Lung	III	Former	30.33	0	1	0	0
P09	Male	65	Lung	III	Former	23.57	0	0	0	0
P11	Male	68	Lung	IV	Former	69.03	0	0	0	0
P13	Female	62	Lung	IV	Former	29.2	0	0	0	0
P15	Female	65	Lung	IV	No	62.77	0	1	0	0
P16	Female	50	Lung	IV	Current	48.27	0	1	1	1
P19	Male	57	Lung	IV	Current	20.07	0	1	1	1
P21	Male	58	Lung	IV	Former	30.6	0	0	0	0

Figure 18 - Sample of 10 patients with features and target classes

The standard data cleaning processes of deleting bad entries and encoding the features into usable values was done, which left the dataset with 105 entries. This represented a huge loss in instances for the dataset, from the patients with missing values and the diseased patients.

Again, the same machine learning and neural networks algorithms were implemented. The Random Forest classifier was again the first choice. The split of training set and test set was made again 80/20 which made the training set have 84 instances and the test set 21 instances. As per usual, 10-fold cross validation was used. The results can be seen in the tables 20, 21, 22 and 23 for the classes Sleep, Light, Intensity (Intensidad) and Time of movement (Tiempomov) respectively.

Sleep	**precision**	**recall**	**f1-score**
Class 0	0.90	0.95	0.92
Class 1	0.00	0.00	0.00
Macro avg	0.45	0.47	0.46
Weighted avg	0.81	0.86	0.84
Accuracy		0.86	

Table 20 - Results for Sleep Classification using Random Forest

Light	precision	recall	f1-score
Class 0	0.32	0.21	0.25
Class 1	0.69	0.79	0.74
Macro avg	0.50	0.50	0.50
Weighted avg	0.57	0.61	0.58
Accuracy		0.61	

Table 21 - Results for Light Classification using Random Forest

Intensidad	precision	recall	f1-score
Class 0	0.50	0.42	0.46
Class 1	0.70	0.76	0.73
Macro avg	0.60	0.59	0.59
Weighted avg	0.63	0.64	0.63
Accuracy		0.64	

Table 22 - Results for IntensidadClassification using Random Forest

Tiempomov	precision	recall	f1-score
Class 0	0.37	0.33	0.35
Class 1	0.57	0.61	0.59
Macro avg	0.47	0.47	0.47
Weighted avg	0.49	0.50	0.49
Accuracy		0.50	

Table 23 - Results for Tiempomov Classification using Random Forest

The results mirror what was obtained with the previous dataset. Once again, the accuracy values are very unsatisfactory for most of the target classes, and when the accuracy is decent, such as the sleep class, the model fails to predict one of the possible classes. The other algorithms already mentioned in the previous dataset were also tested but also performed similarly.

The possible conclusion to this study of predicting whether a patient needs intervention in one of the 4 areas based solely on their general and oncological information, that would be available before the profiling of the patient´s circadian profile was deemed impossible given the features and instances available.

The small sample size and class imbalances are also parameters that negatively affect the results and the certainty of which conclusions can be drawn, given that smaller sample sizes usually lack statistical significance.

4

CONCLUSIONS

The circadian rhythm variables provided by the Kronohealth wearable device give an overview of the daily physiological and behavioral patterns of its wearer. This application has appreciable value for post oncology patients as most of the oncology treatments leave lasting marks on the patients health and well being. The time series description of the circadian variables provides a much more in depth analysis than basic average values. The implementation of a tool that could display that more nuanced information about a patient's circadian rhythm and compare it to a standard value of a control population was requested by Dra. Maria Torrente as she believed it would be a valuable tool to add to the CLARIFY web application. The development of such a tool was followed closely by Dra. Maria Torrente and Professor Manuel Campos, the CTO of Kronohealth, and the end product was approved and deployed to the app which is now being used in a production environment by the clinicians at Hospital Puerta de Hierro Majadahonda, in Madrid.

CLARIFY has since gone through a scientific review from the European Commission, with a very positive outcome. Within that scientific review the work that was the focus of this thesis within the section “Mean Circadian Rhythms” of the Clarify was presented.

