

Machine Intelligence in Healthcare and Medical Cyber Physical Systems: A Survey

OMID RAJABI SHISHVAN^{ID}, (Student Member, IEEE),
DAPHNEY-STAVROULA ZOIS^{ID}, (Member, IEEE),
AND TOLGA SOYATA^{ID}, (Senior Member, IEEE)

The authors are with the Department of Electrical and Computer Engineering, SUNY Albany, Albany, NY 12203, USA

Corresponding author: Tolga Soyata (tsoyata@albany.edu)

ABSTRACT Today, the US healthcare industry alone can save \$300 B per year by using machine intelligence to analyze a rich set of existing medical data; results from these analyses can lead to breakthroughs such as more accurate medical diagnoses, discovery of new cures for diseases, and cost savings in the patient admission process at healthcare organizations. Because healthcare applications intrinsically imply a vast amount of data, the execution of any algorithm on medical data is computationally intensive. Significant advancements made in computational power in the past decade have provided the opportunity for many researchers to successfully implement various machine intelligence-based healthcare applications, which didn't run efficiently on earlier computational platforms. In this paper, we provide a survey of machine intelligence algorithms within the context of healthcare applications; our survey includes a comprehensive list of the most commonly used computational models and algorithms. We view the application of these algorithms in multiple steps, namely, data acquisition, feature extraction, and aggregation, modeling, algorithm training, and algorithm execution and provide details—as well as representative case studies—for each step. We provide a set of metrics that are used to evaluate modeling and algorithmic performance, which facilitate the comparison of the presented models and algorithms. Medical cyber-physical systems are presented as an emerging application case study of machine intelligence in healthcare. We conclude our paper by providing a list of opportunities and challenges for incorporating machine intelligence in healthcare applications and provide an extensive list of tools and databases to help other researchers.

INDEX TERMS Healthcare applications, medical decision support, machine intelligence, statistical signal processing, machine learning, data mining, feature selection.

I. INTRODUCTION

Improving healthcare by taking advantage of today's technology is a global interest; based on a 2011 report by McKinsey Global Institute [1], it is estimated that the US healthcare alone can save 300 billion dollars annually by analyzing the large corpus of healthcare data that has been accumulating for decades. This corpus of data includes patient health records (either in written or digital format) and past diagnoses and outcomes [2], [3]. Despite the availability of this large volume of data, the amount of time it takes to process it—to make practical inferences—is not a trivial task either for computers or healthcare professionals [4]. A set of algorithms designed to find statistical connections among events and results *efficiently* has been the focus of research in both electrical engineering and computer science disciplines for decades. These algorithms are mathematical tools with the

ability to “learn” input-output correlations and create an approximation to describe their relationship; they work on existing data and can find either patterns in the data or analytical relations between the input and output data.

Successful application of a variety of machine intelligence algorithms to medical data can revolutionize a wide range of medical applications by improving diagnosis accuracy, determining its cause and course of disease, and formulating an effective treatment for diseases, among others. For example, by training an algorithm on multiple images of a certain medical condition from a set of prior patients, the existence of that same medical condition can be detected in a new patient by using the new patient's image as the input to the algorithm. A remote health monitoring system consisting of body-worn sensors [5] that acquire data from a patient, transmit to the cloud, and process it using machine intelligence can improve

diagnostic accuracy, reduce healthcare costs due to the reduction in the time for in-patient care [6], and introduce new business opportunities [7].

For a majority of the algorithms, the prediction accuracy depends on an initial “training” phase, where the algorithm learns how to predict the output based on a set of training data, which includes known input-output pairs. This is in contrast to sets of algorithms that do not need any training, because their intended purpose is the discovery of patterns within the data, without designating any of the data as “input” or “output” to the algorithm [8]. The concepts of “training” and “pattern recognition” provide two counteracting forces in using machine intelligence in healthcare; while an increase in the amount of available training data increases the demand for computational resources, a decrease causes a reduction in algorithmic accuracy, despite a reduction in demand for computational resources. Because the priority for medical applications is typically algorithmic accuracy, it is reasonable to expect that future healthcare applications that incorporate machine intelligence will utilize large datacenters, potentially rented from cloud operators such as Amazon EC2 [9].

Before selecting a machine intelligence algorithm, the first step is generally the design of a *model* in which the form of the input and output values are specified, as well as a tolerable accuracy. The choice of algorithm depends largely on the type of data, complexity of the model, and the goal. Since checking and understanding the decision process of the model is sometimes of interest, interpretability of the model for humans is another issue. Hence, the models should show the relation between the variables and how each variable affect the decision process with ease.

The remainder of this paper is organized as follows: In Section II, we categorically list a set of healthcare applications that can benefit from machine intelligence and provide an overview of the algorithmic flow in these applications. In Section III, we provide a conceptual architecture of how machine intelligence algorithms are used in these applications. We study the challenges related to the acquisition of different types of medical data in Section IV and describe how raw data can be turned into feature vectors and algorithms to select the most useful set of these features in Section V. In Section VI, we study the metrics that are used to compare the performance of different models and algorithms. In Section VII, we elaborate on the design of models that are used to map healthcare applications to machine intelligence algorithms and study commonly used models in Section VIII. In Section IX, we provide an overview of the goals, characteristics, and performance metrics for the algorithms that are discussed in this paper and present an overview of a large list of commonly-used machine intelligence algorithms in healthcare applications in Section X. We dedicate Section XI to Artificial Neural Networks (ANNs), which demonstrated major success in the healthcare arena recently and received growing interest. In Section XII, we investigate Medical Cyber Physical Systems (MCPS), which are an emerging application of machine intelligence in healthcare.

We provide a list of challenges and opportunities in Section XIII to highlight the potential future applications of machine intelligence algorithms in the healthcare domain. Our concluding remarks are provided in Section XIV. In Appendixes A and B, we categorically enumerate a list of publicly-available medical *databases* and *tools*, respectively, to aid readers of this paper in finding readily-accessible data and tools for use in their research.

II. USING MACHINE INTELLIGENCE IN HEALTHCARE APPLICATIONS

In this section, we will discuss six specific healthcare application categories in Sections II-A through II-F, which represent the majority of existing applications that can benefit from the usage of machine intelligence.

A. CLINICAL DIAGNOSIS

In clinical diagnosis, machine intelligence algorithms recognize the existence of a symptom or a specific health condition in a patient, which enables real-time monitoring of patients via the analysis of large datasets that may include 3D images and long-term signal recordings. Following studies apply machine intelligence to Electrocardiogram (ECG) recordings of patients to detect the existence health hazards: Hijazi *et al.* [10] determine their likelihood of having two types of arrhythmias (LQT1 and LQT2) and achieve a $\geq 70\%$ accuracy. Thakor and Zhu [11] take a different approach for detecting multiple types of arrhythmias by first removing the noise in ECG recordings and using adaptive filters and the least mean square algorithm. Bsoul *et al.* [12] use single channel ECG recordings to detect obstructive sleep apnea (OSA), thereby eliminating the need for a full sleep study and achieve accuracies as high as $\approx 90\%$ in some cases.

The following two studies demonstrate the diagnosis of cancer using machine intelligence. Mousavi *et al.* [13] identify glioma (a type of cancer that starts in the brain) in histopathological images of patients and classify the existence of glioma into one of two known categories: low-grade glioma and high-grade glioma. They achieve up to an 88% in recognition rate. The study presented in [14] investigates detecting the existence of melanoma using features such as texture and color identify normal/abnormal skin regions.

The following studies detect the existence of mental disorders by using machine intelligence. Patel *et al.* [15] develop a wearable monitoring system that utilizes accelerometers to derive gait and posture information, which identifies the severity of some Parkinson’s disease symptoms. The study reports error rates as low as 1.2%. Klöppel *et al.* [16] analyze the MRI brain images of patients to determine the existence of Alzheimer’s Disease (AD) or Frontotemporal lobar degeneration (FTLD). Their methods rate more than 90% accurate and in some cases are as accurate as 96.4%.

B. PROGNOSIS

In contrast to diagnosis, the term *prognosis* refers to the monitoring of a patient for a specific health condition and predicting how this health condition will evolve in the future.

Machine intelligence is a viable candidate for this application category because knowledge of the specific condition for which the patient is being monitored allows the algorithm to use a well-known model and clearly-defined input/output parameters, potentially yielding more accurate results. Neuvirth *et al.* [17] use machine intelligence for prognosis of diabetes patients to determine the probability of a patient requiring emergency care by analyzing patient claims, pharmacy purchases, lab test results, and personal profile. Another study in [18] uses patient MRI scans to classify a patient's mental state as one of **{Healthy, At-risk-of-psychosis}**; they achieve accuracies higher than 80%. They also detect the probability of a patient's transition from an early stage psychosis to late stage psychosis. Therefore, their application can be thought of as being dual-purpose: i) *diagnosis* of psychosis and ii) *prognosis* of the transition probability from one stage of psychosis to another.

Two other studies that use machine intelligence for prognosis is as follows: Tamaki *et al.* [19] use clinical records of school children to predict dental cavities. The study in [20] takes biometric, lifestyle, and demographic variables of individuals to predict the probability of hypertension. Their best model yields a 65% accuracy.

C. ASSISTIVE TECHNOLOGIES

This category of applications aims to utilize machine intelligence to provide assistance directly to a patient, rather than being used by the doctor. Typically, the input to an underlying machine intelligence algorithm in this category comes from one or more physiological bio-markers of the patient and the output of the algorithm is used to energize multiple actuators (e.g. a robotic or haptic device) or make suggestions to the user. In this category, the receiver of the machine assistance is not able to perform daily activities without some type of assistance, whether machine-based, by a medical professional, or by a family member.

In the iSTRETCH system [21], moderate-level stroke patients are provided with a haptic robotic device to assist their upper-limb reaching rehabilitation. This rehabilitation is typically provided within a hospital setting; an automated rehabilitation strategy was implemented that is $\approx 90\%$ in agreement with a physical therapist. Another study to assist dementia patients with their handwashing [22] is known as the COACH system, which tracks individuals with varying degrees of dementia using a single camera during hand washing and makes suggestions. A 25% reduction on caregiver interventions is reported. In [23], an application scenario for machine intelligence is described, where disabled individuals use intelligent robotic wheelchairs. A set of object recognition algorithms that track and learn the movements of the patient is utilized to provide assistance in using the wheelchair.

D. PERSONAL HEALTH MONITORING

Personal health monitoring applications differ from assistive technologies in that the intended recipient of the machine

intelligence-generated suggestions is a person who is capable of performing daily activities and the goal of using these suggestions is to improve and monitor personal health. This contrasts with assistive technologies, in which the users are not able to function on their own. Following studies use machine intelligence to make critical suggestions to patients who are suffering from an existing medical condition. Turksoy *et al.* [24] take continuous glucose measurements of type 1 diabetes patients in a system named *artificial pancreas control system*, which detects the probable meal that a patient had and its impact on glucose levels. The detection result is used to administer insulin boluses and prevent hyperglycemia. In another application presented in [25], authors focus on recovering stroke patients and their rehabilitation routine. Their system includes sensors on the bottom and back of patients' shoes to detect posture and activities.

The following studies investigate the use of machine intelligence in applications that detect physical activity and mood. MyBehavior system [26] is a smartphone app that infers an individual's physical activity and dietary behavior by utilizing both automated and manual data logging. This app uses the collected data to suggest changes that can lead to a lower calorie intake and more physical activity. In a test of the app involving 14 subjects, authors reported an overall statistically significant lifestyle change. In another physical activity detection study, Zois *et al.* [27] use a Wireless Body Area Network (WBAN) to detect activities of individuals in one of four different categories: **{Sit, Stand, Run, Walk}**. Their WBAN includes a mobile phone, three accelerometers, and one ECG sensor. Zhou *et al.* [28] study mood detection for people interacting with a computer and classify user mood as one of the three states: **{Positive, Neutral, Negative}**. They use multiple types of data (head movement, eye blink, pupil radius, etc.) as input to their algorithm. In another study that focuses on stress detection [29], authors use a wireless chest belt that reads ECG and respiration signals and a hand sensor that detects skin conductance and EMG signals. They classify subjects' mood as one of **{Stress, Non-stress}** conditions.

E. HEALTH-RELATED DISCOVERIES

Discovering previously unknown causal relationships in healthcare datasets is another application of machine intelligence. In these applications, existing databases, which may be months, years, or centuries old and may be gathered from diverse sources globally, are used to analyze the relationships among different variables. Social network data proves to be especially useful for this category, because the acquisition of data is not restricted to narrow geographic regions. Tatonetti *et al.* [30] use data mining techniques to detect previously unknown interactions between drug pairs in FDA adverse event reports. They hypothesize that a specific pair of drugs act together and leads to an increase in blood glucose levels; they use this cue as a guide and perform a clinical test to validate the effect in a real world scenario. A different study [31] aims to determine animals that may carry diseases that are transmittable to humans based on the knowledge

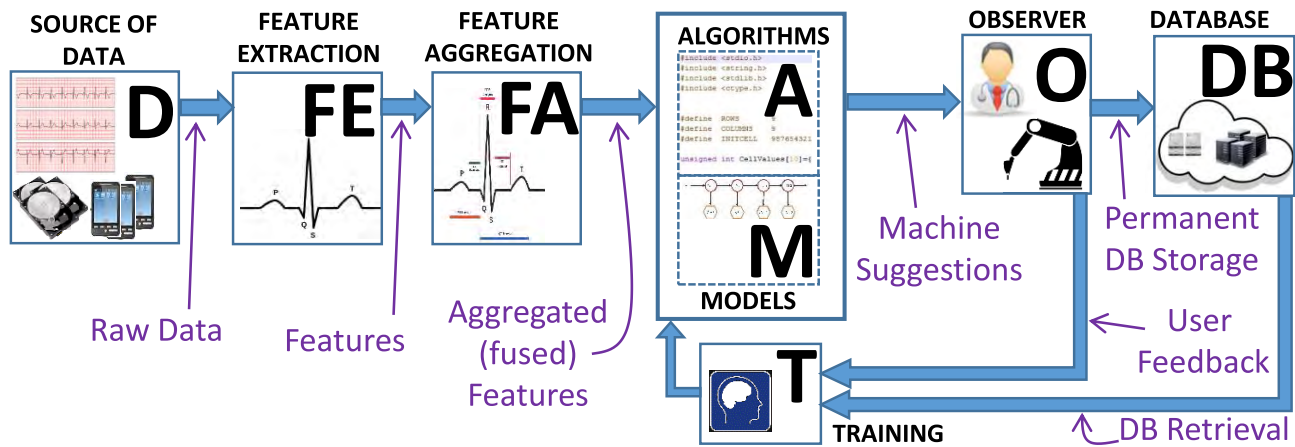


FIGURE 1. A conceptual diagram of a generalized health care application. Each functional unit is depicted as a box with an associated box number: **Box D** is the entry point of the data into the application from one or many sources. **Box FE** is the initial pre-processing step to reduce the amount of data that needs to be processed/stored by turning the raw data into a set of feature vectors. **Box FA** further reduces the number of features by aggregating (fusing) them. **Box A** and **Box M** form the machine intelligence computational core, where the latter is optional in some applications. **Box O** is typically the human that observes the data and uses it in some health care application. **Box T** is utilized in applications that require training for their operation. **Box DB** is the permanent storage point of patient medical –or environmental– healthcare data.

gained by observing Ebola outbreaks. The author argues that machine intelligence can help narrow down the geographical regions and the animals that may carry specific diseases.

F. BUSINESS ANALYTICS

Machine intelligence also has applications in the field of healthcare business management, in which the goal is to predict the existence of certain events to improve business outcomes (e.g., profits). Ginsberg *et al.* [32] use Google search entries to predict physician visits related to influenza-like illnesses days or weeks in advance to help healthcare centers manage their human resources more efficiently. The study in [33] focuses on analyzing claim datasets and detecting insurance fraud to reduce costs for insurance companies.

III. A CONCEPTUAL DIAGRAM FOR USING MACHINE INTELLIGENCE IN HEALTHCARE

A conceptual diagram is shown in Fig. 1 that unifies a majority of the applications in the healthcare domain. In this paper, we will continuously refer to this figure and will reference the number of the “box” as the point of interest; for example, we will use “observer” and “Box O” interchangeably. This will allow us to provide summarized information in a concise manner. Note that every application does not necessarily include all of the boxes shown in Fig. 1. For example, the MyBehavior system [26] does not include Box DB (Database), because the underlying machine intelligence algorithm only requires instantaneous patient data instead of stored data. In another example, Tatonetti *et al.* [30] describe data mining techniques to detect drug interactions that increase blood glucose levels; because their algorithms assume no previously known drug interactions, the concept of training —and consequently Box T (Training)— is not applicable in this case. We will now introduce the function of each box:

Box D (Source of Data): A health care application can use data from heterogeneous sources. For example, certain applications may require the collection of environmental information such as temperature, atmospheric pressure, accelerometer information, or location. Other applications can capture the health data from the remotely-monitored patient [34], [35] or crowd-sourced environmental information [36]. In either case, we assume that the output of this box is *raw data*. Section IV is dedicated to describing this box in detail.

Box FE (Feature Extraction): The amount of raw data exiting Box D is unmanageable for any computer hardware (or software) and is at the heart of the Big Data problem [4]. In addition to its insurmountable volume, this data is vastly redundant; a pre-processing step is always used to turn the raw data into *features* (or, alternatively, *feature vectors*). One example of feature extraction is the conversion of raw patient ECG data into the QT, and RR intervals [37], where the QT and RR are the intervals of a heart rhythm that consists of repeated Q, R, S, T, and U delineators [38]. In this application, 24-hour raw ECG data occupies 40 MB, while when converted to QT and RR intervals, only 10 KB is needed to store a patient’s 2-hour ECG recordings, thereby achieving a 40,000× data compression.

Box FA (Feature Aggregation): While the feature extraction step drastically reduces the amount of data to process —and transmit— there can still be a significant amount of redundancy in the output of Box FE, depending on the application. Combining features to obtain *aggregated features* (or, alternatively, *fused features*) allows machine algorithms to take advantage of the specific characteristics of an application to achieve higher accuracy or better run time [10], [35], [39]. The details of Box FE and Box FA —along with techniques to extract and aggregate features— will be provided in Section V.

Box M (Model): For some medical applications, a model of computation may exist, which can facilitate a better understanding of the application as well as determine some operational parameters. While a model is not necessary, it can substantially reduce the computational requirements. In a case model does not exist, the data flow is from the output of Box FE into Box A. Alternatively, a model may be used to extract features, effectively making Box FE and Box FA unnecessary. The requirements for models along with some known models in healthcare applications are studied in Sections VII and VIII.

Box A (Machine Algorithms): The heart of health care applications is Box A, where one or more algorithms execute to provide an output for presentation to the observer. The output of Box A can be as simple as a “threshold alert” that warns the healthcare professional about a patient health condition (e.g., heart rate going above a given threshold [38]) or as sophisticated as an elaborate suggestion about a pattern of a bio-marker resembling a certain disease (e.g., the QT interval of a patient suggesting the existence of the Long QT syndrome (LQTS) heart condition [10]). Majority of this paper is devoted to the categorization and the description of different types of algorithms that make up Box A starting with an overview of the algorithms in Section IX-A and the metrics involved in gaging the performance of an algorithm in Section VI.

Box O (Observer): This box is conceptualized to be the receiver of the suggestions from the machine. For example, in the case of medical decision support, the observer is a doctor or a nurse who is continuously monitoring a patient for potential health hazards. In case the machine makes a suggestion about a health hazard, the suggestion is analyzed by the health care professional and an appropriate action is taken [34], [40]. The suggestion could simply be ignored if it is believed to be faulty. Note that, in applications such as robot-assisted surgery, the observer is not necessarily a human. The actuator can be the observer that takes commands from the algorithm and actuates automatic suturing devices to assist a surgery [41]. The functionality of Box O is intertwined with Box T and will be detailed in conjunction with one another in Section IX-B.

Box T (Training): While certain applications intend to provide a one-way output (i.e., feed-forward) to the observer, many applications take the observer’s input and re-apply (i.e., feedback) it to a training algorithm (Box T) to continuously *learn* from the feedback. Training is a crucial component of every algorithm that is described in this paper—with the exception of data mining algorithms—and the accuracy of the machine learning algorithms depend significantly on the training phase of that algorithm. The problem of *over-training* exists in the functionality of this box, as will be detailed in Section IX-B.

Box DB (Database): Patient medical data is stored in a permanent database for future use. The storage time of this data is mandated by the law in many cases; for example, HIPAA regulations in the US require the retention of patient

medical records for at least 6 years. This retention period is even longer in some states [42]. Although raw medical data can be stored, this can cause an overload in cloud storage. Almost every algorithm described in this paper works with feature vectors, rather than raw data; therefore, storing the output of Box FA proves sufficient in many applications, however, a challenge exists in determining what happens when a newly-developed application requires different feature vectors for its operation that cannot be derived from the existing ones.

IV. DATA SOURCES (BOX D, BOX DB)

The machine intelligence algorithms described in this paper can take a variety of different data types as their input, as long as the utilized data has a statistical importance in the application that can lead to the generation of the desired outcomes. In this section, these different types of data are studied categorically.

A. STATIC PERSONAL DATA

For machine intelligence to produce an initial “rough-estimate” decision support results, a personal database can be utilized. Such personal data can include family history, smoking habits, gender, weight, and ethnicity. For example, statistically it is a known fact that female QT intervals in ECG recordings are ≈ 20 ms higher than male QT intervals [38]. Furthermore, smokers are more likely to have cardiac problems [43]. Having a recorded database of a patient’s personal information can provide highly valuable input to the algorithms.

B. SHORT TERM PERSONAL RECORDINGS

In applications where a decision has to be made regarding whether a patient requires immediate attention (e.g., an emergency room), the recorded data in the first 5–10 minutes— from the time arrival time of the patient— has significant importance. For example, in the THEW ECG database [2], patients that came to the emergency room with chest pain and had high levels of Troponin in their blood had a much higher mortality rate, because the existence of Troponin signals a cardiac dysfunction. This enzyme should never “leak” into the blood from the heart.

C. UNSTRUCTURED DATA

While the previously mentioned ECG recordings are stored in databases that have a well-defined *structure* (e.g., ISHNE format [44]), some medical information in databases that do not necessarily have not every storage format allows easy access to data; these *unstructured* databases can either be converted to a structured format or algorithms that can handle unstructured data can be utilized to process them. For example, Weng *et al.* [45] take unstructured texts of clinical notes and by using natural language processing algorithms, they classify those notes to indicate which medical subdomain they belong to. In another example presented in [46], authors

study unstructured electronic health records of breast cancer treatments for enhancing precision medicine.

D. LONG TERM PERSONAL MONITORING DATA

Data obtained through a longer observation period, such as 12–48 hr, is very important due to the fact that short-term recordings at a hospital may miss crucially important details about a patient's health condition. An example is a study in [38] that uses Holter recordings. This study shows that the QT interval of an ECG can point to significant cardiac hazards at night and can be missed at the hospital using the short term ECG recordings. An open-source tool to visualize a patient's 24 hr ECG recordings is introduced in [47] to readily visualize such abnormal patterns over a longer period. An important characteristic of the data in this category is that it is not available until the end of the recording period, therefore it can only be used to track long-term trends by a machine intelligence algorithm, rather than providing a mechanism for real-time detection/intervention.

Another example of this type of data is the recordings obtained by the skin-worn sensors by a bio-sensor manufacturer MC10 [48]–[50], which record their data into the memory that is inside the sensor and transmit it into a computer at the end of the day using an RF-based powering and charging mechanism.

While Holter devices and MC10 skin-worn sensors are non-invasive recording devices, some sensors are implanted in a patient's body through some form of surgery; for example, implantable loop recorders [51] are placed inside a pocket created under the skin and are able to record ECG data. This data can be downloaded by physician during a visit by a special programmer. Loop recorder batteries may last for more than 2 years and recording may be activated and deactivated by the patient using an activator.

E. REAL-TIME PERSONAL MONITORING DATA

With the emerging Medical Cyber Physical Systems (detailed in Section XII), long-term data from patients can be obtained in *real-time* (Section XII-A), rather than *at the end of an observation period* (Section IV-D). This has major implications on the type of machine intelligence algorithms that can be used for health monitoring, as well as the applications of it. With real-time monitoring, not only the response to urgent health hazards can be much faster than long-term monitoring, but also the necessity for the patients to bring back the monitoring devices is eliminated.

F. EXTERNAL DATA (ONLINE)

Some machine intelligence-based applications gather auxiliary data from online queries and use them as data sources. This data can supplement the data already being acquired from the user or other external devices. A framework called ContextProvider [52] takes GPS and network-based positions, accelerometer and magnetic orientation data, weather conditions and forecast, and phone calls and SMS usage data—in addition to direct questions from the user—to

build a context-aware health monitoring system. In another application presented in [53], a warning system monitors the GPS data and queries the air quality from online sources to determine if an asthmatic patient's health is threatened.

G. CHALLENGES IN DATA ACQUISITION

Medical data differs drastically from other forms of data in that knowledge about one's health status can be used maliciously. Therefore, handling of personal health information is strictly mandated by HIPAA laws in the United States [54].

1) DATA PRIVACY

HIPAA laws place restrictions on how medical data is transported and stored. For the transportation of the medical data, Kocabas *et al.* [55] survey a set of encryption algorithms that make the data unreadable to someone who does not have the decryption key, rendering the data inaccessible to adversaries with malicious intent. For the storage of the data, a set of *data obfuscation* methodologies prevent the data from being *identified*, although the data can be readily accessed. As an example, Murphy and Chueh *et al.* [56] obfuscate the queries retrieved from a database so that they prevent the identification of individuals when adversaries narrow down their search criteria.

2) NOISE IN MEDICAL DATA

Noise in medical data varies based on the source of the data and each type of data may require specific preprocessing to increase the signal to noise ratio. Some examples of noise in medical data are as follows: (i) noise in MRI images tend to have a Rician distribution [57]; (ii) Lu *et al.* [58] demonstrate that CT images have noise distribution of a Gaussian function rather than the usually-assumed Poisson distribution; (iii) EMG signals manifest themselves as noise induced on the power spectrum of ECG recordings [59]; and (iv) noise introduced by eye movements interferes with EEG recordings [60].

3) MEASUREMENT ERRORS

Measurement errors and errors in software packages that interpret the raw medical data are a major issue in healthcare databases. Eklund *et al.* [61] show that most common software packages that analyze fMRI data can result in 70% false positive rates, which is much higher than the generally-acceptable threshold of 5%. Kimberlin and Winterstein [62] discuss multiple aspects of the reliability and validity of measuring instruments used in medical research for evaluating the quality of measurements. They raise issues about the variability in self-reported measurements and the repeatability of research outcomes.

4) MISSING DATA

Missing data in medical databases is also common. Burton and Altman [63] study 100 published research papers on cancer and report that 81 of the studies showed evidence of missing data. Many databases report whether they have missing

data or not; for example, UCI machine learning repository [3] has a “Missing Value?” field for all of its datasets, which shows if the dataset is complete or some entries are missing. Statistical research techniques to impute missing data has been successfully applied to medical datasets [64].

5) OUTLIERS

Outliers in datasets are another source of uncertainty. In certain research areas, such as fraud detection in healthcare providers, anomalies and outliers are *the* data that researchers seek; in other cases, removing outliers is a necessary step for the consistency of data [65].

H. DATA VARIABILITY AMONG INDIVIDUALS

Since each individual is unique, their physiological signals capture this uniqueness. As a result, the features that describe the same disease may take different values depending on the patient characteristics or even different features may be more informative than others. Therefore, it is necessary to develop personalized training models to ensure that the accuracy of the machine intelligence algorithms is maximized. For example, in [66], personalized training (models and time periods), features, and other model parameters are used to detect the physical activity of an individual as well as determine the optimal sampling rate of the sensors in a WBAN. The authors illustrate —via numerical simulations— that both the accuracy and the lifetime of the WBAN is maximized using personalized training. Swan [67] discuss consumer personalized medicine, where the main idea is to tailor therapies (e.g., drug delivery and dosage) to individuals based on their specific biological characteristics. Typical examples of personalized medicine are personalized genomics services, blood and other biomarker testing, environmental testing, and predictive biosimulation.

I. HISTORICAL DATA AND DATA STANDARDS

Healthcare data is provided in various types and formats. These formats include the type of the data (image, text, video, etc.), the format that the data is stored in, and established standards that the information follows. We elaborate on the data types and standards in Appendix A.

V. FEATURES (BOXES FE AND FA)

In machine learning and pattern recognition applications, the goal is to reach a decision regarding the category of a pattern starting from raw data [68]. As an example, consider the case where we have access to clinical observations of patients and our goal is to determine their health status, e.g., healthy vs. suffering from Parkinson’s disease. Any information *beyond what is required for our algorithm to achieve a specified accuracy* can be omitted during the execution of this algorithm. A *feature* is a measurable property of reduced dimension, as compared to the original data, which is extracted from raw data and captures useful discriminative characteristics that relate to the phenomena being observed. All that is needed for our algorithms is the *features*, rather

than the highly-redundant raw data. Typical examples of features are **(i) temporal features**, such as the mean, minimum, or maximum of a biological signal over an observation interval (e.g., heart rate [38], acceleration [39]), **(ii) spectral features**, such as the spectral entropy and the mean frequency of the magnitude of a biological signal in the frequency domain, **(iii) cepstral features**, such as the cepstral power and the Mel-Frequency Cepstral Coefficients (MFCC) [69], and **(iv) application-specific features**, such as the QT and RR intervals of an ECG recording [10].

In medical applications, features fall into two main categories: **(i) application-independent** [29], [39], [70]–[77], and **(ii) application-specific** [12], [38], [78]. The latter are usually provided by medical domain experts after years of research and result in high accuracy. In contrast, application-independent features do not rely on any knowledge of the application. This eliminates the need for sophisticated procedures to acquire them and results in lower acquisition costs. As far as performance is concerned, the resulting accuracy is within acceptable ranges [29], [35], [39], [71]. In the following subsections, typical examples of such features that belong to various application categories are discussed.

Two crucial steps that ensure the effectiveness of the resulting features are *feature extraction* and *feature selection*. The goal of the former step is to extract a set of variables that represent the initial problem from the initial raw data, which enable the solution to be computed significantly faster due to the reduced number of computations resulting from this new representation [79]. The goal of the latter step is to select the best subset of the initially-extracted features so as to improve the accuracy, generalization, and interpretability of the results, as well as the computational complexity that is required to achieve these performance goals. Both methods are able to (i) identify the best features for each class, (ii) address the *curse of dimensionality*, which arises from the fact that the number of examples needed to train a classifier function grows exponentially with each added dimension, (iii) improve generalization performance, and (iv) facilitate a visualization and intuitive understanding of the problem solution.

A. FEATURE EXTRACTION

To identify the most representative variables, feature extraction [80], [81] either involves the extraction of standard application-independent and application-specific features or usage of various dimensionality reduction methods, in which case an appropriate transformation is applied to the original data —or features— to reduce their dimension. Typical application-independent and application-specific features will be discussed in Sections V-C and V-D, respectively. Some commonly-used feature extraction methods in medical applications are:

1) PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis (PCA) [82] is a dimensional-reduction technique that converts a set of correlated data

points into a set of linearly uncorrelated variables through an orthogonal transformation. The main idea behind PCA lies in determining the eigenvalues associated with the data point matrix of interest and removing the dimensions that exhibit variance below a certain threshold. In [17], PCA is employed to the problem of disease management for diabetic patients, where feature dimensionality is reduced significantly, while at the same time, 90% of the variance in the data is preserved. In [83], the first 12 principal components of PCA were used to classify normal/abnormal ECG signals, achieving a maximum accuracy of 96.88%. Finally, in [84], PCA is used to reduce 108 spectral variables to a minimal set, which covers 75% of the data variance for brain tumor classification.

2) KERNEL PCA

Kernel PCA (KPCA) [85] is an extension of PCA that performs nonlinear projection using an appropriately defined kernel function (e.g., polynomial, Gaussian). Similar to the original PCA, KPCA is employed for dimensionality reduction and feature extraction. For example, in [86], KPCA is used to extract features from single-photon emission tomography images to enable early Alzheimer's disease diagnosis.

3) CANONICAL CORRELATION ANALYSIS (CCA)

Canonical-Correlation Analysis [87] analyzes the correlation between two multivariate random variables. Assuming two random variables of $X = (X_1, \dots, X_n)$ and $Y = (Y_1, \dots, Y_m)$, CCA finds combinations of X_i and Y_j that are correlated to each other linearly. Chen *et al.* [88] use CCA to analyze EEG signals for a Brain-Computer Interface (BCI) application, in which the subjects select letters on an on-screen keyboard by visually focusing on them and the system determines the selected letters by acquiring and analyzing the EEG signals of the users. The acquired signals are treated as variables and their correlation coefficient with reference signals are calculated; the reference signal with highest correlation to the EEG signal is chosen as the selected keyboard letter. Authors show that they are able to detect the letters in ≈ 0.5 seconds with minimal error.

4) LINEAR DISCRIMINANT ANALYSIS (LDA)

Linear Discriminant Analysis (LDA) [89], [90] is a dimensionality reduction method that projects data points to a lower dimension space and selects the projection that ensures the separability between different data classes. Jen *et al.* [91] develop a chronic illness early-warning system, in which they have use LDA and feature selection techniques to determine the 5 most important risk factors (out of 53) for different classes of chronic illnesses. In [86], LDA is combined with KPCA with different kernels to reduce the dimensionality of single-photon emission tomography images for early Alzheimer's disease diagnosis.

5) GENERALIZED LDA (GDA)

Generalized LDA (GDA) [92] is a nonlinear extension of the LDA method, where the data is first moved to a higher

dimensional space and then standard LDA is applied. In [93], a heart arrhythmia classification algorithm is proposed, which employs GDA to reduce the feature dimensions from 15 down to 5. In [94], GDA is used within the context of optic nerve disease classification from visual evoke potential signals, where prediction accuracy is improved up to 10%.

6) CLUSTERING

Clustering refers to the task of mapping similar data points to the same group (i.e., cluster) and dissimilar data to different clusters. It is useful in a variety of tasks including data compression and reduction. For instance, in [95], k -means clustering is used to improve sleep classification using EEG signals by identifying appropriate weight factors for the features, which ensures that similar data points are clustered together. In [96], spectral clustering is used to reduce the dimensionality of a feature set containing texture and graph features extracted from breast tissue images to improve the automatic classification of low and high grades of breast cancer.

7) MULTIDIMENSIONAL SCALING (MDS)

Multidimensional Scaling (MDS) [97] is a nonlinear dimensionality reduction technique that facilitates the understanding of similarity among individual data points in a dataset. The main idea behind MDS is to find a low dimensional space such that the Euclidean distances among a set of data points are able to reproduce the distance matrix in the original space. For example, in [98], MDS is used to identify changing spatial patterns in measles morbidity data to predict measles epidemics in the USA. In [99], MDS is used to identify the key parameters that enable accurate Post Traumatic Stress Disorder (PTSD) diagnosis in police officers.

8) ISOMap

ISOMap [100] is a nonlinear dimensionality reduction method that reveals the underlying global geometry of a dataset. To this end, it estimates the intrinsic geometry of a data manifold using rough estimates of each data point's neighbors on the manifold. In [101], ISOMap is used to enable high-quality visualization of medical image data such as tomography and MRI images so as to facilitate the expert's image interpretation or diagnosis. In contrast, in [102], ISOMap is used to generate a low-dimensional embedding from brain MRI scans, based on which appropriate features are extracted that facilitate the classification between individuals with Mild Cognitive Impairment and Alzheimer's disease.

9) ARTIFICIAL NEURAL NETWORKS

Certain types of Artificial Neural Networks, such as autoencoders, deep belief networks, convolutional neural networks, as detailed in Section XI, are used to extract features from raw data [103]. For example, Auto-encoders [104] reduce the dimensionality of the data by learning its representation, while ensuring that the input can be accurately

reconstructed. In that sense, the lower dimensional data can be used as features. In [105], features from the outputs of a convolutional neural network are extracted and used as part of a computer-aided detection system to automatically identify micro-calcification clusters on digital mammograms for breast cancer diagnosis. In [106], deep belief networks are used to extract features from audio and video signals for emotion recognition purposes.

B. FEATURE SELECTION

Although a large set of features may be available, they do not necessarily contribute to the accuracy of an algorithm equally. Some of the features can be more informative, while others can be almost useless. An example study in [38] shows that to detect the LQT1 cardiac condition, even the same feature, calculated at different times, contains drastically different information; using only the information-rich features allows the utilized algorithm to reach desired accuracy orders-of-magnitude faster by only using a set of selected features. In general, the selected features must be informative, non-redundant, and fast to compute to ensure optimum overall system performance [79].

In many applications, *feature selection* algorithms are employed to select a subset of features to reduce the number of available ones according to certain evaluation criterion [107], [108]. The goal is to improve metrics such as accuracy, generalization, interpretability and computational complexity, similar to the goals of feature extraction. Because the initial set of features may be large and redundant in nature, a set of algorithms are employed that evaluate the importance of each feature and its relationship with others to determine the ones that can substantially improve the aforementioned metrics. Consequently, the feature selection process requires a search strategy to select candidate features and an objective function to evaluate them. The reason behind these two requirements is that in practice an exhaustive search over all possible features is a computationally intensive task. Search strategies can be roughly categorized as (i) *exponential* (e.g., exhaustive search, branch and bound, beam search), where a number of feature subsets that grow exponentially with the search space dimension are evaluated, (ii) *sequential* (e.g., sequential forward selection, sequential backward selection, plus-L minus-R selection, bidirectional search, sequential floating selection), where features are added or removed sequentially in the subset but suffer from local minima, and (iii) *randomized* (e.g., random generation plus sequential selection, simulated annealing, genetic algorithms), where randomness is incorporated into the search procedure.

Feature selection algorithms can be broadly categorized as (i) *filter methods*, (ii) *wrapper methods*, and (iii) *hybrid methods* [107]. Filter methods evaluate feature subsets with respect to their information content and their performance highly depends on the employed metric. Typical measures of information content are distance / separability measures (e.g., Euclidean distance, Mahalanobis distance), correlation measures (e.g., correlation coefficient), and information-theoretic

measures (e.g., mutual information). In contrast, wrapper methods usually evaluate feature subsets with respect to their predictive accuracy under a certain classifier (see Fig. 2). Finally, hybrid methods combine both of these approaches. In summary, the main difference between the filter and wrapper feature selection methods lies in the fact that they allow feature selection either with or without regard to a specific machine algorithm, which affects generality, accuracy and scalability. For more details on the feature selection problem and related methods, the reader is referred to [80] and [107]–[110]. Some commonly-used feature selection algorithms in medical applications are:

1) SEQUENTIAL BACKWARD SELECTION (SBS)

The SBS method [111] is a greedy algorithm that begins with a complete feature set and sequentially removes the feature that reduces a certain metric the least. Yu and Guan [112] use SFS to select 15 out of 31 temporal and image-based features to maximize the detection rate of clusters of micro-calcifications in mammograms, which are early indicators of breast cancer. SBS has also been applied to select between features extracted from a variety of physiological signals such as electromyogram (EMG), electrocardiogram (ECG), skin conductivity, and respiration changes for emotion classification within the context of psychophysiology [113].

2) SEQUENTIAL FORWARD SELECTION (SFS)

The SFS method [110], [114], [115] is a correlation-based greedy algorithm that begins with an empty set and continues to include features in the set in an effort to maximize a certain metric associated with the current subset. At each step it adds to the subset the feature that gives rise to the best outcome. For instance, in Thatte *et al.* [66] use the SFS algorithm with the symmetrical uncertainty (SU) metric [116] to select the most informative features from ACC and ECG raw data for physical activity classification. In [113], SFS is used to select features extracted from a variety of physiological signals (e.g., electromyogram, respiration changes) for emotion classification.

3) PLUS-L MINUS-R SELECTION (LRS)

The LRS method is a generalization of the SFS and SBS methods, where L or R features are added or removed depending on whether the set is empty or full. To reduce the feature set size in automatic cancer diagnosis, plus-2 minus-1 selection (i.e., forward-select 2 new features and backward-eliminate one feature) was used in [117], which resulted in the use of maximum 2 features out of the 7 available ones without compromising accuracy.

4) SEQUENTIAL FORWARD FLOATING SELECTION (SFFS)

The SFFS method [118] dynamically includes or eliminates changing number of features at each step. After each forward step, SFFS performs backward steps as long as the objective function increases. In [70], SFFS is used to examine which features are highly-correlated with the self-reported

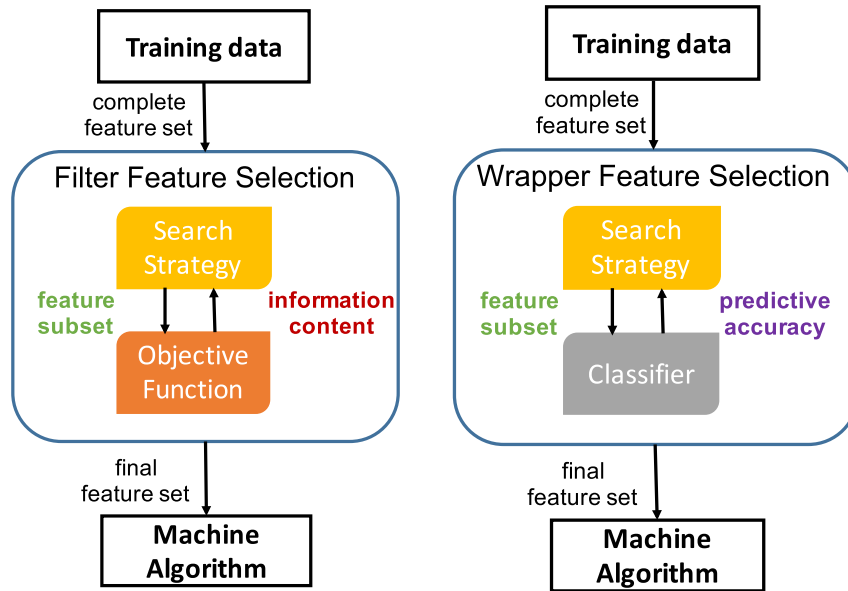


FIGURE 2. Feature selection methods structure: filter feature selection (left) and wrapper feature selection (right).

perceived stress scale ratings. SFFS is also used in [119] to select among a variety of 87 temporal and spectral features for breast cancer detection in mammograms.

5) SEQUENTIAL BACKWARD FLOATING SELECTION (SBFS)

The SBFS method [118] performs forward steps after each backward step as long as the objective function increases. In [120], SFFS and SBFS are employed for feature subset selection from ELM images for automatic melanoma recognition and an increase in the order of 5–10% is observed with only 10–15 features being selected out of the 122 available ones.

6) CORRELATION-BASED FEATURE SELECTION (CFS)

The CFS method [115] is a filter method that selects subsets of features that are not correlated to each other, but are highly correlated with the class of interest. In [121], the CFS method is employed to reduce the size of the features from 38 to 5 or 6 features without loss in the accuracy of stress detection using physiological and sociometric sensors. The CFS method is also used in [122] to evaluate the classification effectiveness of automatic feature selection for cardiovascular disease risk prediction compared to using domain knowledge. It is observed that automatic feature selection improves the predictive power of a classifier, while domain knowledge improves its sensitivity.

7) GENETIC ALGORITHMS (GA)

A Genetic Algorithm (GA) [123], [124] is a method of solving optimization and search problems based on natural selection, i.e., the process that drives biological evolution. More specifically, a GA starts with a population of individual solutions and selects at random a subset of them to produce

children. The population evolves toward an optimal solution. In [125], GA-based feature selection reduces the size of the feature set from 49 down to ≤ 10 for classification of liver tissue (i.e., normal, hepatic cysts, hemangiomas, hepatocellular carcinomas) using computed tomography (CT) images and shows improved classification performance. In [126], GA-based search is also employed for feature selection in the classification of “difficult-to-diagnose” micro-calcifications from mammography, where 20 out of the 40 available features are selected.

C. APPLICATION-INDEPENDENT FEATURES

Typically the goal of feature extraction is to use domain knowledge to extract characteristic information of reduced dimension from the raw data, which successfully represents the phenomenon of interest. However, there are certain types of features that are used in a variety of medical applications, independent from the application. In general, these features can be categorized as (i) temporal, (ii) spectral, and (iii) cepstral. In all of these cases, the amplitude of the medical biomarkers are treated as “signals.” Temporal features focus on the time-domain characteristics of these signals, while spectral features focus on their frequency-domain characteristics. Cepstral features show the rate of change in the spectral bands of a signal and can be thought of as being the spectrum of their spectrum. Note that although these features are gathered by applying the same techniques to different signals, the way they should be interpreted depends on the application and only the extraction part is application-independent.

1) TEMPORAL FEATURES

Typical examples of temporal features are statistical features, including —but not limited to— mean, standard

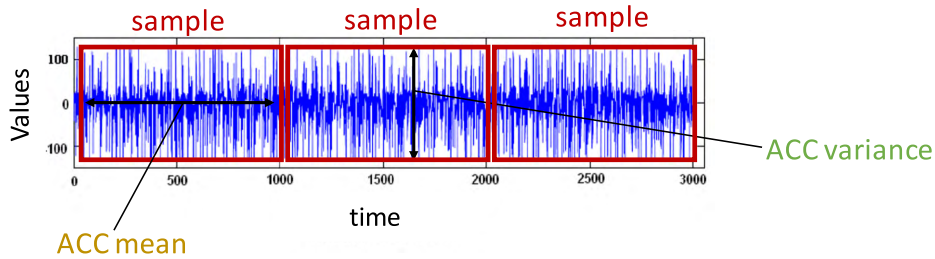


FIGURE 3. Example accelerometer (ACC) temporal features.

deviation, median, mean absolute deviation, kurtosis, skewness, zero crossing rate, mean of maxima, mean of minima, (10, 20, 50, 40, 60, 80, 90)th percentile, cross correlation, mean crossing rate, root mean square, correlation, and absolute value of slope. Fig. 3 illustrates some of these features for raw accelerometer (ACC) data. For instance, such “conventional” features (e.g., mean of maxima, mean of minima) are extracted from ACC and ECG raw data in [39] for physical activity detection. Similarly, a subset of the statistical features (e.g., median frequency) mentioned above are extracted from ECG, EMG and skin conductance raw signals for mental stress detection [29], [70]. In [71], various temporal features (e.g., correlation, absolute value of slope) are extracted from ECG and saturation of peripheral oxygen (SpO₂) signals for real-time sleep apnea detection. More complex processing of biometric signals can result in temporal features, which significantly enhance accuracy, such as the principal component analysis (PCA) error vector, the Hermite polynomial expansion (HPE) coefficients, the central tendency measure, and the Lempel-Ziv complexity. For instance, PCA error vectors have been extracted from ECG raw data and shown to improve physical activity detection [39] and body movement activity recognition [72]. On the other hand, exploiting HPE coefficients extracted from ECG raw data have also shown to improve physical activity detection [39]. Finally, the use of the central tendency measure and the Lempel-Ziv complexity of the SpO₂ signals can enhance real-time sleep apnea detection [71], [73].