The study done at the end of this work, aimed to test the viability of another hypothetical feature to the CLARIFY app that would make Artificial Intelligence recommendations to doctors/clinicians/users of the app, about areas where a patient would likely benefit from an intervention. The first approach, using the circadian information collected by the Kronohealth device was redundant, as the circadian rhythm variables themselves were used for the patient classification. However, it would be interesting to rerun the model once the classification method became more nuanced and done by a doctor using the full spectrum of the information provided by the wearable device, such as the in depth time series graphs. This would allow determining if the average values for the circadian variables could be related and used to predict the intervention prescribed by the clinician.

The second phase of this study was a lot more interesting, as it tested if some non-obvious correlation could exist between the patient's overall information and oncology status and the potential intervention area a clinician could prescribe. The answer was unfortunately no. No real prediction power could be observed by the variables used. This fact however gives strength to the idea that the description of the circadian rhythm profile of a patient is of high importance, as the conclusion drawn from such information cannot be drawn from more mundane data. This can be used as an argument for wider adoption of the wearable device by post oncology patients. Nevertheless some fault might be attributed to the small datasets available and the high number of missing values on some features that prevented their inclusion in the study.

4.1 Future work

As the entire flow from data import, backend cloud operations and user interface components are already fully developed and ready for any type of data, even if hypothetical newer versions of the wearable device that collect more types of data were to be used, the entire mechanism developed and described in this document would need very little changes or reviews. Only the data import script would have to be tweaked to accommodate new types of data as well as the fetching of data done in the respective cloud function.

As far as the classification study portion of the project, many aspects would be worthy of approaching again in the future. As more patients start to use the wearable devices, and their clinicians use the information to prescribe interventions, a wider sample set could be available in the future. For the classification regarding the use of the circadian variables, simply having the data be classified by a clinician in a real world setting would make this part of the study a lot more meaningful and worthy of revisiting.

The data sets used in this study could indeed use larger sample sizes. The issue of missing values in a lot of the available data entries made the already limited available data even smaller. Perhaps it could be interesting to study ways to attribute meaningful values to some of the missing values. This is called imputation of data, however this practice is riskier when dealing with smaller datasets and with few features. The variables that suffer the most from missing values were smoking habits and family history of previous cancer which are notoriously difficult to attribute without making unreasonable assumptions.

4.2 Finishing Thoughts

The entire project showcased the difficulty of working with data. This aspect was mitigated by the high quality of data provided by the Kronohealth team, which made the development of the graphs of the "Mean Circadian Rhythms" tool go smoothly. This was not the case however in the intervention classification case study portion of the thesis. The data sourced from the Hospitals of Spain contained errors and missing values. The number of variables that were present simultaneously on the data tables of the three types of cancer in question was very reduced which also made it difficult to build a classification model using a dataset containing all those patients, given that the number of available features was reduced.

Overall the objectives set for this project were concluded. The developed tool to compare the circadian profile of patients is already present in the CLARIFY web application and ready to be used. Dra. Maria Torrente gave very positive feedback on the user experience and value the tool provided.

As for personal remarks, this project was very motivating to work on. Although with a modest contribution, working on a tool used by clinicians in the fight against the negative effects of cancer was very gratifying.

The technical aspects that were explored in this project are also very modern and useful for a future career in the information technology area. Knowledge and experience in the areas of ETL, cloud and serverless computing, user interface development and machine learning are very useful to have. This project also provided an experience of what working with a team of other developers and colleagues in a corporate setting is like.

BIBLIOGRAFIA

[1] - Siegel RL, Miller KD, Fuchs HE, Jemal A. Cancer Statistics, 2021. CA Cancer J Clin. 2021 Jan;71(1):7-33. doi: 10.3322/caac.21654. Epub 2021 Jan 12. Erratum in: CA Cancer J Clin. 2021 Jul;71(4):359. PMID: 33433946.

[2] - Cancer Genome Atlas Research Network. Comprehensive genomic characterization of squamous cell lung cancers. Nature. 2012 Sep 27;489(7417):519-25. doi: 10.1038/nature11404. Epub 2012 Sep 9. Erratum in: Nature. 2012 Nov 8;491(7423):288. Rogers, Kristen [corrected to Rodgers, Kristen]. PMID: 22960745; PMCID: PMC3466113.

[3] - Aaronson, N. K., Ahmedzai, S., Bergman, B., Bullinger, M., Cull, A., Duez, N. J., ... & Flechtner, H. (1993). The European Organization for Research and Treatment of Cancer QLQ-LC13: a modular supplement to the EORTC Core Quality of Life Questionnaire (QLQ-C30) for use in lung cancer trials. The Journal of Thoracic and Cardiovascular Surgery, 106(6), 1071-1081.

[4] - Wai Hoong Chang, Michail Katsoulis, Yen Yi Tan, Stefanie H. Mueller, Katherine Green, Alvina G. Lai, Late effects of cancer in children, teenagers and young adults: Population-based study on the burden of 183 conditions, in-patient and critical care admissions and years of life lost, The Lancet Regional Health - Europe, 2022, 100248, ISSN 2666-7762, https://doi.org/10.1016/j.lanepe.2021.100248.

[5] -Kalafi EY, Nor NAM, Taib NA, Ganggayah MD, Town C, Dhillon SK. Machine Learning and Deep Learning Approaches in Breast Cancer Survival Prediction Using Clinical Data. Folia Biol (Praha). 2019;65(5-6):212-220. PMID: 32362304.

[6] - Jastania R, Nageeti T, Al-Juhani H, Basahel A, Aljuraid R, Alanazi A, Aldosari H, Aldosari B. Utilizing Big Data in Healthcare, How to Maximize Its Value. Stud Health Technol Inform. 2019 Jul 4;262:356-359. doi: 10.3233/SHTI190092. PMID: 31349341 .

[7] ' Q. Hanlin, J. Xianzhen and Z. Xianrong, "Research on Extract, Transform and Load(ETL) in Land and Resources Star Schema Data Warehouse," 2012 Fifth International Symposium

on Computational Intelligence and Design, Hangzhou, China, 2012, pp. 120-123, doi: 10.1109/ISCID.2012.38.

[8] - Brynjolfsson, E., & McAfee, A. (2011). Race against the machine: How the digital revolution is accelerating innovation, driving productivity, and irreversibly transforming employment and the economy. Digital Frontier Press, ISBN-10 : 0984725113

[9] - Manyika, James & Chui, Michael & Brown, Brad & Bughin, Jacques & Dobbs, Richard & Roxburgh, Charles & Byers, Angela. (2011). Big data: The next frontier for innovation, competition, and productivity.

[10] - Kimball, Ralph, and Margy Ross. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. John Wiley & Sons, 2013.

[11] - Camilleri, C., Vella, J. G., & Nezval, V. (2021). HTAP With Reactive Streaming ETL. Journal of Cases on Information Technology (JCIT), 23(4), 1-19. http://doi.org/10.4018/JCIT.20211001.oa10

[12] - S. K. Bansal and S. Kagemann, "Integrating Big Data: A Semantic Extract-Transform-Load Framework," in Computer, vol. 48, no. 3, pp. 42-50, Mar. 2015, doi: 10.1109/MC.2015.76.

[13] - Z. Jian-hua and Z. Nan, "Cloud Computing-based Data Storage and Disaster Recovery," 2011 International Conference on Future Computer Science and Education, Xi'an, China, 2011, pp. 629-632, doi: 10.1109/ICFCSE.2011.157.

[14] - Kartick Chandra Mondal, Neepa Biswas, and Swati Saha. 2020. Role of Machine Learning in ETL Automation. In Proceedings of the 21st International Conference on Distributed Computing and Networking (ICDCN '20). Association for Computing Machinery, New York, NY, USA, Article 57, 1–6. https://doi.org/10.1145/3369740.3372778

[15] - Lennerholt, C., van Laere, J., & Söderström, E. (2018). Implementation challenges of self service business intelligence: A literature review. In 51st Hawaii International Conference on System Sciences, Hilton Waikoloa Village, Hawaii, USA, January 3-6, 2018 (Vol. 51, pp. 5055-5063). IEEE Computer Society.