2) SPECTRAL FEATURES

In many applications, frequency domain analysis of signals provides useful insights on the structural characteristics of signals. For example, careful frequency domain analysis of heart rate signals provides a unique understanding and a more precise assessment of various heart variability conditions [127]. Typical examples of spectral features are the spectral entropy, low/high frequency variability, median frequency and mean frequency of the magnitude, signal power in each frequency band, and mean phase angle. Figure 4 illustrates some of these features for raw audio data. In [128], spectral features such as the dominant frequency and the normalized spectral entropy are extracted from EEG raw data for automatic epilepsy detection. For the detection of micro-calcification clusters in digitized mammograms, the spectral entropy and the block activity are calculated in [129].

Average spectrum, proportion of low frequency energy under 500Hz/1000Hz, the slope of spectral energy above 1000 Hz, the Harmonic-to-Noise ratio and a variety of other spectral features are extracted from speech signals in an effort to detect sleepiness in [130].

3) CEPSTRAL FEATURES

In certain cases, filtering out the artifacts caused by irrelevant parameters (e.g., sensor displacement, skin muscle activity) can be achieved by using cepstral features to model the frequency information of the signal of interest (e.g., heart rate, moving pace). The cepstrum of a signal is defined as the inverse Fourier transform of the logarithm of its estimated spectrum [131]. The cepstrum of an ECG signal is shown in Fig. 5. Typical examples of cepstral features are the cepstral power, the MFCCs and their derivatives. Features such as the above are extracted from ECG raw data for physical activity detection [39] and arrhythmia detection using neural networks [74], from EMG data to automatically classify head and hand movements [75], from vibroarthrographic (VAG) signals to characterize the knee-joint pathology [76], and from breath sound signals to detect existing breath problems [77].

D. APPLICATION-SPECIFIC FEATURES (BOX FE)

Many applications have their own specific features, based on their specific data format. These features are extracted from the raw data and provide useful information for both the humans and the machine intelligence algorithms. Some examples of these features are as follows:

1) ECG WAVEFORM FEATURES

When working with cardiovascular diseases, a set of features that are extracted from ECG signals—such as the QT and RR intervals—are commonly used (see Fig. 6). QT interval is the time from the QRS onset to the end of T wave in an ECG waveform, whereas the RR is the R wave-to-R wave interval that designates the time between two heart beats [47], which is directly related to the heart rate. These features can be used in machine intelligence algorithms on their own or can be “fused” together to create fewer, yet better features.

2) EEG WAVEFORM FEATURES

EEG signals that are captured by sensors placed on the scalp, are variables that change through time and do not have a

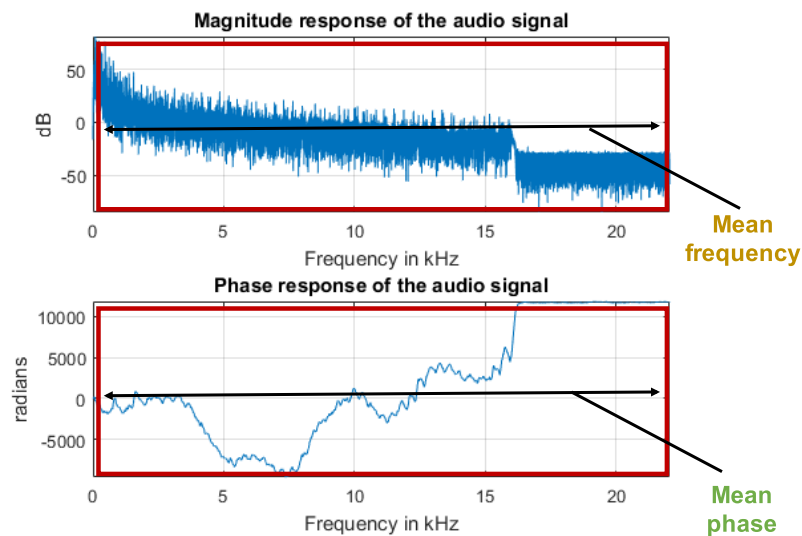


FIGURE 4. Example audio spectral features.

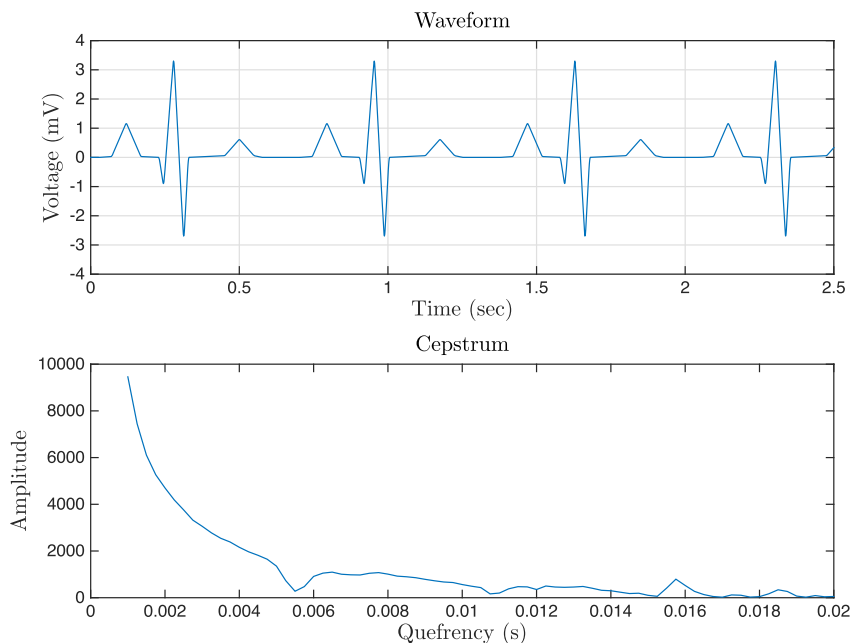


FIGURE 5. An example ECG cepstrum.

certain structure like the ECG signals. Some of the basic features of EEG signals are the power of the signals in different bandwidths. For example, Delta wave represents the EEG signal having frequencies between 0.5 and 4 Hz, while Alpha waves cover the frequencies between 7.5 and 12.5 Hz. There are other types of waves covering different spectrum of EEG signals and each of them can be used as a feature.

3) HEIGHT AND WEIGHT

In many of the medical reports, height and weight of the subjects are reported. These features can be used as an input

for machine intelligence algorithms and are shown to be an important factor in predicting disease outcomes; for example, weight is a useful feature when detecting cardiovascular disease probabilities.

E. APPLICATION SPECIFIC FEATURE FUSING (BOX FA)

In some applications, multiple simpler features can be combined (*fused*) to create features that reduce computational complexity at the expense of higher pre-computation. Fusing features either removes a bias in the original information or provides new information that machine intelligence

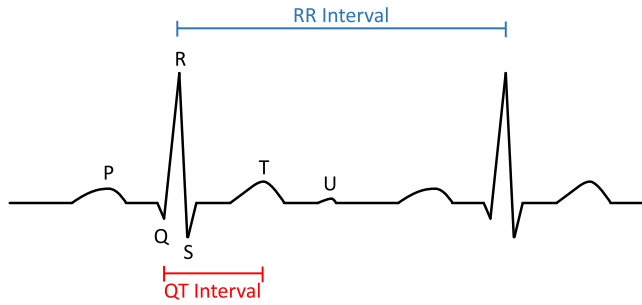


FIGURE 6. A sample waveform of a two heartbeat. QT and RR intervals are shown in the figure.

algorithms are unable to infer from the original features. Some examples of fusion are as follows:

1) BAZETT'S FORMULA/FRIDERICIA'S FORMULA

The QT feature mentioned in Section V-D1 is highly dependent on the heart rate and normalizing it to the heart rate (i.e., the RR interval) provides a more informative feature for the machine intelligence algorithms. Two commonly used formulas that provide the corrected QT (QT_c) are the Bazett's Formula (Eq. (1)) [132] and the Fridericia's Formula (Eq. (2)) [133].

$$QT_{cB} = \frac{QT}{\sqrt{RR}} \quad (1)$$

$$QT_{cF} = \frac{QT}{\sqrt[3]{RR}} \quad (2)$$

These QT correction formulas provide a feature by combining two simpler features that is a better indicator for clinical uses such as detection of long QT syndrome [38].

2) DISCRETE WAVELET ANALYSIS OF EEG SIGNALS

As mentioned before, different frequency bands of an EEG signal can be used as features in machine intelligence algorithms, but using an extra step on the data by passing them through a discrete wavelet transform provides features both in frequency and time domain. This decomposition of signal has been more successful when used for applications such as epileptic seizure detection [134].

3) BODY MASS INDEX (QUETELET INDEX)/CORPULENCE INDEX

Quetelet Index [135] (Eq. (3)) and Corpulence Index [136] (Eq. (4)) are indexes that use weight and height of a person and "normalize" the weight with regard to the height of an individual. They provide a more informative fused feature as a result, because the two metrics they use (Mass, Height) are statistically correlated.

$$BMI = \frac{Mass_{kg}}{Height_m^2} \quad (3)$$

$$CI = \frac{Mass_{kg}}{Height_m^3} \quad (4)$$

VI. PERFORMANCE METRICS FOR MODELS AND ALGORITHMS

Accuracy of models and machine intelligence algorithm outputs is determined by a fair metric that indicates if the model can successfully describe our empirical data or the algorithm is successful in fulfilling its task. In this section, different metrics are studied that are developed—and are in use—to (i) assess goodness of fit between model and data, (ii) compare different models, (iii) quantify the quality of predictions associated with a specific model, (iv) assess the performance of an algorithm, and (v) compare different algorithms. The type of the metric used depends on the context of research and the type of model/algorithm used.

A. ERROR METRICS

When evaluating the accuracy of a model or an algorithm, it is necessary to calculate the cumulative error it makes for a given set of input and output values. The most commonly used to metrics are described below.

1) MEAN ABSOLUTE ERROR (MAE)

MAE is a metric that shows the average difference between the absolute value of the estimated values and the observed values of a phenomenon and is calculated as shown in Eq. (5).

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (5)$$

Ng *et al.* [137] create a surgery duration prediction system and use MAE as one of the metrics that evaluates their model. Correctly predicting the duration of surgeries helps in maximizing the utilization of surgery rooms. Authors develop multiple regression systems and report RMSE and MAE of the surgery duration predictions versus the real duration.

2) ROOT MEAN SQUARE ERROR (RMSE)

RMSE is used to determine the squared difference between predicted values and observed values (i.e., the error). It is identical to the MAE with the exception of squaring each error, which tends to amplify the impact of the errors that are large. RMSE is calculated using Eq. (6) where n is the number of observations, \hat{y}_i and y_i are the estimated and observed values for the i^{th} observation, respectively.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (6)$$

Zhang *et al.* [138] develop a healthcare recommendation system that makes personalized suggestions to its users about choosing the best healthcare provider. Their system takes available patient ratings and reviews for doctors in addition to the preferences of the user and specialties of the doctors to recommend the best matching doctor for them. To analyze the performance of their system, they predict the rating that a patient will assign to a recommended doctor and compare it

TABLE 1. Confusion matrix.

		Predicted Condition	
		Positive	Negative
Actual Condition	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

to the real rating that the patient submits and report the RMSE of these predicted and actual ratings.

B. CONFUSION MATRIX AND RELATED METRICS

A confusion matrix is used to describe the performance of a classifier. A sample confusion matrix for a classifier with two possible outputs (“Positive” and “Negative”) is shown in Table 1. True Positive (TP) denotes the condition where the true condition is positive and is also correctly predicted as positive; for example, assume the case where a patient has a cardiac hazard and an algorithm predicts the condition as “this patient has a cardiac hazard.” This *Positive* prediction is correct (*True*) and the algorithm’s output is considered to be a *True Positive*. Alternatively, assume that the algorithm predicts that “this patient is healthy.” This *Negative* prediction by the algorithm is clearly incorrect (*False*) and is therefore called a *False Negative*. Similarly, *False Positive* corresponds to the case where a healthy patient is detected as a patient with a heart condition, whereas *True Negative* corresponds to the case where a healthy person is correctly identified by the algorithm as healthy. Many other metrics are defined by using the entries in Table 1; we will introduce the most commonly used ones in this section.

1) ACCURACY

Accuracy indicates the percentage of the prediction by an algorithm that is correct, as defined by Eq. (7); this includes cases where a positive case was predicted as positive (TP) or a negative case was predicted as negative (TN).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (7)$$

Accuracy is used widely in the literature. As an example, Hijazi *et al.* [10] report the accuracy of their LQTS syndrome classification system; their algorithm predicts the output correctly 70% of the time, corresponding to an accuracy of 70%.

2) SENSITIVITY/RECALL/TRUE POSITIVE RATE (TPR)

These three terms refer to the same metric. Sensitivity (Eq. (8)) denotes the percentage of positive predictions by an algorithm, when the actual condition is indeed positive; note that the actual condition being positive covers the first row of Table 1, which includes the cases where the algorithm made a positive prediction (TP) and a negative prediction (FN).

$$Sensitivity = Recall = True Positive Rate = \frac{TP}{TP+FN} \quad (8)$$

3) SPECIFICITY/TRUE NEGATIVE RATE (TNR)

These two terms refer to the same metric. Specificity (Eq. (9)) denotes the percentage of negative predictions by an algorithm, when the actual condition is indeed negative; note that the actual condition being negative covers the second row of Table 1, which includes the cases where the algorithm made a positive prediction (FP) and a negative prediction (TN).

$$Specificity = True Negative Rate = \frac{TN}{TN + FP} \quad (9)$$

Specificity and sensitivity are usually paired together. For example, Kushki *et al.* [139] detect physiological anxiety arousal in children and in addition to Accuracy, they report (Sensitivity = 0.99 = 99%) and (Specificity = 0.92 = 92%) for their proposed method.

4) PRECISION

Precision (Eq. (10)) is the ratio of the correct positive predictions to the sum of all positive predictions. To phrase alternatively, this metric denotes what percentage of the positive predictions are correct.

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

Precision is also usually paired with recall (sensitivity) when it is reported as a performance metric. Mozos *et al.* [121] report their results on stress detection models using Accuracy, as well as Precision and Recall. Their precision varies from 0.89 to 0.99, depending on the participant.

5) LIKELIHOOD RATIOS IN DIAGNOSTIC TESTING

Likelihood Ratio (LR) [140] is the ratio of the probability that a test result is correct to the probability that a test result is wrong for a diagnostic application. Two distinct related metric are defined.

- **LR+**: Probability of correct positive predictions divided by the probability of incorrect positive predictions, as formulated by (Eq. (11)), as shown at the top of the next page.
- **LR-**: Probability of incorrect negative predictions divided by the probability of correct negative predictions, as formulated by (Eq. (12)), as shown at the top of the next page.

LR+ and LR- metrics are used in [141] to show the performance of their sleep apnea prediction model. They report LR+ of 2.8 and LR- of 0.46 for their work. Note that this metric should not be confused with “Likelihood Ratio test” introduced in Section X-K.

6) GEOMETRIC MEAN

Imagine a database that contains input-output pairs, where the outputs denote a patient being “healthy.” Assume that this database provides input-output pairs for only 10 healthy patients and 990 unhealthy patients. Because of providing a significant amount of input-output pairs for unhealthy

$$LR+ = \frac{\text{Sensitivity}}{1 - \text{Specificity}} = \frac{\Pr(\text{Prediction}=\text{Positive} \mid \text{Actual}=\text{Positive})}{\Pr(\text{Prediction}=\text{Positive} \mid \text{Actual}=\text{Negative})} \quad (11)$$

$$LR- = \frac{1 - \text{Sensitivity}}{\text{Specificity}} = \frac{\Pr(\text{Prediction}=\text{Negative} \mid \text{Actual}=\text{Positive})}{\Pr(\text{Prediction}=\text{Negative} \mid \text{Actual}=\text{Negative})} \quad (12)$$

$$g = \sqrt{\text{Sensitivity} \times \text{Specificity}} = \sqrt{\frac{\text{TP}}{\text{TP} + \text{FN}} \times \frac{\text{TN}}{\text{TN} + \text{FP}}} \quad (13)$$

patients, this database is termed *imbalanced*. When testing an algorithm using such an imbalanced database, it is more suitable to use the Geometric mean metric, introduced in [142] and formulated in Eq. (13), as shown at the top of this page.

7) F₁ SCORE

F-score is a measure of an algorithm's performance using Precision and Recall.

test

The general form of the F-score is shown in Eq. (14), which puts a higher weight on Precision or Recall based on the value of β .

$$F_{\beta} = (1 + \beta^2) \frac{\text{precision} \times \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \quad (14)$$

The most common form of the F-score is the F₁ score (Eq. (15)), which is also defined as the *harmonic mean* of Precision and Recall.

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (15)$$

Su *et al.* [143] use the F₁ score and the geometric mean in addition to sensitivity and specificity to evaluate their pressure ulcer development diagnosis predictor. They reach an F₁ score of 0.8 and the geometric mean (g) of 0.88 in their study.

8) DICE COEFFICIENT

One of the areas of research that utilizes the F₁ score is image segmentation problems. F₁ score (which is also known as the Dice coefficient [144]) shows how a mask created by the machine that segments an image covers the real areas of interest in the image, like the areas with "tumors" in a photo. Dice coefficient is defined for two sets and is related (i.e., is equal) to F₁ as follows:

$$D = \frac{2 \cdot |X \cap Y|}{|X| + |Y|} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}} = F_1 \quad (16)$$

where X and Y are the two sets; the more elements they have in common, the closer the Dice coefficient is to "1." If the intersection of computer image segmentation and the actual segmentation is defined as TP (i.e., the True Positive ratio), the Dice coefficient is equal to the F₁ score.

Salehi *et al.* [145] develop a brain segmentation in 3D magnetic resonance imaging (MRI) system using convolutional neural networks (see Section XI-E). To show the effectiveness

of their scheme, they report Dice coefficient of more than 95% for two different datasets. Although their reported Dice coefficient outperforms the rest of the algorithms, they do not have the highest sensitivity or specificity when compared to the competing algorithms.

9) MATTHEWS CORRELATION COEFFICIENT (MCC)

Matthews Correlation Coefficient is a metric that is used to measure the quality of a binary classifier [146]. MCC is calculated according to Eq. (17) and takes values between -1 and 1.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (17)$$

An MCC value of 1 shows a perfect classifier, while a value of -1 shows that the classifier has predicted all of the values wrong. An MCC of 0 shows that the classifier has the same performance as a classifier that produces purely random outputs for every input.

Sakar *et al.* [147] use the MCC metric to evaluate the performance of their algorithms, which target diagnosing Parkinson disease through speech. They implement multiple algorithms and calculate the MCC value for each one; the performance of their best algorithm achieves an MCC value of 0.70.

10) COHEN'S KAPPA COEFFICIENT (KAPPA VALUE)

Cohen's kappa coefficient [148] is a metric that, when used in classification problems, removes the effect of random classification accuracy from the achieved accuracy. As an example, assume a dataset with two classes of data and equal number of data points belonging to each class; a classifier with random output will achieve 50% accuracy for this dataset. This makes it feasible to report accuracies of other algorithms with respect to the accuracy of a this algorithm with random outputs. Cohen's kappa coefficient, as formulated in Eq. (18), adjusts the accuracy of an algorithm by taking the random accuracy into account.

$$\kappa = \frac{\text{acc} - \text{rand}}{1 - \text{rand}} \quad (18)$$

In Eq. (18), "acc" is the accuracy of the algorithm and "rand" is the accuracy that an algorithm with random outputs achieves.

Tabar and Halici [149] classify EEG motor imagery signals and the accuracy as well as the κ value for 9 subjects

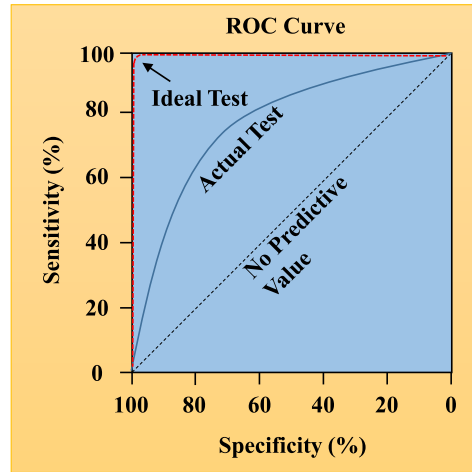
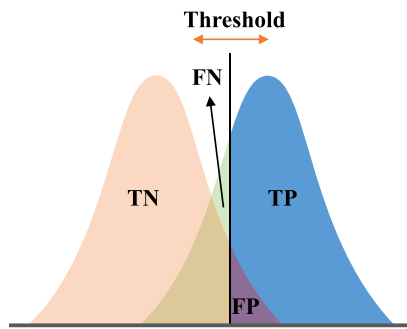


FIGURE 7. An example binary classifier that uses a threshold (left) to determine the regions of the input values that must be classified as TP, TN, FN, and FP. A sample ROC curve (right) that shows the performance of a classifier by plotting Sensitivity vs. 1-Specificity. Notice the labeling on the x axis (100% to 0%), which allows plotting 1-Specificity.

separately for their classification scheme. They report an average κ of 0.55 and $\kappa = 1$ for one of their subjects.

C. ROC (RECEIVER OPERATING CHARACTERISTICS) CURVE

A classification algorithm maps continuous input values to discrete output values; in determining which one of the discrete output values that an input value belongs to, it uses a *threshold* value, which separates the points at which the output is considered to be one of the allowed discrete values. This separation is exemplified for the case of binary classification in Fig. 7, which shows that the values of TP, TN, FP, and FN depend on this threshold and changing the threshold affects the percentages of data points that are truly or falsely classified as positive or negative. This means that Sensitivity and Specificity of an algorithm also depends on this threshold value. An ROC curve shows the dependence of an algorithm’s Sensitivity and Specificity on the threshold value by plotting these two metrics for different threshold values. In practice, Sensitivity vs. 1-Specificity is plotted, which is observed by the reverse labeling, i.e., from 100% down to 0% on the right side of Fig. 7. Although the ROC curve is drawn by sweeping through multiple threshold values, these threshold values are not visible on the ROC curve; each value of the threshold corresponds to a different point on the curve, i.e., a different (Sensitivity, 1-Specificity) pair.

1) AUC (AREA UNDER CURVE)

One of the main metrics derived from the ROC curve is the area under it, which is called Area Under Curve (AUC) [150]. AUC ranges between 0.5 and 1; an AUC of 0.5 represents a worthless algorithm, whereas an algorithm with perfect performance reaches an AUC of 1. ROC curve and AUC are among the most widely reported results in the literature; for example, Kurt *et al.* [151] build multiple coronary

artery disease prediction models and report their Sensitivity, Specificity, and plot the ROC curve; they report AUC values between 0.675 and 0.783 for their algorithms by using their ROC curve.

2) GINI COEFFICIENT

Although the Gini coefficient was initially introduced to identify income inequality within a group of people, it also found use in medical applications to describe distinguishability of two different classes of data. Gini coefficient can be calculated using AUC as follows:

$$\text{Gini} = 2 \cdot \text{AUC} - 1 \tag{19}$$

An AUC of 0.5 will result in a Gini coefficient of 0 and a perfect AUC of 1 will yield a Gini coefficient of 1. So, the Gini coefficient is identical to the AUC with the exception of a 2x wider range to describe the same phenomenon. Nguyen *et al.* [152] classify EEG signals in a brain-computer interface study with various classification methods and report the F score, accuracy, and Gini coefficient as performance metrics. They show that their proposed Naive Bayes algorithm has a Gini coefficient of 0.8, which is higher than other algorithms.