[16] - Prasser F, Spengler H, Bild R, Eicher J, Kuhn KA. Privacy-enhancing ETL-processes for biomedical data. Int J Med Inform. 2019 Jun;126:72-81. doi: 10.1016/j.ijmedinf.2019.03.006. Epub 2019 Mar 23. PMID: 31029266.

[17] - Lu L, Zhang J, Xie Y, Gao F, Xu S, Wu X, Ye Z. Wearable Health Devices in Health Care: Narrative Systematic Review. JMIR Mhealth Uhealth. 2020 Nov 9;8(11):e18907. doi: 10.2196/18907. PMID: 33164904; PMCID: PMC7683248.

[18] - Fuller D, Colwell E, Low J, Orychock K, Tobin MA, Simango B, Buote R, Van Heerden D, Luan H, Cullen K, Slade L, Taylor NGA. Reliability and Validity of Commercially Available Wearable Devices for Measuring Steps, Energy Expenditure, and Heart Rate: Systematic Review. JMIR Mhealth Uhealth. 2020 Sep 8;8(9):e18694. doi: 10.2196/18694. PMID: 32897239; PMCID: PMC7509623.

[19] - In Yae Cheong, So Yeon An, Won Chul Cha, Mi Yong Rha, Seung Tae Kim, Dong Kyung Chang, Ji Hye Hwang, Efficacy of Mobile Health Care Application and Wearable Device in Improvement of Physical Performance in Colorectal Cancer Patients Undergoing Chemotherapy, Clinical Colorectal Cancer, 2018, doi: https://doi.org/10.1016/j.clcc.2018.02.002.

[20] -Haghi M, Thurow K, Stoll R. Wearable Devices in Medical Internet of Things: Scientific Research and Commercially Available Devices. Healthc Inform Res. 2017 Jan;23(1):4-15. doi: 10.4258/hir.2017.23.1.4. Epub 2017 Jan 31. PMID: 28261526; PMCID: PMC5334130.

[21] - Lu L, Zhang J, Xie Y, Gao F, Xu S, Wu X, Ye Z. Wearable Health Devices in Health Care: Narrative Systematic Review. JMIR Mhealth Uhealth. 2020 Nov 9;8(11):e18907. doi: 10.2196/18907. PMID: 33164904; PMCID: PMC7683248.

[22] - Huang Y, Upadhyay U, Dhar E, Kuo LJ, Syed-Abdul S. A Scoping Review to Assess Adherence to and Clinical Outcomes of Wearable Devices in the Cancer Population. Cancers (Basel). 2022 Sep 13;14(18):4437. doi: 10.3390/cancers14184437. PMID: 36139602; PMCID: PMC9496886.

[23] - Choi RY, Coyner AS, Kalpathy-Cramer J, Chiang MF, Campbell JP. Introduction to Machine Learning, Neural Networks, and Deep Learning. Transl Vis Sci Technol. 2020 Feb 27;9(2):14. doi: 10.1167/tvst.9.2.14. PMID: 32704420; PMCID: PMC7347027.

[24] - Deo RC. Machine Learning in Medicine. Circulation. 2015 Nov 17;132(20):1920-30. doi: 10.1161/CIRCULATIONAHA.115.001593. PMID: 26572668; PMCID: PMC5831252.

[25] - Uddin S, Khan A, Hossain ME, Moni MA. Comparing different supervised machine learning algorithms for disease prediction. BMC Med Inform Decis Mak. 2019 Dec 21;19(1):281. doi: 10.1186/s12911-019-1004-8. PMID: 31864346; PMCID: PMC6925840.

[26] - Eshaghi A, Young AL, Wijeratne PA, Prados F, Arnold DL, Narayanan S, Guttmann CRG, Barkhof F, Alexander DC, Thompson AJ, Chard D, Ciccarelli O. Identifying multiple sclerosis subtypes using unsupervised machine learning and MRI data. Nat Commun. 2021 Apr 6;12(1):2078. doi: 10.1038/s41467-021-22265-2. Erratum in: Nat Commun. 2021 May 20;12(1):3169. PMID: 33824310; PMCID: PMC8024377.

[27] - Chao CJ, Kato N, Scott CG, Lopez-Jimenez F, Lin G, Kane GC, Pellikka PA. Unsupervised Machine Learning for Assessment of Left Ventricular Diastolic Function and Risk Stratification. J Am Soc Echocardiogr. 2022 Dec;35(12):1214-1225.e8. doi: 10.1016/j.echo.2022.06.013. Epub 2022 Jul 12. PMID: 35840082.

[28] - Cheong JH, Wang SC, Park S, Porembka MR, Christie AL, Kim H, Kim HS, Zhu H, Hyung WJ, Noh SH, Hu B, Hong C, Karalis JD, Kim IH, Lee SH, Hwang TH. Development and validation of a prognostic and predictive 32-gene signature for gastric cancer. Nat Commun. 2022 Feb 9;13(1):774. doi: 10.1038/s41467-022-28437-y. PMID: 35140202; PMCID: PMC8828873.

[29] - Sun Z, Dong W, Li H, Huang Z. Adversarial reinforcement learning for dynamic treatment regimes. J Biomed Inform. 2023 Jan;137:104244. doi: 10.1016/j.jbi.2022.104244. Epub 2022 Nov 17. PMID: 36402277.

[30] - Han Z, Wei B, Leung S, Nachum IB, Laidley D, Li S. Automated Pathogenesis-Based Diagnosis of Lumbar Neural Foraminal Stenosis via Deep Multiscale Multitask Learning. Neuroinformatics. 2018 Oct;16(3-4):325-337. doi: 10.1007/s12021-018-9365-1. PMID: 29450848.

[31] - Sadiku, M., Shadare, A. E., Musa, S. M., Akujuobi, C. M., & Perry, R. (2016). Data visualization. International Journal of Engineering Research And Advanced Technology (IJERAT), 2(12), 11-16.

[32] - Schleier-Smith, Johann, et al. "What serverless computing is and should become: The next phase of cloud computing." *Communications of the ACM* 64.5 (2021): 76-84.

[33] - P. S. Diouf, A. Boly and S. Ndiaye, "Variety of data in the ETL processes in the cloud: State of the art," *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, Bangkok, Thailand, 2018, pp. 1-5, doi: 10.1109/ICIRD.2018.8376308.

[34] - Poudel PG, Bauer HE, Srivastava DK, et al. Online Platform to Assess Complex Social Relationships and Patient-Reported Outcomes Among Adolescent and Young Adult Cancer Survivors. *JCO Clin Cancer Inform*. 2021;5:859-871. doi:10.1200/CCI.21.00044

[35] - Lujan MR, Perez-Pozuelo I, Grandner MA. Past, Present, and Future of Multisensory Wearable Technology to Monitor Sleep and Circadian Rhythms. *Front Digit Health*. 2021;3:721919. Published 2021 Aug 16. doi:10.3389/fdgth.2021.721919

[36] - DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue, and Svelte Levlin, Mattias (2020)

[37] - Torrente M, Sousa PA, Hernández R, Blanco M, Calvo V, Collazo A, Guerreiro GR, Núñez B, Pimentao J, Sánchez JC, Campos M, Costabello L, Novacek V, Menasalvas E, Vidal ME, Provencio M. An Artificial Intelligence-Based Tool for Data Analysis and Prognosis in

Cancer Patients: Results from the Clarify Study. Cancers. 2022; 14(16):4041. https://doi.org/10.3390/cancers14164041

[38] - Almaida-Pagan, P.F., Torrente, M., Campos, M. et al. Chronodisruption and Ambulatory Circadian Monitoring in Cancer Patients: Beyond the Body Clock. Curr Oncol Rep 24, 135–149 (2022). https://doi.org/10.1007/s11912-021-01158-z

[39] - https://www.h2020-faith.eu/ (as of July 2023)

[40]- Mattiuzzi C, Lippi G. Current Cancer Epidemiology. J Epidemiol Glob Health. 2019;9(4):217-222. doi:10.2991/jegh.k.191008.001

[41] https://www.definitivehc.com/blog/most-common-clinical-trials-by-therapy-area (as of August 2023)

[42] - Jastania R, Nageeti T, Al-Juhani H, et al. Utilizing Big Data in Healthcare, How to Maximize Its Value. Stud Health Technol Inform. 2019;262:356-359. doi:10.3233/SHTI190092