D. PEARSON CORRELATION COEFFICIENT

The Pearson correlation coefficient (R) and its square (R^2) constitute metrics of linear correlation between variables, where the former one is computer as follows:

$$R = \frac{n \sum x_i y_i - n \bar{x} \bar{y}}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}, \tag{20}$$

where n is the number of samples, x_i and y_i are the values of the variables, and \bar{x} and \bar{y} are the mean of the samples

(Eq. (21)) defined as:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ and } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i. \quad (21)$$

R varies between -1 and 1 and R^2 of 0 shows no correlation between the variables and R^2 of 1 implies a perfect correlation between them.

Ma *et al.* [153] predict the effects of drugs based on their structure and use R^2 as a measure to show their success. They compare the R^2 value of their proposed algorithm (A deep neural network) with conventional algorithms (such as random forests) and show that the average R^2 of their algorithm is 0.411 as opposed to 0.361 for the other algorithms, which is a significant improvement in their field of study.

E. p -VALUE

In many cases, the results of machine intelligence algorithms must be checked for “statistical importance,” i.e., whether they are created by chance or reflect the correct statistical relationship between the data and the produced results. As the default, a “Null Hypothesis (H_0)” is assumed, which indicates “zero relationship between the data and the produced results,” i.e., results that could have been obtained by consecutive coin tosses. An alternative hypothesis is tested against it with the data provided by the algorithm. For example, Page *et al.* [38] analyze the ECG recordings of patients with LQT1, LQT2, and LQT3 syndromes and as a part of their study, they claim that median QTc values for men with LQT2 syndrome is longer when compared to median QTc values for men with LQT1 syndrome (471 ms vs. 455 ms); this constructs their alternative hypothesis H_1 and strong opposition to the null hypothesis H_0 must be provided; both of these hypotheses are shown below:

H_1 : {“Median QTc values for men with LQT2 syndrome is longer when compared to median QTc values for men with LQT1 syndrome.”}

H_0 : {“There is no difference between median QTc values for men with LQT2 syndrome and men with LQT1 syndrome.”}

To provide evidence to reject H_0 , they need to assume that H_0 is true and compute the probability of two random samples having median values of 471 ms vs. 455 ms, without any relationship to LQT1 or LQT2. This probability is called the p -value and in their case it is 0.03; a p -value of 0.03 means that there is only a 3% chance that the alternative hypothesis H_1 can be replicated with randomly selected set of ECG samples. Therefore, there is a 97% chance that the result is statistically-meaningful. A generally-accepted p -value to reject H_0 is 5%; p -values less than 5% provide strong evidence in rejecting the H_0 and supporting the alternative hypothesis H_1 . In the same study, authors elaborate on the QTc values at two specific time intervals, between 3AM-4AM and between 3PM and 4PM, for all patients with LQT1, LQT2, and LQT3 syndromes. For the cases of LQT1 and LQT2, authors claim that the QTc value difference between

these two time periods is significant and a p -value of $p \leq 0.01$ provides strong support for their claim. On the other hand, for LQT3, due to the limited number of patients (9 men and 5 women) the p -values are within the range 0.2–1.0, which means that they can not draw any conclusion from the QTc values between these two periods, as far as LQT3 is concerned.

F. CONCORDANCE INDEX (C-INDEX)

Algorithms used for survival analysis produce a “time-to-event” output, which indicates how long it will take for an event to happen for a given input. For example, predicting the next time that a person, who is recently discharged from a hospital, will be readmitted to the hospital again, is a form of survival analysis. Concordance index is a metric that is used to rate the performance of such algorithms. C-index is defined as “the probability that considering two randomly chosen patients, the one who is predicted to have a shorter survival time —by the algorithm— will actually end up having an actual shorter survival time in reality.” The mathematical definition of C-index is as follows:

$$C = \Pr \left(g(\vec{Z}_1) > g(\vec{Z}_2) \mid T_2 > T_1 \right) \quad (22)$$

where \vec{Z}_1 and \vec{Z}_2 are the two input vectors (e.g., ECG recordings of two patients), $g(\vec{Z}_1)$ and $g(\vec{Z}_2)$ are the estimated survival times for these inputs vectors (e.g., machine-estimated survival probability of these patients), and T_1 and T_2 are the *actual* survival times (e.g., the amount of time these patients lived) [154].

A C-index of 1.00 shows perfect prediction and a C-index of 0.5 corresponds to a completely random predictor. Note the resemblance between the C-index (which is defined for continuous values) and the AUC, which is defined for discrete classification outcomes. Khosla *et al.* [155] develop stroke prediction algorithms and one of the metrics they use is the C-index. They report C-indexes between 0.73 and 0.777.

G. SHAPE SIMILARITY METRICS - HAUSDORFF DISTANCE

For applications that deal with detecting a shape in an image, a metric is required to quantify the similarity of the shape produced by the algorithm to the the actual shape of the object in the image. Hausdorff distance is one of the metrics that is used to quantify the similarity between two shapes, which is defined as the biggest “smallest distance” between two sets; for each point in set A, the minimum distance to the corresponding point in set B is calculated and the maximum of these distances is selected as the Hausdorff distance. When the coordinations of the edges of a 2D shape is considered a set, Hausdorff distance can be defined as a metric that shows how two objects are similar to each other, in terms of their shape. For example, Chen *et al.* [156] perform gland segmentation on histology images using deep neural networks and to show how close their detected glands are to real glands; they report Hausdorff distance of less than 50 pixels for one of their algorithms.

H. BLEU SCORE

For applications that involve written text and language, the Bilingual Evaluation Understudy (BLEU) [157] is an algorithm that assigns a score to the quality of texts that are translated from one language to another by machine. Shin *et al.* [158] annotate chest X-ray images by generating a description for them and score the effectiveness of their annotation generation using the BLEU score.

I. GOODNESS OF FIT TESTS

There are many tests that show if a produced distribution of a model's (or algorithm's) results represent the actual dataset or they are due to chance. These tests, known as *goodness of fit tests*, are used in many healthcare related machine learning applications. They include Hosmer-Lemeshow test, Chi-square test, and Kolmogorov-Smirnov chart and are studied below.

1) CHI SQUARED (χ^2) TEST

The chi-squared test is designed to check if there is a significant difference between the observed number of data entries in a category of data and the expected number of data entries in the same category. The Chi-squared test is defined as:

$$\chi^2 = \sum_{j=1}^n \frac{(O_j - E_j)^2}{E_j} \quad (23)$$

where n is the number of different classes (i.e., categories) present in the data and O_j and E_j are the observed and expected number of cases, respectively, in the j^{th} class. Network *et al.* [159] develop a model that tracks the effects of child care quality on the development of young children. They develop different models, where each model is related to one aspect of the child's development; e.g., (i) cognitive competence, (ii) caregiver report of social competence, and (iii) mother report of social competence. An example output of their cognitive competence model includes standard tests such as *incomplete words*, *memory for sentences*, *letter-word identification*, and *auditory competence*. Authors calculate the χ^2 value for their models by comparing the expected values of these outcomes created by their models and the actual results acquired from the tests; they show that their model provides a good fit for the data.

2) HOSMER-LEMESHOW TEST

Hosmer-Lemeshow test [160] works for models that make a binary prediction (with two possible outcomes) and the observations are usually divided into 10 groups based on their predicted probabilities and the test is calculated as follows:

$$G_{HL}^2 = \sum_{j=1}^{10} \frac{(O_j - E_j)^2}{E_j(1 - \frac{E_j}{N_j})} \quad (24)$$

where N_j is the number of observations (for both outcomes) in the j^{th} group and O_j and E_j are the observed and expected

number of only one case (e.g the positive case) in the j^{th} group. Hansen *et al.* [161] develop a pregnancy predictor model among couples with unexplained infertility. Their models take multiple inputs such as age, income level, smoking, and duration of infertility; they predict the possibility of the pregnancy outcome. They validate their model by performing the Hosmer-Lemeshow test and show that their model is a good fit for the data.

3) KOLMOGOROV SMIRNOV TEST

The Kolmogorov-Smirnov (K-S) test [162] is another goodness of fit metric and it is based on the cumulative distribution function (CDF) of the data; it is only applied to continuous data distributions. To define the K-S test, the CDF of two data are plotted together (e.g., the CDF of the observed data vs. the CDF that is produced by the system that is modeling the observed data). The maximum absolute distance between these CDF plots is calculated and this distance is called D . The next step is to calculate the p -value (see Section VI-E) for this D value, i.e., the statistical importance of D in representing the data. For example, Chen *et al.* [163] use K-S test to determine the quality of medical images when they are compressed. In irreversible image compression techniques some of the data is lost and it is important to reach an optimal compression ratio that both reduces the size of the data and keeps the lost information at a tolerable rate. Authors sample data from both the original picture and the compressed image with size n and create the CDF for these samples. They compare these two CDFs by performing the K-S test and find the maximum distance (D) between the curves. They show that by setting a bound of $D = \frac{1.92}{\sqrt{n}}$, the amount of lost information during the compression process is acceptable.

J. CONFIDENCE INTERVAL

Confidence intervals assess the reliability of statistical estimates. *Interval estimates* are different from *point estimates* in that while only a single value is presented in the latter case (e.g., mean or median), an interval estimate is provided for a given *confidence* (or *reliability*) level (e.g., 95%) in the former case. The range of the confidence interval varies based on the desired reliability; for example, a 99% confidence interval for a value has a wider range compared to a 95% confidence interval range for the same value. The number of samples and the variance in the data affect confidence intervals as follows: low sample sizes and high data variance both result in wider confidence intervals.

An example usage of confidence intervals is presented in [164], where the authors develop a framework for predicting the future trajectory of diabetes and mental illness of patients based on their medical records. In addition to providing a precision value for their prediction, which is a point value, they also provide a confidence interval for the precision metric, at a 95% reliability.

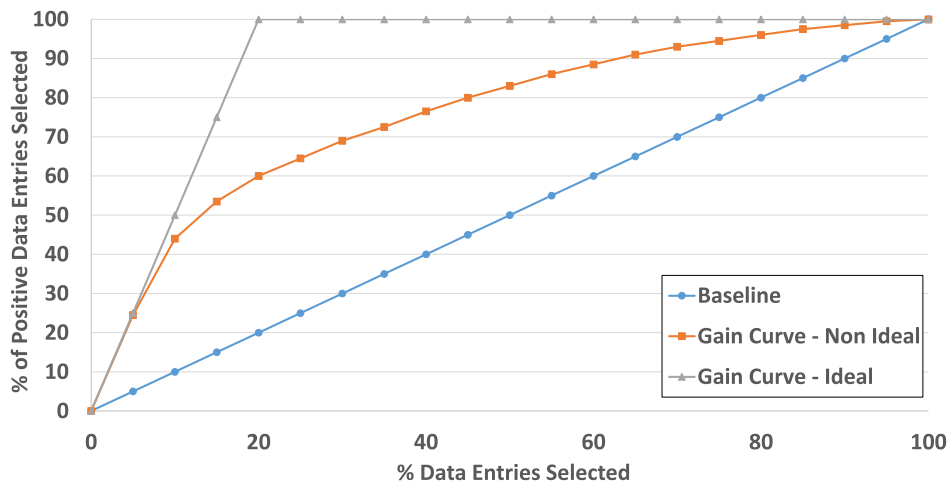


FIGURE 8. An example Gain chart for a population where 20% of the data entries are positive (e.g., unhealthy patients). The baseline selects the positive entries in proportion with the selected data entries, denoting a purely random selection. The ideal selection yields only positive data entries. A practical (i.e., non-ideal) algorithm selects a higher proportion of positive entries compared to the baseline, although it can never reach the ideal curve.

K. GAIN AND LIFT CHART

Gain and Lift charts are visualization techniques that show how good the performance of a predictive model (or algorithm) is.

1) GAIN CHART

An example Gain chart is shown in Fig. 8, which depicts the characteristic of an algorithm that is designed to select unhealthy patients from a pool of 1000 patients. Out of these 1000 patients, 200 of them are known to be unhealthy and 800 are healthy. Therefore, if the patients were selected purely randomly from 1000, we would expect to discover, say, 40% of the unhealthy patients ($200 \times 40\% = 80$) when 40% of the patients have been exhausted ($1000 \times 40\% = 400$); in other words, the ratio of the unhealthy patients discovered by this purely random selection would be 20%, which matches the original ratio of unhealthy patients ($200 \div 1000 = 20\%$). This corresponds to the “Baseline” curve in Fig. 8. Although a statistical near impossibility, in the case that every selection was correct (i.e., yielded an unhealthy patient), the best we can expect is the “Gain Curve - Ideal” in Fig. 8, which reaches the 20% ratio (i.e., 200 unhealthy patients discovered) after making a mere 200 selections. Between these two extreme cases, a realistic scenario for a selection algorithm is depicted as the “Gain Chart” in Fig. 8, which performs much better than the Baseline, although it cannot reach the ideal curve. The example in Fig. 8 selects 60% of the unhealthy patients ($200 \times 0.6 = 120$) after selecting from only 20% of the entire population ($1000 \times 0.2 = 200$).

Francis et al. [165] develop an algorithm to predict delayed discharge of the patients and readmission to a hospital following a cancer surgery. In addition to AUC, they provide a gain chart that shows their predictive validity compared to a random selection.

2) LIFT CHART

The Lift Chart, shown in Fig 9, depicts how good the algorithm performs in comparison to a purely random selection. For example, for 20% of the population, a purely random selection would discover only 20% of the unhealthy patients (Baseline), while the algorithm is able to select 60% of the patients; this means that the algorithm performed $3\times$ better than the random selection (i.e., $0.6 \div 0.2 = 3$). This is termed a lift of 3.0, which is plotted as the “Lift Curve - non-ideal” in Fig. 9. Using a similar logic, the “Baseline” of a Lift Chart has 1.0 for every point in the plot and the ideal Lift Chart has values of 5.0 until it reaches 20% of the selection values and goes down hyperbolically after that until it reaches 1.0.

Yom-Tov [166] design a classifier to predict drug recalls in the future by using internet search queries. To test their classifier, authors use a Lift chart in addition to AUC values. Their predictor is able to achieve a maximal lift of ≈ 7.5 , which implies a $7.5\times$ better prediction than a random one.

VII. COMPUTATIONAL MODELS (BOX M)

In many medical applications, it is necessary to understand and appropriately describe the structure of the phenomenon of interest before addressing the task at hand. In this section, we will provide an overview of modeling, which is a necessary procedure that enables the understanding and detailed characterization of a phenomenon, such as detection or tracking of a disease, using mathematical concepts. We emphasize that creating or using a model is, in many cases, a necessary step before using a specific machine intelligence algorithm.

A. DEFINITION

The goal of modeling is to define, understand, quantify, and visualize a phenomenon or system by generating an appropriate model that conceptually represents its structure.

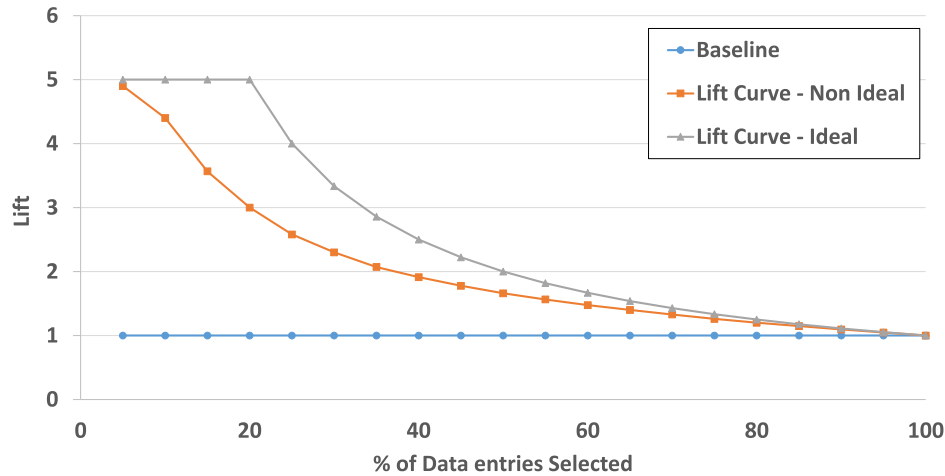


FIGURE 9. An example Lift chart, which depicts the Lift values for the data shown in Fig 8. Ideal and non-ideal lift curves are calculated by dividing the ideal and non-ideal gain values by the gain of baseline respectively.

A variety of models exist in practice (e.g., mathematical, graphical, logical) and serve different purposes. In machine learning and pattern recognition applications, mathematical and graphical models are used predominantly to describe a system using mathematical concepts and graph-based representations. Models can serve three purposes: (i) facilitate the understanding of a system and the interactions of its constituent components (e.g., an individual's disease trajectory [167]), (ii) predict the behavior of a system (e.g., predict the course of interstitial lung disease [167]), and (iii) optimize or control the behavior of a system (e.g., perform upper-limb reaching rehabilitation [21]).

1) MODEL VARIABLES

A mathematical model consists of a set of *variables* and a set of *equations* that characterize the phenomenon of interest. The variables represent quantifiable parameters of the phenomenon and can take various types of values (e.g., integer, real, boolean, or string). Typically, variables are deterministic or stochastic and can be categorized as

- 1) *state* variables, which describe the future behavior of the phenomenon,
- 2) *output* variables, which relate to the unobserved state variables and are used to make decisions regarding the state variables,
- 3) *decision* variables, which are able to control the interactions between the constituent components of a phenomenon, and
- 4) *exogenous* or *constant* variables, which are auxiliary variables that affect the structure of the phenomenon.

2) MODEL EQUATIONS

The equations, on the other hand, represent relationships that characterize any subset of the variables and are usually described by algebraic operators and functions. In general, equations can be categorized as

- 1) *state* equations that describe how the values of the state variables will change (usually with respect to time),
- 2) *observation* equations that describe the mathematical relationship among model variables,
- 3) *defining* equations that define new variables in terms of the ones that are already known, and
- 4) *constraints* that describe the phenomenon of interest through a set of conditions that must be satisfied among the variables.

For instance, Conforti *et al.* [168] address the problem of maximizing efficient delivery of services in a “weekly hospital” (i.e., a hospital that admits and discharges patients on weekly basis) by scheduling patient visits based on the hospital's available resources. Input variables in this case can be the number of available beds, the waiting list of patients, and the capacity of the clinical services. At the same time, decision variables may involve if a patient should occupy a bed or undergo a clinical service at a given time slot. Different type of constraints can be imposed in this case, such as a limit on the number of visits by a patient (e.g., one visit per week) or the total number of patients that can be admitted in a specific week (e.g., maximum 100 patients per week).

In the rest of this section, we delve into the details of modeling as follows. In Section VII-B, we classify models based on their structure and provide relevant medical applications. In Section VII-C, we discuss the challenges in building and selecting appropriate models and related methods. In Section VII-D, we summarize important model evaluation strategies. Finally, we discuss the important topic of learning model parameters in Section VII-E. Widely used models in the medical domain are studied in Section VIII.

B. CATEGORIES OF MODELS

Models can be classified in one or more categories based on their structure (i.e., form of variables and equations, model assumptions and goals). For instance, a linear stochastic

dynamical model is an example of an *explicit, linear, probabilistic, continuous* and *dynamic* model that has been used in various medical applications (e.g., heart rate estimation [169], respiratory tumour motion prediction [170]). Next, we briefly discuss the different model classification criteria and provide related examples within the medical domain.

1) EXPLICIT VS. IMPLICIT

A model is characterized as *explicit* if all input parameters are known and can be used to determine the output parameters through a finite series of computations. For example, Wang *et al.* [171] propose an explicit dependency model to appropriately fuse medical images in an effort to capture the extant dependencies between different highpass subbands. They present experimental results based on actual MRI images from The Whole Brain Web site of the Harvard Medical School and show that their proposed approach achieves better fusion performance compared to other typical fusion methods (e.g., PCA, discrete Wavelet transform fusion). In contrast, a model is *implicit* if only the output variables are known and are used to determine the corresponding inputs. For instance, Alagoz *et al.* [172] study the problem of accepting or declining an offered organ of a given quality from patients who are in a waiting list and have end-stage liver disease. They propose an implicit model that relates the patient health, the organ quality, and the effects of the waiting list to estimate the probability of the patient accepting or declining the transplant.

2) RULE BASED

In rule-based models, a set of rules are used to indirectly specify a mathematical model. As opposed to other modeling approaches, where all possible interactions must be specified ahead of time and require careful revision of the model in the case of even minor changes, in rule-based models, a characteristic of the phenomenon or system of interest may be introduced or modified by simply adding or updating a rule that relates to this particular characteristic. Rules are usually defined and visualized through graphs and graph rewriting approaches [173]–[176], where a new graph is created based on an original graph in an effort to describe the interactions between variables of the model. Rules are usually employed to define interactions between various variables and the associated consequences. Alternatively, a rule based model can be transformed into an equation based model such as Markov chains or differential equations. In general, rule based modeling approaches are adopted when it is much simpler to use a set of rules than identifying an appropriate equation based mathematical model. For instance, in [177], rule based models are discussed and used within the context of protein-protein interactions and other biochemical systems, where the types of interactions between chemical components are described by a set of rules.

3) LINEAR VS. NONLINEAR

A model is characterized as *linear* if the equations describing the system or phenomenon of interest are linear with respect to model variables. As an example, Smith and West [178] present a linear growth model for the evolution of the serum creatinine chemical indicator in an effort to monitor the progress of the kidney function of individual patients who recently received transplants. A model is referred to as *non-linear* in any other case. In [179], a nonlinear model that considers the spatial correlations between neighboring pixels is proposed for segmenting brain magnetic resonance images. At this point, we note that the linearity/nonlinearity entirely depends on the context, in the sense that many linear models may include nonlinear expressions. For example, in a linear model, the state and output variables are described by linear equations, but certain parameters can be characterized by nonlinear defining equations. In general, most nonlinear models are difficult to study due to the complexity of the associated equations. A common approach that facilitates the understanding of the behavior of nonlinear models and their analysis is *linearization*, where nonlinear equations are replaced by their linear approximation at given points.

4) DETERMINISTIC VS. PROBABILISTIC (STOCHASTIC)

A model is referred to as *deterministic* if the values of all variables are calculated by using the model variables and no random effects exist. In that sense, a deterministic model will always generate the same values for a given set of initial conditions. In contrast, in a *probabilistic* model, the values of model variables are described by probability distributions and as a result, such a model generates different values for a given set of initial conditions. For instance, in [66], a probabilistic model based on Gaussian distributions is proposed to model a variety of features extracted from accelerometer and heart-rate signals to perform physical activity detection.

5) DYNAMIC VS. STATIC

In a *dynamic* model, variables evolve in time and capture the changes of the phenomenon of interest as time progresses. In general, dynamic models are described by difference and differential equations as well as probabilistic update rules that are compactly represented by transition probability matrices. In [27], physical activity is modeled as a finite state Markov chain that evolves in time and is observed via accelerometer and heart rate data. On the other hand, in a *static* model, variables are not subject to change over time and thus, such a model is suitable for the description of time invariant phenomena or systems. For instance, in [180], a static nonlinear finite element model is proposed for accurate brain deformation prediction due to individual anatomical structure differences. Accurate knowledge of brain deformation is necessary when determining the current position of a tumor and other pathologies during surgery.

6) DISCRETE VS. CONTINUOUS

A model is characterized as *discrete* if model variables take on discrete values (finite or countable). For instance, in [35], both the actual physical activity and the related noisy observations are represented as categorical random variables that take values such as sitting, standing, running and walking. Contrary to a discrete model, in a *continuous* model, variables take continuous values and thus can model phenomena or systems that are described by such parameters. In [169], heart rate estimation is modeled as a Kalman filtering task, where the heart rate and features extracted from an ECG and arterial blood pressure signals are represented by a continuous valued linear stochastic dynamic model. A model that includes both discrete and continuous valued variables is referred to as a *mixed* or *hybrid* model. For instance, in [27], physical activity is modeled as a discrete random variable, whereas features extracted from accelerometer and heart rate signals are modeled as continuous random variables with Gaussian distributions. Furthermore, models can be characterized as *discrete time* or *continuous time* if the related time variable takes discrete or continuous values, respectively. For example, Zois et al. [35] assume that physical activity changes take place in discrete time points and as a result, they adopt a discrete time model for physical activity evolution.

7) DECISION MAKING MODELS

Sequential Decision Making (SDM) mechanisms (also referred to as *agent decision processes*) provide a mathematical framework for modeling decision making in cases where outcomes are partially random and partially under the control of a decision maker. They are useful for a wide range of applications [181] including but not limited to machine maintenance, structural inspection, autonomous robots, marketing, target tracking, education, estimation of sparse signals, and communications. Markov Decision Processes (MDPs) [182]–[184], Partially Observable Markov Decision Processes (POMDPs) [183], [185] and Multi Armed Bandits (MABs) [186], [187] are typical examples of such a framework that can successfully capture the complex sequential decision making nature of medical diagnosis and treatment. The common characteristic among these models is the existence of a mechanism that makes decisions at specific steps. In all of these cases, the goal is to find the optimal decision strategy that optimizes (maximize or minimize) a certain objective. Such mathematical models are very useful in modeling and solving complex, stochastic, and dynamic problems. Since medical decision making is inherently complex and uncertain with respect to the treatment outcomes and costs associated with different diagnostic tests, such models are expressive enough to capture the interaction among all relevant features. As a result, various medical applications have used these mechanisms to enable automatic decision making and more efficient usage of the related resources. Examples include physical activity tracking for obesity

prevention [35], [188], stroke rehabilitation [21], assisting people with dementia [21] and stress management interventions [189]. For more information regarding SDM models and techniques, the interested reader is referred to [182]–[187] and [190].

There are various challenges associated with the application of decision making models in medical cyber physical systems. First, deciding on the detailed structure of the state is a very challenging task. Common issues are the size of the state space (i.e., a fine grained state space yields better decision strategies but leads to more complicated models) and data limitations (e.g., for some state control pairs, there may be no observations). Second, when trying to optimize a treatment or intervention plan, a model of the individual's health evolution (before and after the treatment or intervention) is necessary. In fact, to support proactive and preventive healthcare delivery, time variable models are required to capture the progression of diseases and the effects of treatments and interventions. Solving such stochastic models is a computationally intensive task, but exploiting the structure and characteristics of the particular applications can lead to efficient solution strategies (e.g., [35], [191]). Third, selecting between the available decision making models and determining the rewards/costs associated with various control actions may require the doctor's intervention. Fourth, the inherent variability of individuals with respect to the response to treatments/interventions suggests the need for personalized models and methods. Fifth, considering the healthcare medical cyber physical systems' scale with millions of EHRs, biometrics signals, treatments, interventions and patient-doctor interactions, scalable decision making models and approaches must be devised. Finally, in a free living setting, non-compliance of individuals and missing data will be prevalent requiring the design of appropriate decision making strategies.

C. MODEL SELECTION

Model selection refers to the process of selecting a model from a set of *candidate models* based on either a description of the phenomenon of interest or a pre-existing dataset. In general, the set of candidate models is selected based on the characteristics of the problem at hand, focusing initially on simple models. Building a mathematical model usually requires some sort of an abstraction of the problem and is achieved by having appropriate *assumptions*; a model can accurately predict the behavior of a phenomenon if it is built upon valid assumptions. Selecting an appropriate model is a difficult problem since it involves a trade-off between simplicity and accuracy. Among different models with similar predictable accuracies, the simplest model is usually selected. The quality of a model depends on how well the model agrees with the observations made during empirical studies. To create realistic mathematical models, subjective information such as expert input, intuition, and mathematical convenience are usually considered.

1) A PRIORI INFORMATION

The amount of available *a priori information* on the problem of interest usually determines both the accuracy of the model and the approach that can be used. In the latter case, models can be categorized as (i) *black box* if no a priori information is available, (ii) *white box* in the case all required information is known, and (iii) *grey box*, where a model is constructed based on insight, a priori information, or experimental data and a set of unknown parameters need to be estimated. Grey box modeling is adopted in practice, since we have access to various types of a priori information for most problems of interest. Exploiting as much a priori information as possible leads to more accurate models.

Usually, a priori information can be given in the form of equations that relate problem variables, assuming that only certain parameters related to the variables are unknown and need to be estimated. For instance, in an ECG recording (see Fig. 6), the ST elevation is directly related to a potential heart condition and must be estimated [38]. In other cases, it is necessary to estimate the functional form of equations. In the same figure, it is a well-known fact in the cardiology field that the morphology of the T wave has a direct correlation to an upcoming cardiac hazard, however, a functional form of an equation is not easy to determine [192].

In cases where there is no access to a priori information one can use very general functions. Artificial neural networks—discussed in Section XI—are often used, since no assumptions are imposed on the incoming data. Next, we briefly discuss a number of topics that relate to a priori information:

- **Subjective Information:** Intuition, experience, domain knowledge, or even mathematical convenience can be used to incorporate subjective information into a model; as an example, we may use an earlier probability distribution to describe the behavior of a variable and then update its distribution based on experimental data. Furthermore, domain knowledge, which refers to knowledge that is related to the particular problem of interest, can not only lead to more accurate models, but also simplify the modeling process; one example of this from the field of cardiology is that while an ECG recording (Fig. 6) has a rich set of features such as QT, ST, PR, TQ, etc. [38], [47], the part of the ECG that contains the most useful information is generally the QT and RR segments [10].
- **Data Analysis:** In a variety of problems, we have access to empirical data without an explicit knowledge of the physical behavior of the phenomenon of interest. In this case, our first step is to identify a set of input, output, and internal variables that can successfully describe the phenomenon. Data analysis methods including, but not limited to, histograms and scatter plots as well as domain knowledge, intuition, and experience enable us to extract useful information that can be used to build robust and accurate models later. At the same time, feature extraction techniques, as discussed in Section V, can be used

to select which variables will go into a model. On the other hand, feature selection techniques can provide us with a way to build simple but accurate models.

- **Relationship Estimation:** During the data analysis stage, it is usually the case that important variables are selected based on their ability to accurately describe the phenomenon or system of interest. However, since models are designed to predict or optimize the behavior of a system, a necessary step is to start from the measurements of the system behavior and identify useful relations between model variables. This process is termed *system identification*, *structure learning*, or *data-driven modeling* depending on the methodology used and its focus, the form of the final outcome, and the field from which an approach was originated. In *system identification* [193]–[196], statistical methods are used to build mathematical models of dynamic systems based on observed input/output data. Based on the model structure (i.e., linear versus nonlinear), various approaches have been proposed including, but not limited to, least squares, maximum likelihood, Volterra series and NARMAX methods. In [197], a new algorithm named the Common Model Structure Selection (CMSS) is proposed, which is used to select a common model structure. As a case study, the authors develop a time-varying common structure for EEG signals that is able to follow the test data with a 0.27% mean square error. In the context of probabilistic graphical models [198], it is usually the case that the structure of the Bayesian network (i.e., the relationships between the variables in the network) that we wish to use to describe our problem of interest is not known in advance. In this case, there are three main approaches: (i) *constraint-based structure learning*, where the goal is to find a network that best explains these dependencies and interdependencies in the data, (ii) *score-based structure learning*, where the goal is to find the highest scoring network structure that explains the data assuming an appropriately defined scoring function, and (iii) *Bayesian model averaging*, where an ensemble of possible structures is generated and all such predictions are averaged. Structure learning problems are often formulated as *multiple hypothesis testing problems*, where each hypothesis corresponds to a different structure, or as an optimization problem, where the optimal solution will correspond to the highest scoring structure. For example, Kontis *et al.* [199] study the life expectancy in 35 industrialized countries. They deploy a probabilistic Bayesian Model Averaging (BMA) approach with an ensemble of 21 models that project age-specific death rates. They train all of these different models and pool them together to get age-specific death rates under the BMA and show that the performance of BMA projection is better than any of the single models. They predict that the life expectancy will increase in all the 35 countries for at least 65% of the women and 85% of the men.

- **Training:** As already discussed, all types of models (excluding white box models) include parameters and relations that need to be estimated and optimized based on available data in order to accurately describe the system or phenomenon of interest. This process is referred to as *training* and will be discussed in more detail in Sections VII-E and IX-B. Since the quality of training depends on the quality of the available data, *active learning* methods [200] have been employed to efficiently collect informative data that can lead to low-complexity but accurate models. The main idea is to interactively query an expert (i.e., user or any other information source) to obtain informative input/output data points. Kholghi et al. [201] develop an active learning based system that extract medical concepts from clinical reports. The language of clinical reports are usually unstructured and may not follow standards which makes annotating them a costly task, as they have to be annotated in a supervised fashion; this motivates the authors to analyze the effectiveness of active learning techniques on reducing annotation efforts of medical records by developing a model on a portion of the data in an active learning framework and analyzing the rest of the data based on the developed model. Authors investigate the trade-off between the accuracy of the model output and the effort that it saves in the annotation process. They are able to show that by using active learning, they can get within an acceptable range of accuracy with much less effort, although the accuracy is not as high as in supervised annotation.

Model parameters can be either estimated from the data through training or considered fixed. In the latter case, they are referred to as *hyper-parameters* and need to be optimized to ensure the accuracy and generalization of the model, a process known as *hyper-parameter optimization* or *tuning*. For example, in [202], a model is proposed that analyzes the electronic health records of patients and makes a prediction on their probability of heart failure. This model includes many hyper-parameters such as the dimension of their input feature data, regularization factors that bound the parameters of the model, and the complexity of their model. They discuss their hyper-parameter tuning phase and describe their selection process. They show that their model is more accurate on heart failure prediction compared to other methods, especially when the models are trained on smaller portions of the data. For example, when the models are trained on 20% of the data, the proposed model achieves a 32% accuracy, while other models have accuracies lower than 28%.

Typically, hyper-parameters are selected to maximize a carefully selected metric defined in terms of modeling accuracy, while model generalization is verified via *cross-validation*. Some optimization methods employed for this purpose are: (i) *Grid search*, which exhaustively searches over the hyper-parameter space and is guided

by some performance metric; (ii) *Random search*, which randomly samples the hyper-parameter space; (iii) *Bayesian optimization*, which iteratively selects hyper-parameters in a way that balances exploration (i.e., consider values that have not been explored before) with exploitation (i.e., consider values that have shown to yield good results) to obtain optimal results relatively quickly; and (iv) *Gradient-based optimization*, which involves the optimization of hyper-parameters through gradient descent.

2) METHODS

As discussed earlier, model selection is guided by the amount of a priori information available (i.e., description of the phenomenon of interest, access to data, domain knowledge), while the trade-off between realism and complexity usually dictates the selection process. Even though it appears that this process is based on intuition and experience rather than a deterministic set of steps, there are methods that can not only facilitate but improve this process. Some examples of such methods are:

- **Optimal Design of Experiments:** When we wish to decide among different models, we can design a set of experiments that can help us identify the model that best describes our data. In this case, we first need to specify a model for the design and a carefully selected statistical criterion; then, optimal design methods can be used to reach an optimal design [203]. Within this context, Bayesian experimental design [204] provides a general theoretical framework that uses probability measures to account for both prior knowledge and observations collected during the experiment. The goal is to design experiments that maximize the expected utility of experimental outcomes defined usually in terms of the accuracy of collected information and the cost of experiments. Since experimentation is an iterative process, optimizing the design of sequential experiments [205]–[207] has also been studied in the literature starting from the pioneering work of Wald [205]. We emphasize that the “optimal design of experiments” framework enables us to carry out optimal tests between specified models, where we can exploit multiple hypothesis testing approaches (see Section X-K) to test and decide amongst alternative hypotheses. Optimal experimental design has also been used to efficiently generate data for both model fitting and reduction [208], [209]. In general, non-optimal designs require a larger number of experiments to achieve the same accuracy as the optimal ones.

Experimental optimization is used in [210], where authors develop a human seated postural control system for patients with postural control deficit. The main idea in their study is that human subjects tend to fatigue quickly in motor control tests, which distorts the input data; hence, there is a need to design experiments optimally to get the best input data in the shortest possible

time to avoid biased data due to fatigue. This leads the authors to experimental optimization of their model that minimizes the variance of the parameters gathered in experiments. They show that their developed model using the optimized data is more stable (has less variance) in helping subjects in controlling their seated posture compared to models using non-optimized data.

- **Regression Analysis:** As already discussed, a mathematical model consists of a set of equations that expresses the relationship among model variables. In many real problems, the relationship among model variables is usually unknown and needs to be estimated. To this end, we can employ regression analysis [211], [212] to quantify the effect of independent variables on dependent variables to determine the appropriate model for our problem of interest. Regression analysis can also be used to infer causal relationships, but caution needs to be taken to avoid misinterpretations [213], [214]. In all cases, the goal is to estimate the *regression function*, which is a function of an independent variable, i.e.,

$$\mathbf{Y} \approx f(\mathbf{X}, \beta), \quad (25)$$

where β denotes the unknown scalar or vector parameters and \mathbf{X} and \mathbf{Y} represent the independent and dependent variables, respectively. Note that determining the distribution of the variance of values of the dependent variables, as computed through the regression function, is also of equal importance, since it is directly related to the resulting estimation error.

Techniques for regression analysis can be categorized as either (i) *parametric* (e.g., linear regression [215], [216], ordinary least squares regression, nonlinear regression [217]), where the regression function is defined in terms of a finite number of unknown parameters, or (ii) *non-parametric* (e.g., Gaussian process regression (Kriging) [218], kernel regression [219], [220], regression trees [221], [222]), where the regression function belongs to a specified set of functions. In both cases, the parameters or functions need to be estimated from the data. The performance of such methods typically depends on how well the regression approach employed matches the process that governs the generation of data. In most cases, this process is unknown and thus, assumptions need to be made, which can be tested assuming we have access to a sufficient quantity of data. An example application of a parametric regression analysis is presented in [223], where authors study the relationship between the body mass index (BMI) and physical activity in children and adolescents. By using regression analysis, they are able to show that spending more time on moderate-to-vigorous physical activity is associated with lower BMI. A non-parametric regression application is shown in [224], where a system is developed to classify lung nodules (an abnormal swelling of cells in the body) as benign or malignant. By deploying

kernel regression models, authors are able to develop a framework that has a 85% accuracy in clustering nodules into their respective groups (benign and malignant), beating other methods by $\approx 5\%$.

- **Log-linear Analysis:** In a similar vein, log-linear analysis [225] is used to determine if there is a statistically significant relationship among three or more discrete-valued variables [226], while accounting for the variance in the available data. In this sense, the goal is to determine which model components need to be retained to ensure an accurate representation of the data. To this end, the likelihood ratio statistic of the form shown below is calculated:

$$X^2 = 2 \sum O_{ij} \ln \frac{O_{ij}}{E_{ij}}, \quad (26)$$

where O_{ij} and E_{ij} represent observed and expected frequencies, respectively. If the resulting value is larger than a critical value, we conclude that there exists a statistically significant relationship among the variables of interest.

As an example application, Christin *et al.* [227] study the relationship between living with a chronic condition in adolescence, the quality of interactions between adolescent and their parents, and the adolescent's psychological development by using log-linear analysis. Their model shows that having a chronic disease is not connected directly to poor parent-adolescent relations, but they are connected indirectly by two variables — sensation seeking and suicide attempt— in adolescent psychological health.

3) CRITERIA

To facilitate the process of model selection, various criteria have been proposed, which we summarize next:

- **Akaike Information Criterion (AIC):** It corresponds to a metric that compares the quality of two models with respect to the same dataset by balancing *goodness-of-fit* (see Section VI-I) with model complexity (i.e., number of variables present) [228]. In this context, we consider a model with k parameters and define the AIC as:

$$\text{AIC} = 2k - 2 \ln p(y|\hat{\theta}), \quad (27)$$

where $p(y|\hat{\theta})$ denotes the conditional probability of the observations (y) given the maximum likelihood estimate ($\hat{\theta}$) of the parameters. We emphasize that the AIC provides a means of comparison of different models and cannot provide a quality measure of a single model in an absolute sense.

As a use case of AIC, Vigen *et al.* [229] compare the information from self-reports and electronic health records for four common comorbidities including diabetes, hypertension, myocardial infarction, and other heart diseases in women diagnosed with breast cancer. As part of their study, they show the Cox hazard ratio (Section VIII-I) for these four comorbidities, based on

two models: one with the model based on the self-reported data and one based on medical records. They use AIC as a metric to show which one of these two models works better for predicting the hazard ratio. The results show that for the hypertension and other heart diseases, the latter works better; for diabetes and myocardial infarction, the former model exhibits a better performance.

- **Bayesian Information Criterion (BIC):** This criterion is closely related to AIC, however it places more emphasis on penalizing the number of model parameters [230]. In particular, the BIC can be calculated as follows:

$$\text{BIC} = k \cdot \ln(n) - 2 \ln p(y|\hat{\theta}), \quad (28)$$

where n denotes the sample size.

As an example use case, Ide *et al.* [231] study the difference between effectiveness of treatments of hepatitis C in different regions of Japan. They develop different models that take various types of inputs and report BIC to compare these models. They show that taking the *region* as a variable input to the model, as opposed to a fixed input, yields a lower BIC, which means that regional differences may exist in treating the hepatitis C virus infection in Japan. Another example usage of BIC is presented in [232], where the authors detect patients with recent transmission of tuberculosis (TB) for the purpose of preventing future spread of the disease. They show the input data features and discuss how some data inputs are selected specifically for their case; for example, since their study is conducted in Montreal, Canada, they take *Haitian born* as an input, as they know it has an impact on their subject of study. They discuss their algorithm selection process; they choose a logistic regression algorithm (Section X-H) to predict whether a given TB case was involved in a recent transmission. They analyze the interaction of their input features by comparing the base model's BIC with BIC of a model that has an interaction of two separate input features. They determine that combining two features of "living in an apartment" and "cavitary lesion on chest X-Ray" actually reduces the BIC and consequently yields a better model for this problem.

- **Deviance Information Criterion (DIC):** This criterion constitutes a generalization of both the AIC and BIC [233] and is formulated in Eq. (29), as shown at the top of the next page where n denotes the sample size and θ_i is the parameter vector for the i^{th} sample. An example usage of DIC is presented in [234], where the authors develop statistical models for polio outbreaks in different countries. From these models, they choose the best model by using DIC as the selection metric. They show that their models are able to identify polio outbreaks in different countries with specificities mostly higher than 90%.
- **Focused Information Criterion (FIC):** Similar to the previous criteria, the FIC selects the most appropriate

model among a set of available models for a given dataset. However, it does not assess the overall fit of the candidate models; instead, it directly focuses on the parameter of interest [235]. In this sense, we first need to determine exact or approximate expressions that quantify each estimator's quality and evaluate them for the given dataset to select the model with the best estimated quality. *FIC plots*, which display estimates along with their FIC scores, are also often used to provide an informative picture of all estimates across models. Since the FIC focuses on a particular parameter of interest, the form of the resulting expressions relates to the problem of interest. FIC is used in [236] to distinguish the best predictive models in personalized medicine. Authors study prostate cancer patients and design a predictive model on whether the cancer cells have spread away from the prostate or not, based on some input features related to the tumor and the person. They use FIC and AIC to select the best models and show that the model selected by FIC is more accurate compared to the one selected by AIC.

- **Bayes factor:** The goal of the Bayes factor is to decide between two statistical models based on the probability of a model M given data D determined by the Bayes rule:

$$\Pr(M|D) = \frac{\Pr(D|M) \Pr(M)}{\Pr(D)}, \quad (30)$$

where $\Pr(D|M)$ denotes the probability that the set of data D is observed under model M . In particular, the Bayes factor K is defined in terms of the ratio of the probabilities of observing the data D under the two assumed models:

$$K = \frac{\Pr(D|M_1)}{\Pr(D|M_2)}, \quad (31)$$

where a value greater than 1.0 suggests that model M_1 is more appropriate than model M_2 to describe the data D . More elaborate scales have been proposed (e.g., [237]), where the different values of K can precisely quantify the support of one model versus the other.

An example use of Bayes factor can be found in [238], where the authors study the association between the course of depressive symptoms in older adults and the risk of dementia for them. They develop multiple models with linear, quadratic, and cubic terms and compare them against each other by using the Bayes factor. Their best model shows that older adults with high depressive symptoms show a higher probability of developing dementia.

- **Cross Validation:** This technique assesses the quality of a model by partitioning the available data into *training* and *testing* subsets, performing model construction on the former and validating model fit on the latter. In practice, multiple rounds of cross validation are performed using different partitions and the results are averaged in order to reduce variability. Cross validation is heavily

$$\text{DIC} = -2 \left(\log p(y|\hat{\theta}) - 2 \left(\log p(y|\hat{\theta}) - \frac{1}{n} \sum_{i=1}^n \log p(y|\theta_i) \right) \right), \quad (29)$$

used by the algorithms in Section X to validate the accuracy of their underlying models.

- **False Discovery Rate (FDR):** This criterion uses the expected value of false prediction percentage:

$$\text{FDR} = \mathbf{E}[Q] \quad (32)$$

to quantify the rate of *type I errors* (i.e., the incorrect detection of an effect that is not present), where Q is the proportion of false predictions made by a model (see also Section VI). Ideally, we would like to select models that have FDR values below a threshold q .

Nelson *et al.* [239] study coronary artery disease and the genome locations (loci) that affect it by using FDR as a metric. They extend on previous studies on the same topic and by setting a 5% threshold for FDR, they show that they are able to detect a new list of genes that are associated with coronary artery disease risk.

- **(Log)-Likelihood Ratio Test:** This method constitutes a statistical test that compares the goodness-of-fit of two statistical models (i.e., which model is more likely to have generated the available data) by computing an appropriate ratio that captures the likelihood of the data under the two models. This ratio is usually compared against a threshold that enables us to decide in favor of one model versus the other. The reader is referred to Section X-K for a more detail discussion of the likelihood ratio test.
- **Mallow's C_p :** This criterion is used to assess the appropriateness of a regression model by computing the following index [240]:

$$C_p = \frac{\text{SSE}_p}{s^2} - (N - 2p), \quad (33)$$

where p is the number of parameters in the model, s^2 denotes the mean squared error for the full model, SSE_p is the residual sum of squares for the subset model, and N is the dataset size. The goal is to identify the best model that includes a subset of parameters from the initial model. Smaller values of C_p suggest that the model is considerably accurate.

Luzak *et al.* [241] study the factors that influence lung function in adolescents. They take early life events and current environmental/lifestyle elements as input and study their effect on allergic diseases in 15 year-olds. Multiple models are developed and the best one is selected based on Mallow's C_p metric. The adopted model shows that factors such as early lung infections and indoor second-hand smoke exposure are major factors in allergic disease on lung function.