[43] - Shirwaikar RD, Acharya U D, Makkithaya K, M S, Srivastava S, Lewis U LES. Optimizing neural networks for medical data sets: A case study on neonatal apnea prediction. *Artif Intell Med*. 2019;98:59-76. doi:10.1016/j.artmed.2019.07.008

[44] - Wade, D. (2007). Ethics of collecting and using healthcare data. BMJ, 334(7608), 1330–1331. doi:10.1136/bmj.39247.679329.80

[45] - https://www.cdc.gov/phlp/publications/topic/hipaa.html (as of August 2023)

[46] - https://gdpr-info.eu/ (as of August 2023)

[47] - International Journal of Enhanced Research in Science, Technology & Engineering ISSN: 2319-7463, Vol. 5 Issue 12, December-2016 Fakhrulddin, Mohammed. (2016). Penetration Testing on FTP Server.

[48] - https://www.rfc-editor.org/rfc/rfc114 (as of August 2023)

[49] - Chris Rapier and Benjamin Bennett. 2008. High speed bulk data transfer using the SSH protocol doi: 10.1145/1341811.1341824

[50] - Title: Securing FTP with TLS Author(s): D. Ford, P. Higginson, M. Thomas, P. Black Series: Request for Comments (RFC) Number: 4217 Publisher: Internet Engineering Task Force (IETF) , October 2005, URL: https://tools.ietf.org/html/rfc4217

[51] - Zippia. "25 Amazing Cloud Adoption Statistics [2023]: Cloud Migration, Computing, And More" Zippia.com. Jun. 22, 2023, https://www.zippia.com/advice/cloud-adoption-statistics/

[52] - https://cloud.google.com/compute/docs?hl=en (as of August, 2023)

[53] - https://cloud.google.com/appengine?hl=en (as of August, 2023)

[54] - https://www.heroku.com/platform (as of August, 2023)

[55] - https://cloud.google.com/saas (as of August, 2023)

[56] - https://firebase.google.com/docs (as of August, 2023)

[57] - source of image: https://www.spaculus.org/google-cloud-computing-services, all rights belong to google and spaculus (as of August 2023)

[58] - Guoqiang Zhang, B. Eddy Patuwo, Michael Y. Hu, Forecasting with artificial neural networks:: The state of the art, https://doi.org/10.1016/S0169-2070(97)00044-7

[59] - Kriegeskorte N, Golan T. Neural network models and deep learning. Curr Biol. 2019;29(7):R231-R236. doi:10.1016/j.cub.2019.02.034

[60] - Renganathan V. Overview of artificial neural network models in the biomedical domain. Bratisl Lek Listy. 2019;120(7):536-540. doi:10.4149/BLL_2019_087

[61] - Liimatainen K, Huttunen R, Latonen L, Ruusuvuori P. Convolutional Neural Network-Based Artificial Intelligence for Classification of Protein Localization Patterns. Biomolecules. 2021 Feb 11;11(2):264. doi: 10.3390/biom11020264. PMID: 33670112; PMCID: PMC7916854.

[62] - https://atomicdesign.bradfrost.com/ (as of August 2023)

[63] - Torrente M, Sousa PA, Franco F, Guerreiro G, Sousa A, Parejo C, Pimentao J, Provencio M. Understanding prognosis and survival outcomes in patients with early-stage non-small-cell lung cancer. Clin Med (Lond). 2022 Jul;22 (Suppl 4):38-40. doi: 10.7861/clinmed.22-4-s38. PMID: 36220219; PMCID: PMC9600825.

[64] - HILL AB. THE ENVIRONMENT AND DISEASE: ASSOCIATION OR CAUSATION? Proc R Soc Med. 1965 May;58(5):295-300. PMID: 14283879; PMCID: PMC1898525.

[65] - Gresham G, Meinert JL, Gresham AG, Piantadosi S, Meinert CL. Update on the clinical trial landscape: analysis of ClinicalTrials.gov registration data, 2000-2020. Trials. 2022 Oct 6;23(1):858. doi: 10.1186/s13063-022-06569-2. PMID: 36203212; PMCID: PMC9540299.

[66] - https://pro2col.com/blog/ftps-sftp (as of July 2023)