- **Minimum Description Length (MDL):** This principle is based on the idea of selecting the model that is

able to accurately represent the original dataset using a shorter description [242] in an effort to balance fit with complexity. It is based on information theory principles (particularly the idea of *compression*), which exploits regularities in data to describe them using fewer symbols. The description length of a model is defined as:

$$\mathcal{D}(Y; M_{\hat{\theta}(Y)}) = \ell(Y; M_{\hat{\theta}(Y)}) - \log_2 L(Y|M_{\hat{\theta}(Y)}), \quad (34)$$

where M_θ denotes a parametric model indexed by a parameter vector θ ; the $L(\cdot|M_\theta)$ is the likelihood function, Y represents the dataset, and $\hat{\theta}(Y)$ is the maximum likelihood estimate of the parameter vector θ . The first part of Eq. (34) captures the model complexity by including the number of parameters in the model. On the other hand, the second part of Eq. (34) represents the fit of the model to the data. The MDL principle selects the model with the minimum description length [243].

van Sloun *et al.* [244] propose a dynamic linear system approach to detect and localize prostate cancer from ultrasound images. Since they have access to a limited set of observations, they apply the MDL principle to select the model that best fits the available observations while taking into account the complexity of the model.

- **Minimum Message Length (MML):** The main idea behind this criterion is to select the least complicated model even if the models from which we need to choose exhibit different goodness-of-fit [245]. It constitutes a Bayesian model comparison method, since it computes and assigns a score to each model based on the following expression:

$$-\log_2(P(M, Y)) = -\log_2(P(M)) - \log_2(P(Y|M)), \quad (35)$$

where $-\log_2(P(M))$ represents the number of information bits needed to represent the model M , and $-\log_2(P(Y|M))$ is the number of information bits used to represent the data Y assuming they are encoded via the model M (e.g., parameters, initial conditions). The term $-\log_2(P(M, Y))$ denotes the complexity of model M for dataset Y in terms of the number of information bits needed for representation. According to the MML criterion, the model that generates the smallest complexity (in terms of the information bits needed for representation) is more likely to have generated the data. Given its form, this criterion can be used to compare models of different structure.

Ameli *et al.* [246] develop a system to detect fatigue in patients who are undergoing chemotherapy. They gather kinematic data from 23 body segments during a six-minute walk test and cluster the data into different

subcategories to find the physical performance status of the patients which leads to classification of the patients' fatigue. They fit multiple clusters to different portion of the data and use MML to select the best models from those. They demonstrate that their clusters show high relation with the patients tiredness.

- **Structural Risk Minimization (SRM):** This method attempts to balance model complexity with the model's fitting ability on the training data [247]. To this end, it uses the *VC dimension* [248], which captures the power of a model. To represent *complexity* and *empirical error* to capture the fitting quality in the training data, a more powerful model is more complex but might lead to overfitting, while a less powerful one has limited modeling capabilities but is less prone to overfitting. The SRM process consists of the following four steps:

- (1) Select candidate models for the given dataset, guided by a priori domain knowledge.
- (2) Order models in terms of increasing complexity.
- (3) Determine each model's parameters that yield the minimum empirical risk defined as:

$$R_{\text{empirical}} \triangleq \frac{1}{N} \sum_{i=1}^N L(y_i, \delta(\mathbf{x}_i)), \quad (36)$$

where $L(\cdot)$ is an appropriately defined loss function (e.g., 0–1, squared error), $\{\mathbf{x}_i\}_{i=1}^N$ is the number of points in the training dataset, $\{y_i\}_{i=1}^N$ are the true but unknown labels, and $\delta(\cdot)$ is our prediction rule.

- (4) Select the model that yields the minimum sum of VC dimension and empirical risk.

In [249], a support vector machine (SVM)-based regression approach is devised to model the nonlinear relationship between cardiovascular response and exercise. The proposed formulation is essentially based on the SRM principle and balances the quality of the approximation with the complexity of the approximating function. The authors are able to show that the relationship among various cardiovascular variables during steady-state incremental exercise is nonlinear in nature, contrary to prior work, while explicitly demonstrating the structure of the nonlinearity.

- **Stepwise regression:** The goal of this method is to identify the minimum number of relevant predictive variables in a regression model automatically. To this end, variables are added or subtracted from the set of relevant predictors by evaluating a specific criterion at each step. In practice, stepwise regression can be accomplished by: (i) *forward selection*, where predictors are sequentially added to the set at each step by selecting the variable that gives rise to the most significant statistical improvement of the model fit until improvement saturates; (ii) *backward elimination*, where predictors are sequentially subtracted from the set at each step by selecting the variables that yields the least significant statistical loss; and (iii) *bidirectional elimination*, which combines the

previous two approaches and simultaneously determines which variables should be added and which should be eliminated.

Grandner and Winkelmann [250] study Nocturnal Leg Cramps (NLC) and use forward stepwise regression to find the most important factors that affect it. This technique shows that factors such as age, unemployment, shorter sleep duration, higher BMI, and smoking play a more significant role in higher NLC frequencies.

Even though we discussed a large number of criteria for model selection, note that the most commonly used criteria are the AIC, BIC, and the Bayes factor.

D. MODEL EVALUATION

One of the most important steps of the modeling process is *model evaluation*, i.e., evaluating whether a model can accurately describe the phenomenon or system of interest. To this end, we first need to evaluate if the model we selected is consistent with the available empirical data. Obviously, if this is not the case, we must either appropriately modify the model or identify another model that is consistent with our data. As already discussed, cross-validation can be employed to validate the accuracy of a selected model and is heavily used by the algorithms in Section X. Among the various performance metrics discussed in Section VI, the following metrics are also used for model evaluation: confidence interval, confusion matrix, gain & lift chart, Kolmogorov-Smirnov chart, χ^2 , ROC curve, Gini coefficient, RMSE, and L_1 version of RMSE. An accurate model will be able to match the testing data, even though the parameter values of the model were learned from the training data. Evaluating how closely the predicted data matches the actual data requires defining appropriate *metrics*, which can either evaluate the appropriateness of model parameters or quantify the validity of the general mathematical form of a model. In the former case, *loss* (or *utility*) functions are used to penalize (or reward) mismatch (match) between actual and observed data. The latter case, however, is not so straightforward and may require the use of non-parametric statistics approaches [251], [252] to evaluate the validity of the mathematical form of the model (e.g., the ability of a probability distribution to accurately describe the data). Other factors need to be also considered while evaluating a model, such as consistency with unknown data, generalizability, complexity, degree of confidence, and cost. In general, a good model should be able to balance accuracy with all or subset of the above factors.

E. LEARNING MODEL PARAMETERS (BOX T)

As already discussed earlier, a model typically includes a set of parameters that need to be learned from the available data. This problem is referred to as *parameter estimation* and various approaches have been proposed in the literature to address it. Next, we briefly go over the methods that have been mostly used in the healthcare domain and provide relevant examples. For a more detailed exposition on the problem of parameter estimation, the interested reader is referred to [253]–[257].

1) LEAST SQUARES ESTIMATION

The main idea behind *least squares estimation* is to find the unknown values of a set of parameters by minimizing the sum of the squared deviations between the observed and the estimated output of a model. In mathematical terms, the sum of squared deviations is expressed as follows:

$$\sum_{i=1}^n (y_i - f(\hat{\theta}))^2, \quad (37)$$

where $\{y_i\}$ represent the observed output data sequence, $\hat{\theta} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_K\}$ are the unknown parameters and $f(\cdot)$ represents the model type (e.g., linear, nonlinear). Depending on the structure of the model, least squares minimization can be either achieved analytically (by calculus for linear models) or through an iterative numerical procedure (in the case of nonlinear models).

Least squares estimation is used in [258], where authors investigate the effects that informal care of older patients by adult children have on formal care (done by a medical professional) and the cost of healthcare. Authors model utilization of formal care as a function of inputs such as informal care, health, demographics, income, and other measures such as insurance. They use least squares estimation to find the parameters that best fit the formal care utilization function to real world data. Their model shows that informal care done by children can substitute not only nursing home and long-term home health care, but it can also reduce the number of hospital care and physician visits.

2) MAXIMUM LIKELIHOOD ESTIMATION

A basic measure of the quality of estimated parameters with respect to observed data is the *likelihood*, which corresponds to the joint probability $p(\mathbf{y}|\theta)$ of a set of observations \mathbf{y} conditioned on the unknown parameters θ . As a result, a natural approach to parameter estimation is to select the set of parameter values that maximize the likelihood of the observed data. This estimation approach is referred to as *maximum likelihood estimation* and is formally defined as follows:

$$\hat{\theta} = \arg \max_{\theta} p(\mathbf{y}|\theta). \quad (38)$$

Similar to the least squares estimation, maximum likelihood estimation can be achieved analytically by computing the derivatives of the likelihood function with respect to all unknown parameters and simultaneously solving the resulting equations. This in general generates intuitive results. For instance, in the case of multinomial distributions, the maximum likelihood estimate coincides with the relative frequency estimate.

Poletto *et al.* [259] develop a transmission model that models the spread of Middle East respiratory syndrome coronavirus (MERS-CoV) in different countries and use maximum likelihood analysis to select the best parameters in the model. They are able to show that outbreaks of MERS-CoV can not have a self-sustained epidemic and most cases of the

epidemic are sporadic cases of zoonotic or environmental transmissions.

Alternatively, in cases where the corresponding model of interest is complex and equations cannot be solved directly, the expectation-maximization (EM) algorithm [256] can be used to find local maximum likelihood parameter estimates. Typically, these models also involve latent variables (e.g., due to missing values in the data) and as a result, computing the derivatives of the likelihood function results in a set of equations in which the solution to the unknown parameters requires the values of the latent variables and vice versa. The EM algorithm, on the other hand, constitutes an iterative algorithm that solves these equations numerically. The main idea is to alternate between an expectation (E) step, which generates a function for the expectation of the log-likelihood function evaluated using the current value of the parameters estimate, and a maximization (M) step, which estimates the parameters by maximizing the expected log-likelihood function computed at the E step. Finally, the estimated parameters are used to determine the distribution of the latent variables in the following E step.

Schulam *et al.* [260] address the problem of nonhomogeneity among patients of diseases such as autism or cardiovascular disease and point out that in order to have an effective treatment for these patients, they need to be subcategorized into homogeneous classes where treatments have similar influence. To this end, they propose the Probabilistic Subtyping Model that clusters time series of clinical markers and use EM to find the best parameter estimates for their model. They evaluate their proposed model on multiple types of diseases and illustrate that their model is able to discover subtypes of these diseases. For example, when they track the thickness of skin score (TSS) in Scleroderma patients, they show 5 different possible trajectories for it through time that most of the patient TSS values follow.

3) BAYESIAN ESTIMATION

In *Bayesian estimation*, prior knowledge with respect to the unknown parameters is incorporated into the estimation process along with the available set of observations. This prior knowledge usually results from previous observations or engineering assumptions and takes the form of a probability distribution over the model parameters. By incorporating this prior information about the parameters, a posterior distribution for the parameters can be obtained through Bayes' rule:

$$p(\theta|\mathbf{y}) = \frac{p(\mathbf{y}|\theta) \cdot p(\theta)}{p(\mathbf{y})}, \quad (39)$$

where $p(\theta|\mathbf{y})$ is the posterior distribution over the parameters θ , $p(\mathbf{y}|\theta)$ represents the likelihood function of the data \mathbf{y} given the parameters θ , $p(\theta)$ denotes the prior distribution over the parameters θ , and $p(\mathbf{y})$ is the marginalized likelihood. Note that it is usually better to work with prior probability distributions that facilitate analytical tractability, which are referred to as *conjugate priors* [254], such as

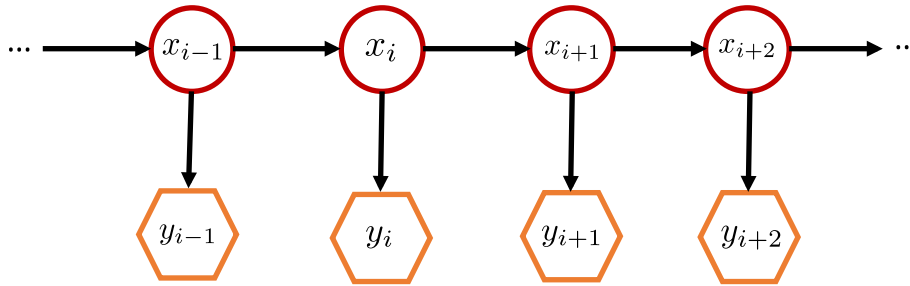


FIGURE 10. Hidden Markov Model Schematic. Each shape represents a random variable, where circles and hexagons denote hidden states and observations, respectively. Arrows denote conditional dependencies between random variables.

the binomial-beta distribution pairs. At this stage, parameter estimation is typically performed by selecting the set of parameter values that maximizes the posterior distribution defined in Eq. 39. This approach is referred to as *maximum a posteriori estimation* and can be performed either analytically or using the EM algorithm discussed above. Alternatively, one can adopt a more principled approach and utilize the entire posterior distribution for performing statistical inference (e.g., by marginalizing over the model parameters). Finally, in certain cases where there are no closed form expressions for the prior or posterior distributions, sampling-based approaches [198] can be used along with density estimation techniques [68] to approximate any probability distribution of interest and perform parameter estimation.

Marlin *et al.* [261] show an example application of Bayesian estimation by creating a model that captures patterns in physiological data of patients and place the patients that have similar patterns in their data in the same group. Because the data for their model is gathered from patients through time, it is sparse, have missing values, and in some cases have high error rates. For the model to be smooth, there is a need for smoothing the gathered data by taking into account the values that were measured prior to the new measurements for which the authors use Bayesian estimation to create a degree of smoothness in the model. Authors show that their model is able to distinguish among patients that have a higher mortality rate and a lower mortality rate and is able to have good trajectories for the physiological data of the patients in time.

VIII. KNOWN MODELS

In this section, we will study a selected set of widely applicable models, and discuss their characteristics. A large set of these models have demonstrated to be a good fit in various healthcare applications. The applicability of a particular model to a given healthcare application depends on how well the model can describe the phenomenon of interest as well as its complexity. For example, although Model A may be more accurate than Model B, it may also be a lot more complex. Thus, if the application is more sensitive to execution time, Model B may be preferred for that specific application. For instance, Wang *et al.* [262] study mobile health monitoring

systems and the trade-off between latency and accuracy for processing ECG data. They show that more accurate settings will result in higher latency and to achieve lower latency in data processing, accuracy of the results must be sacrificed.

A. HIDDEN MARKOV MODEL (HMM)

A Hidden Markov model (HMM) [263]–[266] is a statistical Markov model, where the related process is modeled by a Markov process with unobserved/hidden states. The Markov assumption ensures that computations remain tractable by imposing that future states depend only on the current state and not on the entire history of the states. In this model, the state of the process is not directly observed. Instead, we have access to a set of noisy observations, which are probabilistically related to the hidden state values. As a result, the set of noisy observations provides some information about the sequence of hidden states. HMMs are widely used in various applications such as speech and handwriting recognition [267], [268], bioinformatics [269], and activity recognition [270].

The general architecture of an HMM is shown in Fig. 10. Each shape represents a random variable, where circles denote states (i.e., x_k is the hidden state at time k) and hexagons denote observations (i.e., y_k is the observation at time k). The arrows represent conditional dependencies between random variables, and $p(x_k|x_{k-1})$ and $q(y_k|x_k)$ characterize the state and observation transition probabilities, respectively. We observe that due to the Markov property, the conditional distribution of the hidden state x_k at time step k depends only the value of the hidden state x_{k-1} at time step $k - 1$. Furthermore, the value of the observation y_k at time step k depends only the value of the hidden state x_k at time step k . In standard HMMs, the state space is discrete, while the observation space can be either discrete or continuous (e.g., Gaussian distribution).

The most common task performed in HMMs is inference. A typical inference task pertains to the computation of the probability of a specific observed sequence given the HMM model parameters as shown in Eq. (40), as shown at the top of the next page.

On the other hand, one may want to determine the probability of a state subsequence based on a sequence of observations

$$P(y_0, y_1, \dots, y_{L-1}) = \sum_{x_0, x_1, \dots, x_{L-1}} P(y_0, y_1, \dots, y_{L-1} | x_0, x_1, \dots, x_{L-1}) P(x_0, x_1, \dots, x_{L-1}). \quad (40)$$

and the model parameters. Filtering refers to the problem of determining the distribution $P(x_k | y_0, y_1, \dots, y_k)$ of the hidden state x_k at time step k given the history of observations up to time step k . Smoothing, on the other hand, refers to the problem of determining the probability $P(x_k | y_0, y_1, \dots, y_L)$ of hidden state x_k at time step k given the history of observations up to time step n , where $L > k$. Last but not least, finding the most probable sequence focuses on finding the probability $P(x_0, x_1, \dots, x_{L-1} | y_0, y_1, \dots, y_{L-1})$ of the entire sequence of hidden states that generated a specific observation sequence. For a detailed exposition of HMM models and algorithms to accomplish the above tasks, the interested reader is referred to [263]–[266].

HMMs are widely used in the healthcare literature and have witnessed great success. For instance, Qin *et al.* [271] use hidden Markov models to detect if individuals are smoking using data collected from mobile phone sensors. They are able to demonstrate that their model achieves an average AUC of 0.66 in detecting smoking activity. As another example, Son *et al.* [272] develop a smart asthma management system that models a patient's frailty as an HMM. Their proposed system consists of a Bluetooth device attached to a rescue inhaler, which records and transmits the usage patterns of the inhaler. The collected data is fed into an HMM that keeps track of the patient's frailty in addition to providing decision support for asthma management. The authors show that their system is able to estimate the rescue inhaler usage while providing a diagnostic classification on the asthma control status with high accuracy compared to other methods.

B. TIME SERIES MODELS

Time series models are used to represent a series of data points throughout time and can take various forms. There are three broad classes of linear time series models: (i) autoregressive (AR), (ii) integrated (I), and (iii) moving average (MA). An AR model has the form of a stochastic difference equation, where each time series variable linearly depends on all previous variables of the time series and a random variable term.

1) AUTOREGRESSIVE (AR) MODEL

In its simplest form, an AR model of order p (AR(p)) is expressed as follows:

$$X_t = \sum_{i=1}^p \alpha_i X_{t-i} + \varepsilon_t, \quad (41)$$

where $\alpha_i, i = 1, 2, \dots, p$ constitute the model parameters, and ε_t corresponds to white noise.

Yu *et al.* [273] use an autoregressive model to predict seizures in temporal lobe epilepsy. The input to their model

is raw EEG recordings and they model it using an AR model and use the coefficients of the extracted model as features of a logistic regression classifier to classify the state of the EEG as preictal (before seizure) or interictal (during seizure). The error rate of their classification scheme is 4.4%; see Section X-H for a detailed description of Logistic Regression.

Another example of using AR models is presented in [274], where authors classify EEG signals for use in a brain-computer interface (BCI). They use a Support Vector Machine (SVM) classifier to classify different tasks, such as mental multiplication of two numbers, counting tasks, and visualizing of a rotating 3D object, from EEG signals. They use AR models to model the EEG signals and use the coefficients of the model as inputs to their classifier. The accuracy of their framework ranges from 80% to 100% for different subjects; see Section X-A for a detailed description of the SVM classifier.

2) MOVING AVERAGE (MA) MODEL

In an MA model, each time series variable linearly depends on the current and previous variables of a stochastic sequence. In its simplest form, an MA model of order q (MA(q)) is expressed as follows:

$$X_t = \varepsilon_t + \sum_{i=1}^q \beta_i \varepsilon_{t-i}, \quad (42)$$

where β_i (for $i = 1, 2, \dots, q$) constitute the model parameters, and $\{\varepsilon_{t-i}\}_{i=0}^q$ represents a white noise sequence, which is usually assumed to consist of independent and identically distributed random variables, sampled from a normal distribution with zero mean and variance σ^2 .

An MA model is used in [275] for noninvasive central aortic systolic pressure (CASP) estimation. For the process to be noninvasive, authors use radial artery pressure waveform and develop and validate an MA model based on the waveforms to estimate CASP. They show that the output of their model has high correlation with the real sensed numbers of CASP.

3) AUTOREGRESSIVE MOVING AVERAGE (ARMA) MODEL

Combining an AR model with an MA model gives rise to an autoregressive moving average (ARMA) model, which has both AR and MA components. An ARMA model that consists of p autoregressive terms and q moving-average terms (ARMA(p, q)) is defined as follows [276], [277]:

$$X_t = \varepsilon_t + \sum_{i=1}^p \alpha_i X_{t-i} + \sum_{j=1}^q \beta_j \varepsilon_{t-j}, \quad (43)$$

where terms α_j and β_j are defined earlier.

ARMA models are used in [278], where authors estimate the radial artery blood pressure from blood pressure data

that is obtained from noninvasive monitoring systems. They model the transfer of blood from the heart to the fingertips using an ARMA model and find the radial artery blood pressure by sensing the blood pressure on fingers. They show that although their model still has a non-negligible error, the bias of their model is improved compared to other methods used for the same application.

Generalizations of the above models such as the autoregressive integrated moving average (ARIMA) model, the autoregressive moving average with exogenous inputs (ARMAX) model, the autoregressive conditional heteroskedasticity (ARCH) model, and the generalized autoregressive conditional heteroskedasticity (GARCH) model can be used to model more complicated trends in time series.

Revels *et al.* [279] use an ARIMA analysis to model obesity-related data and predict the percentage of the overweight people in the United States in coming years. They gather data for the prevalence of overweight people among adults from CDC (Center for Disease Control) and fit different ARIMA models to forecast the change in the population of overweight, obese, and morbidly-obese people. Their models also show the prediction of healthcare costs of this issue in the future.

An ARMAX model usage is presented in [280], where authors model coronary heart disease development. Their model takes different data such as blood glucose, cholesterol, and blood pressure of individuals throughout multiple years as inputs and predicts the severity of coronary heart disease in subjects. Their model agrees with the already-implemented clinical scores for the same disease and on majority of the subjects; the ARMAX model has a $\leq 10\%$ error rate compared to the standard score.

For a detailed survey of different time series models along with discussions regarding how to choose their order and estimate the associated coefficients, the interested reader is referred to [276], [277], [281], and [282].

C. STOCHASTIC STATE SPACE MODELS

Similar to time series models, stochastic state space models can be used to represent a series of data points throughout time. Within this context, the evolution of the data points in time is modeled as a dynamical system subject to random noise, where there exists a probabilistic dependence between a latent state variable, known as the *system state*, and an observed measurement, termed *observation*. Both the system state and observation can be either discrete or continuous and in that sense, HMMs constitute a special case of stochastic space models with discrete system state and observations. One of the most widely used stochastic state space models is the **general linear stochastic state space model**, which is described by Eq. (44) and Eq. (45), as shown at the bottom of the next page where Eq. (44) describes the system state evolution and Eq. (45), as shown at the bottom of the next page represents the relationship between system state and observations. Note that $\mathbf{x}(k)$ represents the state vector

(with the initial state $\mathbf{x}(0)$), $\mathbf{y}(k)$ denotes the observation vector, $\mathbf{u}(k)$ is a deterministic control input vector, $\Phi(k)$ and $\mathbf{H}(k)$ comprise deterministic coefficient matrices, and $\{\mathbf{w}(k)\}$ and $\{\mathbf{v}(k)\}$ constitute white noise processes uncorrelated with each other.

An example application of this model is presented in [283], where the authors aim to remove ballistocardiogram (BCG) components from an EEG signal induced by the strong magnetic field of MRI. Since BCG noise is induced on both the EEG and electro-oculogram (EOG) signals, authors filter the EOG signal and compare it to the EEG signal; using this methodology, they find the common induced noise created by the BCG. Note that because the BCG noise characteristics change in time (e.g., consecutive heartbeats do not have the same BCG characteristics), filtering of the noise is achieved via a Kalman filter. The study shows that this method is effective in removing the noise with real-time response.

In many real-world applications, a nonlinear model is better suited to describe the nonlinear phenomena that may take place. In this case, the *general nonlinear stochastic state space model* described by Eq. (46) and Eq. (47), as shown at the bottom of the next page is widely used where Eq. (46) describes the system state evolution and Eq. (47) represents the relationship between system state and observation. Note that it is usually assumed that both functions $\mathbf{f}_k(\cdot)$ and $\mathbf{h}_k(\cdot)$ are continuous and continuous-differentiable with respect to all the elements of the state and control input vectors. Furthermore, the noise sequences $\{\mathbf{w}(k)\}$ and $\{\mathbf{v}(k)\}$ satisfy the same properties as in the case of the linear stochastic state space model.

An example of nonlinear modeling is presented in [284], where the authors study continuous glucose monitoring. The problem that is addressed by the paper is that estimating the errors of sensors or glucose pumps in the long-term is not achievable with a linear model; therefore, it is modeled using a nonlinear model using an extended Kalman filter (EKF). The authors show that this model is able to estimate the blood glucose levels with low error rates; furthermore, the error rate decreases as the complexity of the EKF increases and more data is available to use in the EKF.

Another study that uses a nonlinear model is conducted in [285], where the authors detect regional heart motion abnormalities by using an unscented Kalman filter (UKF). This study models the dynamic behavior of the heart's left ventricle in time, which is a nonlinear system and the gathered data for modeling is noisy; these characteristics make UKFs a good fit for modeling and estimating the state of the left ventricle. The estimated state produced by the UKF is then classified by a classifier algorithm to detect abnormal cases among the patients. Their system is able to achieve a $\approx 90\%$ accuracy in detecting these cases.

The most common tasks performed in stochastic space models are parameter estimation and system state estimation. Parameter estimation involves obtaining estimates of a collection of parameters that appear in the stochastic space

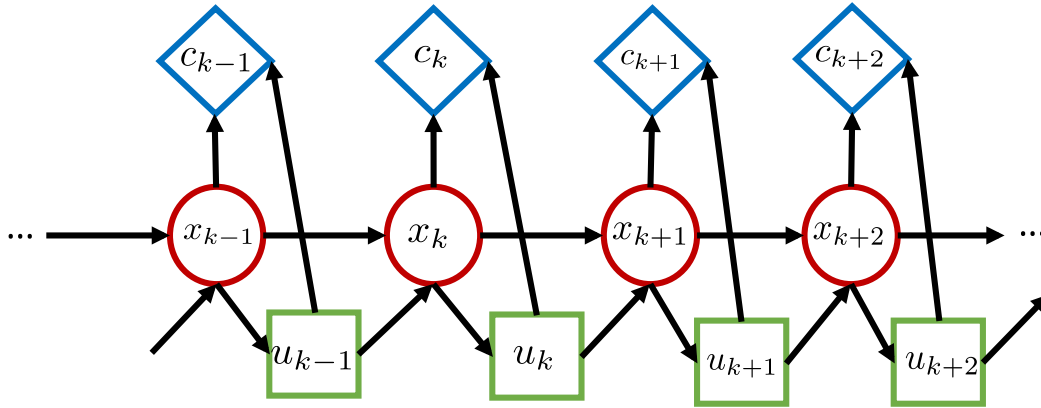


FIGURE 11. Markov Decision Process Block Diagram. Note that $x_i \in \mathcal{X}$ and $u_i \in \mathcal{U}$.

model and can be achieved by any of the methods discussed in Section VII-E. There are three types of system state estimation:

- (i) *prediction*, which focuses on estimating the state at a future time k , given all past observations (including the most recent observation at time $j < k$),
- (ii) *filtering*, which focuses on estimating the state at time k , given all past observations (including the most recent observation at time k), and
- (iii) *smoothing*, which focuses on estimating the state at time k , given both past and future observations (i.e., observations available at a future time $j > k$).

The Kalman filter discussed in Section X-B, along with its extensions, the Extended Kalman filter (EKF) and the Unscented Kalman filter (UKF), are used for filtering, while the other two tasks use the same principles as these filters to perform prediction and smoothing. For a detailed exposition of stochastic space models and the related algorithms, the interested reader is referred to [183], [255], and [286].

D. MARKOV DECISION PROCESSES

As shown in Fig. 11, a Markov Decision Process (MDP) [182]–[184] is a stochastic control process that consists of the following components:

- $\mathcal{X} = \{x_i \in \mathbb{R} \mid i \in \{0, 1, \dots, k, k + 1, \dots\}\}$: a finite set of process states (e.g., patient states, disease types).
- $\mathcal{U} = \{u_i \in \mathbb{R} \mid i \in \{0, 1, \dots, k, k + 1, \dots\}\}$: a finite set of control actions (e.g., medical tests, sensor types).
- $P : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \rightarrow [0, 1]$: a set of transition probabilities between the process states that capture the dynamics of

the system model (i.e., how the system moves from one state to another).

- $c_k : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$: a function that assigns rewards (or costs) to state transitions to model payoffs associated with them.

Given an MDP, the goal is to select control actions that optimize (minimize or maximize) a certain criterion. In practice, there are three commonly used criteria:

- Total expected reward:

$$J = \mathbb{E} \left\{ \sum_{k=0}^{T-1} c_k(x_k, u_k) + c_k(x_T) \right\}, \quad (48)$$

where x_k , u_k and $c_k(x_k, u_k)$ denote the state, the control action, and the reward/cost at time step k , respectively.

- Total discounted expected reward is:

$$J = \mathbb{E} \left\{ \sum_{k=0}^{T-1} \gamma^k c_k(x_k, u_k) + \gamma^T c_k(x_T) \right\}, \quad (49)$$

where $0 < \gamma \leq 1$ is a discount factor that weighs the contribution of different states differently in time, and

- Average expected reward is:

$$J = \frac{1}{T} \mathbb{E} \left\{ \sum_{k=0}^{T-1} c_k(x_k, u_k) + c_k(x_T) \right\}. \quad (50)$$

The horizon length T can be either finite or infinite depending on the application requirements and the optimal decision strategy can be determined by using dynamic programming (DP) techniques [183] (see also Section X-G1).

$$\mathbf{x}(k) = \Phi(k-1) \cdot \mathbf{x}(k-1) + \Psi(k-1) \cdot \mathbf{u}(k-1) + \mathbf{w}(k-1), \quad k = 1, 2, \dots, \quad (44)$$

$$\mathbf{y}(k) = \mathbf{H}(k) \cdot \mathbf{x}(k) + \mathbf{v}(k), \quad k = 1, 2, \dots, \quad (45)$$

$$\mathbf{x}(k) = \mathbf{f}_k(\mathbf{x}(k-1), \mathbf{u}_{k-1}) + \mathbf{G}(k-1) \cdot \mathbf{w}(k-1), \quad k = 1, 2, \dots, \quad (46)$$

$$\mathbf{z}(k) = \mathbf{h}_k(\mathbf{x}(k)) + \mathbf{v}(k), \quad k = 1, 2, \dots, \quad (47)$$

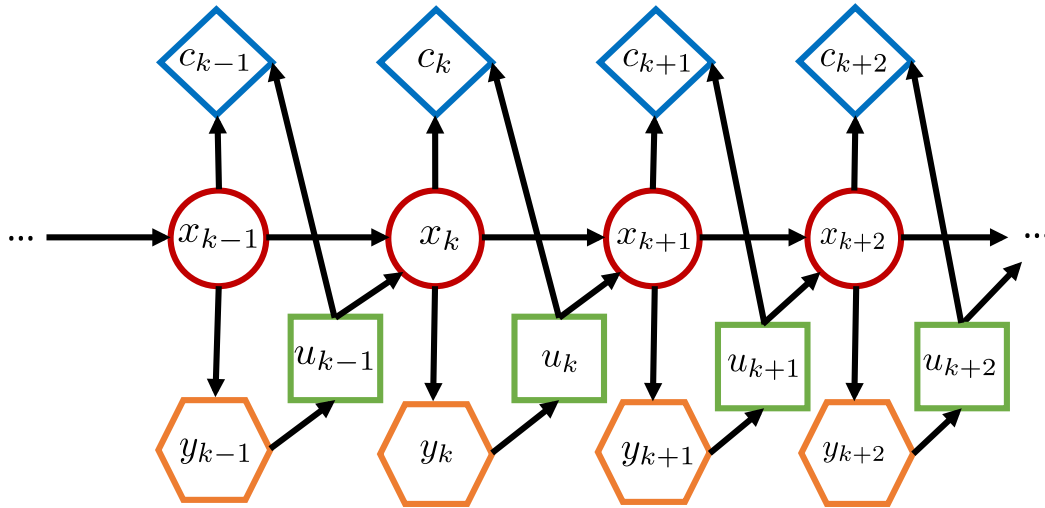


FIGURE 12. Partially Observable Markov Decision Process Block Diagram. Note that $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, and $u_i \in \mathcal{U}$.

In [287], an MDP formulation was proposed to decide on a planning therapy for individuals with spherocytosis, which is a disease that results in chronic destruction of red blood cells. The cost was defined in terms of quality adjusted life years and the transition probabilities were estimated based on factors such as risk of surgical mortality and natural causes of death. Assuming year-by-year decisions, the authors were able to determine the optimal treatment strategy in closed form.

The goal of the work in [288] was to determine when adherence-improving interventions are necessary based on an individual’s electronic health record (EHR). To this end, an MDP model formulation was proposed and the optimal DP intervention strategy was evaluated for cardiovascular disease management of 54,036 patients with type 2 diabetes. It was shown that the use of such a framework can delay the onset of adverse events or death and reduce expected costs of treatment, hospitalization, and follow-up care. The authors also provide structural results with respect to interventions of different effectiveness and patients of different risk levels.

In many applications, a decision maker may have several objectives (e.g., a doctor may wish to try a treatment, while minimizing consequences, a health monitoring system may need to operate under energy or time constraints). In this case, one can use an MDP formulation, where the cost function is defined as the weighted sum of the different objectives. Alternatively, one may use a constrained MDP (CMDP) [289] formulation, where one of the performance criterion is optimized, while the rest are kept below some given thresholds. Note that the basic components of the CMDP model coincide with the MDP components.

Wang et al. [290] develop a framework based on CMDP to obtain sensor signals from a mobile device to detect the context of the environment and the state of the user. They categorize the state of the user as {stable, moving, in contact, not in contact} and show that their framework can detect

the state with low error. The main contribution of the paper is that it shows that by utilizing a framework based in CMDP, they can reduce the number of sensing periods and increase the battery life of the mobile device as a result.

E. PARTIALLY OBSERVABLE MDPs

In all the above models, we have assumed that the system state is perfectly observable. Unfortunately, this is not the case in many applications, especially in the healthcare domain. A partially observable Markov decision process (POMDP) [183], [185] is a stochastic control process that assumes that system states are hidden, but the decision maker has access to noisy observations. Furthermore, control actions can either be used to drive the system into a particular state (e.g., administer certain drugs to cure a disease) or infer the system state (e.g., determine if a patient has a certain disease).

A POMDP consists of the following components (see also Fig. 12):

- $\mathcal{X} = \{x_i \in \mathbb{R} \mid i \in \{0, 1, \dots, k, k + 1, \dots\}\}$: a finite set of process states.
- $\mathcal{U} = \{u_i \in \mathbb{R} \mid i \in \{0, 1, \dots, k, k + 1, \dots\}\}$: a finite set of control actions.
- $\mathcal{Y} = \{y_i \in \mathbb{R} \mid i \in \{0, 1, \dots, k, k + 1, \dots\}\}$: a finite set of observations (e.g., results of medical tests).
- $P : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \rightarrow [0, 1]$: a set of system state transition probabilities.
- $f : \mathcal{X} \times \mathcal{U} \times \mathcal{Y} \rightarrow [0, 1]$: a set of observation probabilities that capture the relationship between states, control actions and observations.
- $c : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$: a reward/cost function

Since the actual system state is hidden, decisions are based on the history of observations and control actions performed so far. More specifically, the belief state \mathbf{p}_k [183], which is defined as the following conditional probability over the

system states:

$$\mathbf{p}_k \triangleq [p_k^1, p_k^2, \dots, p_k^{|\mathcal{X}|}]^T, \quad (51)$$

where $p_k^i \triangleq p(x_k = i | y_0, \dots, y_k, u_0, \dots, u_{k-1})$, $|\mathcal{X}|$ is the size of the state space, is used at each time step by the decision maker to select control actions. Note that in contrast to MDPs, where we have access to a finite number of states that are fully observable, a belief state can take an infinite number of values. In POMDPs, we can use the same optimization criteria as with MDPs and dynamic programming can be applied to find the optimal decision strategy (see also Section X-G2).

In [35] and [188], a POMDP formulation was devised to perform energy efficient physical activity detection for obesity prevention using the KNOWME wireless body area network (WBAN) [188]. The KNOWME WBAN consists of a set of biometric sensors (e.g., accelerometer, electrocardiograph) and a Nokia N95 mobile phone. The goal was to design sensing strategies for the mobile phone to dynamically decide from which biometric sensors to receive data at each step, so that the physical activity of the individual is detected (e.g., run, sit, stand, walk), while maximizing the lifetime of the mobile phone. Sensors are heterogeneous in energy usage and detection capabilities for different physical activities. The authors derived the optimal sensing strategy via DP and three low-complexity, near-optimal sensing strategies. Evaluation on real data collected from the KNOWME network show energy gains as high as 64% with detection error in the order of 10^{-4} . Main challenges of the study are the need for personalization and the estimation of physical activities transitions.

Another application of POMDP is shown in the COACH system [21], which uses a single video camera to track and assist individuals with dementia during the task of handwashing. A POMDP model was formulated to estimate the individual’s level of dementia and assist him/her through the various activity steps. Based on the individuals’ ability to correctly perform the handwashing task, there are three actions of intervention: assistance prompts (e.g., task description, cue the individual), do nothing, and call caregiver. The COACH system has been tested with six individuals of varying degrees of dementia achieving a 25% reduction on caregiver interventions. Main challenge is the need for personalization with respect to system intervention and detection of certain steps of the handwashing task.

F. SEMI-MARKOV MDPs AND SEMI-MARKOV POMDPs

We have so far focused on models where the time between decisions is fixed. However, in some cases, such as physical activity tracking or treatment planning, decisions need to be made continuously. Semi-Markov decision processes (SMDPs) and partially observable SMDPs (POSMDPs) [190] are generalizations of the MDP and POMDP stochastic models, where the time between state transitions may depend on the selected control action or occur randomly. The optimal decision strategy in these cases can be determined by using techniques similar to MDPs and POMDPs [190].

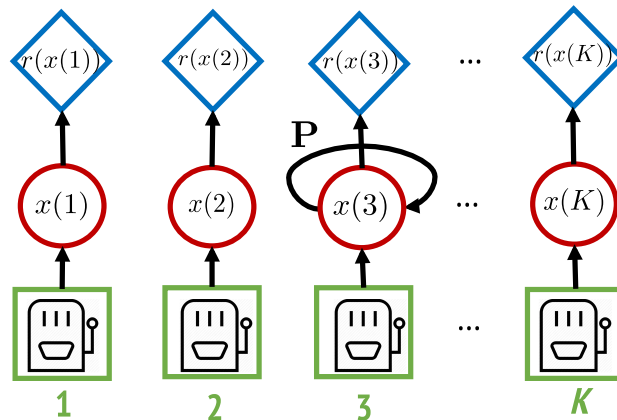


FIGURE 13. Multi-Armed Bandit Block Diagram.

Wang *et al.* [291] develop a human activity recognition framework through mobile sensing and real-time state estimation with a semi-Markov-like mechanism. They optimize their model to balance the battery power consumption and error rate. Although they can achieve 0% error in their state estimation, they show that by having an error rate of 1.66% they can triple the lifetime of the battery in the mobile sensing device.

G. MULTI-ARMED BANDITS

The Multi-Armed Bandit (MAB) [186], [187] problems constitute a special class of decision-making problems that focus on the trade-off between *exploration* (i.e., explore decision options with the prospect of better rewards) and *exploitation* (i.e., continue with decision options that are provingly associated with high rewards). MAB formulations can be used among others to model medical tests performed on individuals to infer a disease or available treatments to treat a disease.

In its simplest form (see Fig. 13), the decision maker has access to K different arms (options), each of which has an associated reward. At each time step, she decides to use exactly one arm and observes the reward associated with it. The goal of the decision maker is to maximize the reward defined below by selecting to pull an appropriate arm at each time step:

$$J = \mathbb{E} \left\{ \sum_{k=0}^{\infty} \gamma^k r(x_k(u_k)) \right\}, \quad (52)$$

where $\gamma \in (0, 1)$ is a discount factor, $r(\cdot)$ is a bounded reward function, and x_k denotes the state of arm u_k that was selected at time step k . Note that the state of the selected arm evolves in a Markovian way with respect to a known transition probability matrix \mathbf{P} ; alternatively, the states of the rest of the arms remain the same. Another slightly different formulation of the MAB problem focuses on minimizing the *regret*; the goal of the decision maker is to maximize her reward (or minimize her cost) with respect to the best decision strategy that could be used if she had access to the rewards

(or costs) of all arms [292]. The above formulation is known as the *stochastic MAB problem* and can be solved via dynamic programming (see also Section X-G3).

Various MAB formulations and algorithms have been proposed and studied in the literature. Examples are the *Markovian MAB* [186] (the reward of each arm follows a Markov distribution), the *adversarial MAB* [186] (the reward of each arm is generated by an adversary), the *contextual MAB* [293] (the decision maker has access to a context vector before making a decision) and *imperfect state MAB* [294] (the state of each arm is observed through noisy measurements).

An example application of MAB is presented in the MyBehavior system [26], which focuses on inferring an individual’s physical activity and dietary behavior and suggests changes that can lead to a healthier lifestyle. The platform that maximizes calorie loss while ensuring that the suggestions are easy to adopt is based on a MAB formulation; the EXP3 strategy [295] is used to select suggestions, where beneficial behaviors are frequently adopted in contrast to less beneficial ones. The MyBehavior system was evaluated during a 14-week study with 17 participants, where it was shown that subjects increased physical activity and decreased food calorie intake. Another example is provided in [189], where a contextual MAB formulation is proposed that matches stress-coping interventions to individuals and their temporal circumstances over time. The expected stress reduction of each intervention was determined experimentally for each individual at a given context.

For more information regarding the MAB problem and its variants, the reader is referred to [186].

H. LATENT DIRICHLET ALLOCATION (LDA)

Latent Dirichlet allocation (LDA) [296], [297] constitutes a generative probabilistic model used to describe collections of discrete data such as text corpora and population genetics. It constitutes a hierarchical Bayesian model according to which, (i) each item of the collection is modeled as a finite mixture of an underlying set of topics, and (ii) each topic is modeled as an infinite mixture of an underlying set of topic probabilities. Within the context of natural language processing, this means that documents can be represented as random mixtures of hidden topics, where each topic is characterized by a distribution over words. In this sense, LDA can be graphically represented as shown in Fig. 14, where the outer rectangle represents documents, and the inner rectangle denotes words. The generative process adopted by LDA for each document w in a corpus D has the following structure:

- 1) Choose the number of words $N \sim (\text{Poisson}(\xi))$.
- 2) Choose the topic mixture $\theta \sim \text{Dir}(\alpha)$
- 3) For each of the N words in the document:
 - a) Pick a topic $z_n \sim \text{Multinomial}(\theta)$
 - b) Use the topic to generate the word itself.

An example application of LDA is shown in [298], where the authors link the physical structure of drugs to

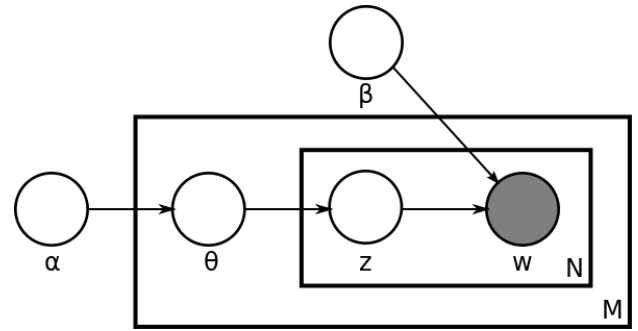


FIGURE 14. Latent Dirichlet Allocation Model.

the adverse events that are caused by them. The data required for this tasks comes from adverse drug reaction reports and the authors build a model based on LDA that assigns topics to different reports. They also create representative features for different drug structures, where drugs that share some substructures are represented with features that are shared among them. They design a predictive model that relates drug structure features to topics of the adverse event reports and show that their scheme perform better than other methods in predicting the outcome of drugs.

I. SURVIVAL ANALYSIS/PROPORTIONAL HAZARD MODELS

Proportional hazard models analyze the time that it takes for an event to occur. These models, if used to predict the survival of a patient, find the most important features that affect the length of the survival time. The Cox proportional hazard model [299] is one of the most commonly used survival analysis models and is described by Eq. (53), where $\mathbf{x} \in \mathbb{R}^d$ is the feature vector for an individual, $h(t|\mathbf{x})$ is the hazard value at time t for that individual, $h_0(t)$ is a baseline hazard function, $\beta \in \mathbb{R}^d$ are the adjustable parameters, and d is the number of features.

$$h(t|\mathbf{x}) = h_0(t) \exp(\beta^T \mathbf{x}) \tag{53}$$

Lusivika-Nzinga et al. [300] use a variation of the Cox model to analyze how multiple treatments change the survival time of HIV-positive patients. The outcome of their model is the probability of an adverse event for a patient and they show that their model yields higher quality predictions compared to other methods.

The Kaplan-Meier survival rate estimator is another model that can be used to estimate the survival function. An example of the Kaplan-Meier estimator is used in [301], where authors profile the survival rate of dental restoration materials.

IX. ALGORITHMS (BOXES A, O, T)

Before we study a set of known algorithms, we will provide an overview of their goals (Section IX-A), training process (Section IX-B), and implementation challenges (Section IX-C).

A. GOALS OF ALGORITHMS (BOX A, O)

In this paper, we study machine intelligence algorithms based on their application goal using four different goals, which are detailed below.

1) KNOWLEDGE DISCOVERY

Algorithms used for knowledge discovery aim at discovering relations in a dataset. These algorithms work with datasets that are not labeled and do not indicate any presumption toward the data. Clustering and anomaly detection are two applications that focus on extracting knowledge from a dataset without knowing anything about it a priori. For example, in [30], the Food and Drug Administration (FDA) Adverse Event Reporting System database is analyzed (see Appendix A) to identify previously-unknown drug pairs that elevate the blood glucose level when used together.

2) CLASSIFICATION/DETECTION

When the data is divided into subpopulations (i.e. classes or categories), an algorithm needs to learn the relation between the input data points and the subpopulation that input data points belong to. The goal is to classify a new previously-unobserved data point into the subpopulation that it belongs. The output of a classification algorithm is always discrete, but the number of sub-populations varies based on the application; for example, if the goal of the algorithm is to detect a cardiac hazard among a set of three potential outcomes, {Healthy, LQT1, LQT2}, where LQT1 and LQT2 are the Long QT Syndrome Type 1 and Type 2 cardiac conditions [7], [10], this is a three-category classification problem. Alternatively, deciding if a discharged patient will return to the hospital in the next year or not, is a two-category (i.e., Yes/No answer) classification problem.

3) REGRESSION/ESTIMATION

In some cases, the output is not categorized as subpopulations, but is represented as a range of quantitative outcomes that usually belong to a continuous set. For example, predicting the next time that a discharged patient will be re-admitted to the hospital is a regression problem; based on the adopted model, it may have an answer such as “1–30 days” or “never.” In an example application of regression Schulam and Saria [167] predict the evolution of a patient’s health based on their initial health condition.

4) SEQUENTIAL DECISION-MAKING

In many medical applications, the goal is to infer/monitor a medical phenomenon, or intervene so as to improve a certain health condition, or both. In these cases, sequential decision-making models —such as MDP, POMDP, and MAB — can be used to model various decision-making tasks including but not limited to exploiting the inferred information to make decisions that can possibly improve the estimation or tracking task of interest. For example, Zois *et al.* [35] propose a POMDP model to automatically decide which features to

use from ACC and ECG signals to improve the tracking of the physical activity of an individual, while minimizing communication energy consumption.

B. TRAINING COMPONENT (BOXES O, T)

Most machine intelligence algorithms require a *learning* phase, during which the algorithm adjusts its internal parameters based on input-output data provided, and an *application* phase, during which the algorithm predicts an output when a previously-unknown input is provided. In general, seven different learning methods are employed in practice, namely *unsupervised learning*, *supervised learning*, *semi-supervised learning*, *active learning*, *reinforcement learning*, *learning to rank*, and *structured learning*. In this section, we discuss how these methods are applied to healthcare and any issues associated with them.

To train a machine intelligence algorithm, it is usually necessary to have access to input-output data. For example, consider an ECG recording for a cardiac patient, who had a myocardial infarction (MI) 10 minutes after arriving at an emergency room. When entering this ECG record in a database that is associated with this patient, the ECG recording corresponds to the input data, while the MI event corresponds to the output data, which is provided by a medical expert. The resulting status of the patient (referred to as the “end points” in medicine, which is “MI” in this case) is commonly referred to as a *label*; a database containing such labels is said to contain *labeled data*. Alternatively, if the raw ECG recordings of the same patient are provided in a database with no end points associated with these ECG recordings, this database contains *unlabeled data*. The type of machine intelligence algorithm that can be used for a given healthcare application is determined by the labeling that is available in the acquired data.

1) UNSUPERVISED LEARNING

Unsupervised learning [302], [303] allows machine intelligence algorithms to operate on unlabeled data; because the output values are not known, the goal of such algorithms is knowledge discovery, as discussed in Section IX-A1. Clustering is an example of an unsupervised learning algorithm, which divides a dataset into multiple “clusters” that share common characteristics. Due to the elimination of the output pairs, a training phase is consequently eliminated in unsupervised learning.

2) SUPERVISED LEARNING

Algorithms that use supervised learning [303]–[305] utilize databases that contain labeled data. The labels that are provided in a given database are converted to quantitative output values for use in machine intelligence algorithms; for example, the aforementioned “MI” label can be mapped to a quantitative metric 1.000, while another label, such as “healthy,” can be mapped to 0.000. Such a label-to-quantity mapping allows the algorithm to produce human-interpretable labels as its output (e.g. “MI”, which is interpreted as the “patient has

a danger of heart attack”), while operating internally using purely quantitative values.

3) SEMI-SUPERVISED LEARNING

In semi-supervised learning [306], [307], the learning processes use both labeled and unlabeled data. Typically, we have access to a smaller amount of labeled data, while the majority of the data is unlabeled. For instance, in [308], semi-supervised learning is employed to improve the performance of physical activity classifiers after they are deployed in the field.

4) ACTIVE LEARNING

In many medical applications, it is often impractical to obtain labeled data that is both correct and representative of all the possible input-output relations. Active learning [200] is a form of semi-supervised learning, in which the learning algorithm interactively queries an expert (i.e., user or any other information source) to obtain the outputs at unlabeled data points. For example, in [309], active learning is applied to the problem of medical image classification for heart disease and breast cancer detection, where the goal is to select a number of informative data points to maximize classification accuracy.

5) REINFORCEMENT LEARNING

Having the same motivation as active learning, reinforcement learning [310], [311] adopts an agent-based approach based on which learning input-output relations are modeled as an interaction between an agent and its environment with the goal of selecting actions to maximize some notion of cumulative reward. This is in direct contrast to unsupervised learning, where the goal is to find similarities and differences between data points. One of the main challenges of reinforcement learning is the trade-off between exploration and exploitation: to maximize reward, the agent tends to select actions that have been found to be effective in the past; on the other hand, to discover such actions, the agent must select new untested actions. Within the context of healthcare, reinforcement learning is used to determine optimal structured treatment interruption strategies for HIV infected patients directly from clinical data [312].

6) LEARNING TO RANK

Learning to rank [313], [314] involves the construction of ranking models typically for information retrieval systems. The goal is to exploit techniques from supervised, semi-supervised, or reinforcement learning to produce an accurate ranking of data points for which the rank is unknown. In the medical domain, learning to rank has been successfully applied to predict a clinical score ranking for the severity of a disease based on fMRI images [315].

7) STRUCTURED LEARNING

Structured learning [316] is essentially a special case of supervised learning, where the focus is to learn the input-output relation between data points and structured objects

(e.g., tree, Bayesian network, random field), rather than scalar discrete or real values. Due to the complexity of the structured objects (i.e., number of inter-dependencies among variables), training in structured learning is frequently computationally infeasible and approximate learning approaches are usually employed. Within the context of medical applications, structured learning is employed in [317] to create a system that extracts relevant information from narrative clinical discharge summaries and generates an appropriate structured representation to enable fast and accurate clinical decision making.

8) TRAINING PROCESS

Using a large database containing labeled data, the training process involves splitting the database into two parts; the first part is used as the training data, which contains the majority of the entries in the database, while the second part is the test data, which contains the samples that are used to verify the performance of the algorithm (e.g., by using performance metrics described in Section VI). This process can also be repeated by choosing a different training vs. test set and is known as *cross-validation*. In many cases, a *loss function* is defined, which is a function of the difference between the predicted output data points and the real output data points. The main goal of the training process is to minimize this loss function by changing the internal parameters of the algorithm. This minimization is achieved through methods such as *Gradient Descent* that find the local minima of a function.

9) TRAINING ISSUES

Two major issues in the training process are *overfitting* and *underfitting* and are briefly described below:

- **Overfitting:** Overfitting arises when a model is excessively complex resulting in an inability to generalize to previously unseen data points. As a result, prediction performance during testing may be poor, even though the error during training may be minimal. Consider, for example, the problem of fitting a polynomial to a set of data points that exhibit approximately linear behavior with small noise fluctuations. Selecting a high-degree polynomial to describe the data in this case can potentially lead to a large number of errors during testing, since the underlying data trend is linear. Overfitting can be avoided using various techniques such as **(i)** adopting a simpler model, **(ii)** gaining access to more training data, **(iii)** removing redundant features, **(iv)** performing cross-validation, and **(v)** regularizing model parameters.
- **Underfitting:** Underfitting arises when the adopted model is not able to completely capture the structural relations that characterize the available data. For instance, assuming a linear model in the case of nonlinear data will result in poor prediction performance.

C. ALGORITHMIC IMPLEMENTATION CHALLENGES

Application requirements and resource limitations generally dictate in which cases a machine intelligence algorithm can be applied; for example, convolutional neural networks

(Section XI) were not practically-implementable until 2005, when the computational power of GPUs made them feasible. There are multiple important factors that need to be considered when deciding which machine intelligence algorithm is suitable for a given application. We briefly discuss these factors in this Section.

1) ACCURACY/PERFORMANCE

Accuracy of a machine intelligence algorithm is a measure of how successful the algorithm is in fulfilling its goal. For instance, the methods proposed in [35] for physical activity detection are evaluated with respect to average detection error (i.e., how many times the algorithm mis-classifies a specific activity on average), which is shown to be on the order of 10^{-4} . We discuss metrics used to quantify accuracy in depth in Section VI.

2) ROBUSTNESS

Robustness measures how effective a machine intelligence algorithm is when being tested with **(i)** datasets that contain outliers or **(ii)** models that contain parameters that are prone to errors. As an example of the latter case, if a linear model is assumed, but the actual correlation was super-linear, the accuracy of the algorithm should not be impacted significantly, i.e., the algorithm should be robust in handling imperfect models. In [318], an automatic liver CT scan segmentation—which is necessary for liver tumor ablations and/or radiotherapy—is proposed that shows good robustness behavior (F1-score = $94.2 \pm 1.1\%$).

3) COMPUTATIONAL COMPLEXITY

Computational complexity is a theoretical measure of the resources needed by a machine intelligence algorithm to achieve its goal. In general, machine intelligence algorithms need to be analyzed with respect to their *time complexity* (i.e., execution time as a function of the size/dimension of their input), *storage space complexity* (i.e., memory needs as a function of the number of storage locations used), *input/output complexity* (i.e., number of inputs and outputs between internal memory and secondary storage required) [319], and *communication complexity* (i.e., amount of communication required for distributed computations) [320]. Furthermore, since most machine intelligence algorithms consist of a learning phase and an application phase, we need to determine the computational complexity of these two phases. Determining the computational complexity of an algorithm is beneficial, since we can **(i)** decide which tasks should be executed *online* or *offline*, **(ii)** determine storage assignments (e.g., mobile phone versus remote server), and **(iii)** suggest modifications that improve the computational requirements of the machine intelligence algorithm. Let us consider the MIC-T3S algorithm for physical activity detection [35] as an example; the time complexity of the *training phase* is $\mathcal{O}(n^4\alpha L)$ and space complexity is $\mathcal{O}(n^3\alpha)$, where n denotes the number of different physical activities, α denotes the number of different sample combinations, and

L the planning horizon. The authors suggest to implement this phase (training) *offline*. In contrast, for the *application phase*, the time complexity is $\mathcal{O}(n^4)$ and the space complexity is $\mathcal{O}(n^3\alpha)$, which is executed *online*.

In reality, computational complexity of machine intelligence algorithms within the context of medical applications is usually evaluated with respect to the number of seconds/minutes/hours/days required for the learning and application phases. While a standard computational hardware (such as a laptop) may be employed to derive—and report—these runtime results, applications that can take advantage of massive parallelism can significantly benefit from servers that incorporate GPUs [321] and other specialized computational accelerators, such as FPGAs. An example of reporting the runtime of an algorithm is in [322], where authors detect mitosis in a single breast cancer histology image using deep convolutional neural networks, which requires roughly 8 minutes. As far as the input/output complexity is concerned, a 24-hour ECG recording with a 1000 Hz frequency and 12 leads with a precision of 16 bits, amasses 2 GBs of data and a collection of 1000 of these samples will have a size of 2 TBs. Assuming the data is stored in an Solid State Disk (SSD) with a reading speed of 500 MB/s, it takes more than an hour for just transferring the files from the SSD to the memory of the computer. If the data is stored in a hard drive rather than an SSD, this time will be an order of magnitude longer; this is not negligible and should be taken into consideration when dealing with large databases on machines with slow disks. Depending on the medical application of interest, it may be possible to overlap input/output operations with computations; this provides an avenue to overlap a portion of the I/O time with a portion of the actual computation time, thereby reducing the apparent time complexity of the algorithm.

4) DATA SPARSITY

Data sparsity describes the situation where there is insufficient training data to enable accurate statistical decisions. This is a very difficult and serious problem, since most machine intelligence algorithms require a significant amount of accurate training data to achieve the desired performance. Zhou *et al.* [323] develop a data-driven framework that addresses data sparsity within the context of robust patient phenotyping by exploring the latent structure of electronic medical records and apply it to the tasks of early prediction of Congestive Heart Failure and End Stage Renal Disease. *Class imbalance* constitutes a related problem, where the data samples of a class are far less than the number of data samples of another class. For example, the number of newborn infants with the Long QT heart disease was 17 in a database of 43000 infants [324], which represents a substantial imbalance between the available training samples (17 vs 43000) for two classes (healthy vs. LQT). In [325], the effect of class imbalance is investigated within the context of computer-aided medical diagnosis when using neural network classifiers and it is shown that standard backward

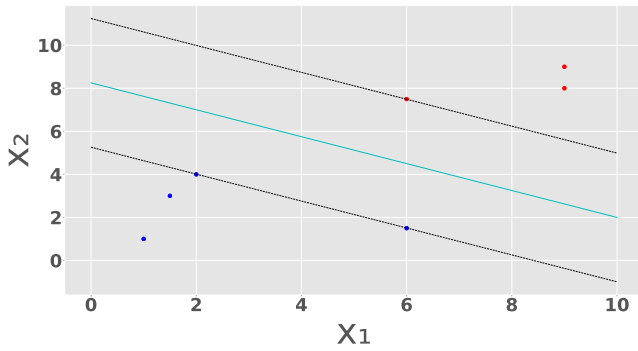


FIGURE 15. A depiction of an SVM classifier that separates the data points into two classes. The x_1 and x_2 axes show the input parameters and the color of the data points indicate the class that the data points belong to. The cyan line (in the middle) is the line that has the maximum distance to both classes.

backpropagation is preferable over more elaborate optimization techniques such as particle swarm optimization.

X. KNOWN ALGORITHMS

Due to the diverse training and performance metrics of different machine intelligence algorithms (as detailed in Section IX), a large set of them have demonstrated to be a good fit in various healthcare applications. The key for a machine intelligence algorithm’s applicability to a given healthcare application is not only the parameters that are associated with the algorithm, but also the demands of the application. For example, although an Algorithm A may be significantly more accurate than Algorithm B, it may also be much slower. If the application is more sensitive to the runtime, e.g., because of the criticality of the latency in determining an answer, Algorithm B may be preferred for that specific application. For example, Özdemir and Barshan [326] analyze the daily activities of a person using wearable sensors and aim to detect when the person falls. They use six different classifiers for this purpose and show that the training and testing time varies widely among these classifiers. Although every classifier achieves high accuracy, one of them has a response time of 33 seconds while the response time of the others are under 100 ms; therefore, the first algorithm is clearly not a good choice.

In this section, we will study a selected set of machine intelligence algorithms, which have demonstrated wide applicability, and elaborate on the characteristics and performance metrics of these algorithms that enabled them to be utilized in a given set of applications.

A. SUPPORT VECTOR MACHINES (SVM)

Support Vector Machines (SVMs) [327], [328] are used as classifiers and determine the boundaries of the hyperplanes that separate different classes in a dataset based on a distance metric, i.e., maximum possible separation. A simple linear SVM classifier is shown in Fig. 15, where the classifier computes the line that separates the two classes to yield the maximum distance between them.

Assume a dataset $\{(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)\}$, where $\vec{x}_i \in R^m$ is the i^{th} input with m dimensions, and n is the number of subjects in the training dataset. y_i is the class that the i^{th} subject belongs to and can take the values of 1 and -1 . A linear SVM classifier tunes \vec{w} and b in Eq. (54), so that the constraints in Eq. (55) are met:

$$\vec{w} \cdot \vec{x} - b = 0 \tag{54}$$

$$y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1 \text{ for all } 1 \leq i \leq n \tag{55}$$

The SVM classifier described above performs *linear classification*. SVM classifiers can use nonlinear *kernels* to classify data sets that are not separable by a linear function. Some of the common nonlinear kernels are *polynomial* and *hyperbolic tangent* kernels. SVMs can also be used for *regression* and *outlier detection*. Many variants of SVMs are used in the literature such as SVM-RFE (Recursive Feature Elimination) [329] and Least Squares Support Vector Machine [330].

SVMs are widely used in literature and have witnessed great success. Patel et al. [15] estimate the severity of certain Parkinson’s disease symptoms by feeding the accelerator data acquired from a wearable monitoring system into SVM classifiers. They reach estimation error rates less than 6%, and get error rates as low as 1.2% by using more features for their SVM. Koutsouleris et al. [18] use SVM classifiers to analyze pre-processed MRI images and classify each image as healthy vs. people who are at risk of early or late stages of psychosis. Their scheme results in classification accuracies of $\approx 90\%$ for each case. Another study using SVM classifiers is presented in [12], where the authors detect obstructive sleep apnea (OSA) by feeding single channel ECG recordings into an SVM and detect apnea episodes in patients. Their algorithm achieve accuracies of $\approx 90\%$ in best case scenarios.

B. KALMAN FILTERING

Kalman Filtering (KF) algorithm [331], [332] addresses the problem of estimating the state of a discrete-time linear dynamic system using multiple measurements, each of which contain a certain amount of statistical noise. Kalman filters work by determining the system state variables in two steps: (i) *Prediction* step uses the current state of the system (at time step $k - 1$) to make a prediction for time step k , based on a physical model and the (ii) *Update* step uses the actual measurements (at time step k) to fine-tune this predicted state to determine a more accurate state (at time step k). The process continues to determine the next state (time step $k + 1$) by using the next step’s measurements (at time step $k + 1$). The connection among Kalman filters, belief propagation for Bayesian networks, and some other related topics is studied in [333]. An overview of Kalman filters is provided in [286].

Kalman filters compute the least-squares estimate of the new state of a linear system from their previous state, which, in essence, is a recursive computation; because of this recursive nature of Kalman filters, they are very computationally efficient for a large number of measurements. As first proposed by Kalman [331] and Kalman and Bucy [332], this

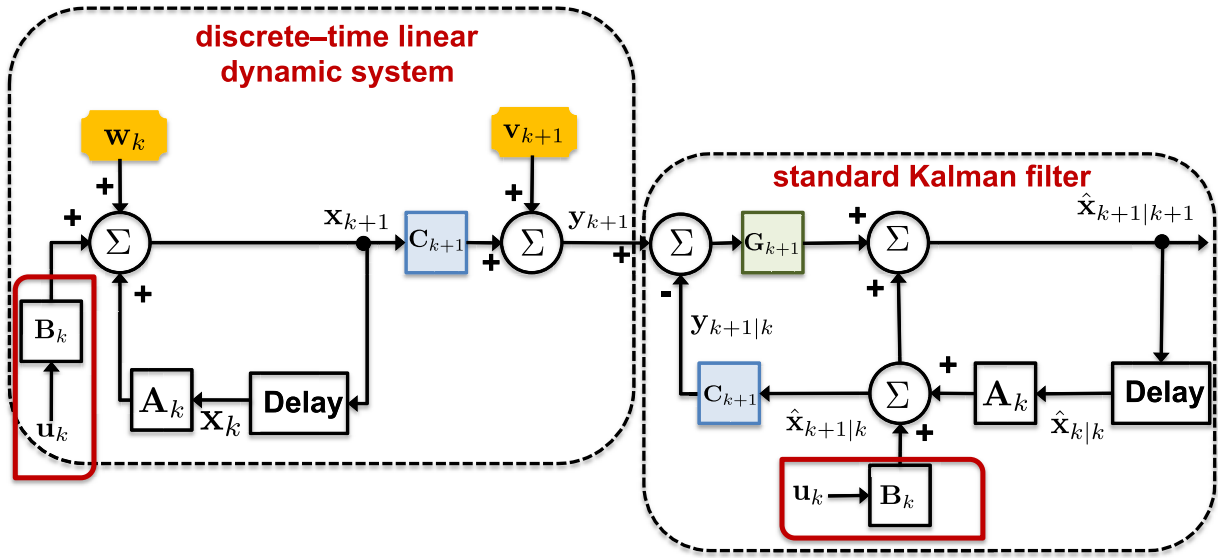


FIGURE 16. Interconnection of system block diagram and Kalman filter block diagram.

sequential structure of the problem can be taken advantage of to estimate at each time step by means of a simple equation that only involves the previous state estimate and the new measurement(s). Due to their low computational cost, the KF filter and its extensions have found a wide application domain, including target tracking [334], aircraft and spacecraft guidance, navigation and control of vehicles [335], robotic motion planning and control [336], physical activity tracking [27], and central nervous system movement control [337].

Consider the discrete-time linear dynamic system characterized by the following state and observation equations:

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k, \quad (56)$$

$$\mathbf{y}_{k+1} = \mathbf{C}_{k+1} \mathbf{x}_{k+1} + \mathbf{v}_{k+1}, \quad (57)$$

where $k = 0, 1, \dots$, $\mathbf{x}_k \in \mathbb{R}^n$ denotes the state vector, $\mathbf{y}_k \in \mathbb{R}^m$ denotes the observation vector, $\mathbf{w}_k \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^m$ denote the state and observation noise vectors, respectively, and matrices \mathbf{A}_k , \mathbf{B}_k , and \mathbf{C}_k are assumed to be known. In this model, random vectors \mathbf{w}_k and \mathbf{v}_k are mutually uncorrelated jointly Gaussian white noise sequences with known positive semi-definite covariance matrices \mathbf{Q}_k and \mathbf{R}_k , respectively. Furthermore, \mathbf{u}_k is a known control vector, the initial state vector \mathbf{x}_0 follows a multivariate Gaussian distribution, which is uncorrelated with \mathbf{w}_k and \mathbf{v}_k .

For each $k = 0, 1, \dots$, Kalman filter provides the best linear state estimator $\hat{\mathbf{x}}_{k+1|k+1}$ for the system state \mathbf{x}_{k+1} (at time step $k + 1$) in terms of the measurement sequence $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k, \mathbf{y}_{k+1}\}$. The following set of equations constitute the Kalman filtering algorithm:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{G}_{k+1}(\mathbf{y}_{k+1} - \mathbf{C}_{k+1}\hat{\mathbf{x}}_{k+1|k}), \quad (58)$$

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}_k \hat{\mathbf{x}}_{k|k}, \quad (59)$$

$$\mathbf{G}_{k+1} = \Sigma_{k+1|k} \mathbf{C}_{k+1}^T \left(\mathbf{C}_{k+1} \Sigma_{k+1|k} \mathbf{C}_{k+1}^T + \mathbf{R}_{k+1} \right)^{-1}, \quad (60)$$

$$\Sigma_{k+1|k} = \mathbf{A}_k \Sigma_{k|k} \mathbf{A}_k^T + \mathbf{Q}_k, \quad (61)$$

$$\Sigma_{k+1|k+1} = \left(\mathbf{I} - \mathbf{G}_{k+1} \mathbf{C}_{k+1} \right) \Sigma_{k+1|k}, \quad (62)$$

for $k = 0, 1, \dots$, where $\hat{\mathbf{x}}_{0|0} = \mathbf{m}_{\mathbf{x}_0}$ denotes the mean of the multivariate Gaussian distribution model, $\Sigma_{k+1|k}$ represents the prediction error covariance matrix and $\Sigma_{k+1|k+1}$ denotes the filtering error covariance matrix. Figure 16 illustrates the system model along with the standard KF. Due to the wide applicability of the Kalman filter, various extensions, including the Extended Kalman filter (EKF) [255] and the Unscented Kalman filter (UKF) [338], have been proposed in the literature and accommodate more sophisticated and/or nonlinear discrete-time dynamic systems. Furthermore, in an effort to improve performance, Kalman smoothers, which utilize future measurements to provide better state estimates, have been devised for linear and nonlinear systems following similar principles. Detailed descriptions and a survey of different family of Kalman filters can be found in [255] and [286].

Zois *et al.* [27] use a Partially Observable Markov Decision Process (POMDP) formulation in addition to Kalman-like filters and smoothers to detect the activity of individuals; the data from the individuals is acquired from a WBAN (A mobile phone, three accelerometers and one ECG sensor). In their application, they show that their proposed approach achieves a detection accuracy as high as 87%, while smoothing can improve the achieved accuracy by an additional 2%.

Messer *et al.* [339] develop a predictive low glucose suspend (PLGS) system coupled with an artificial pancreas technology that reduces hypoglycemia. Their system takes the

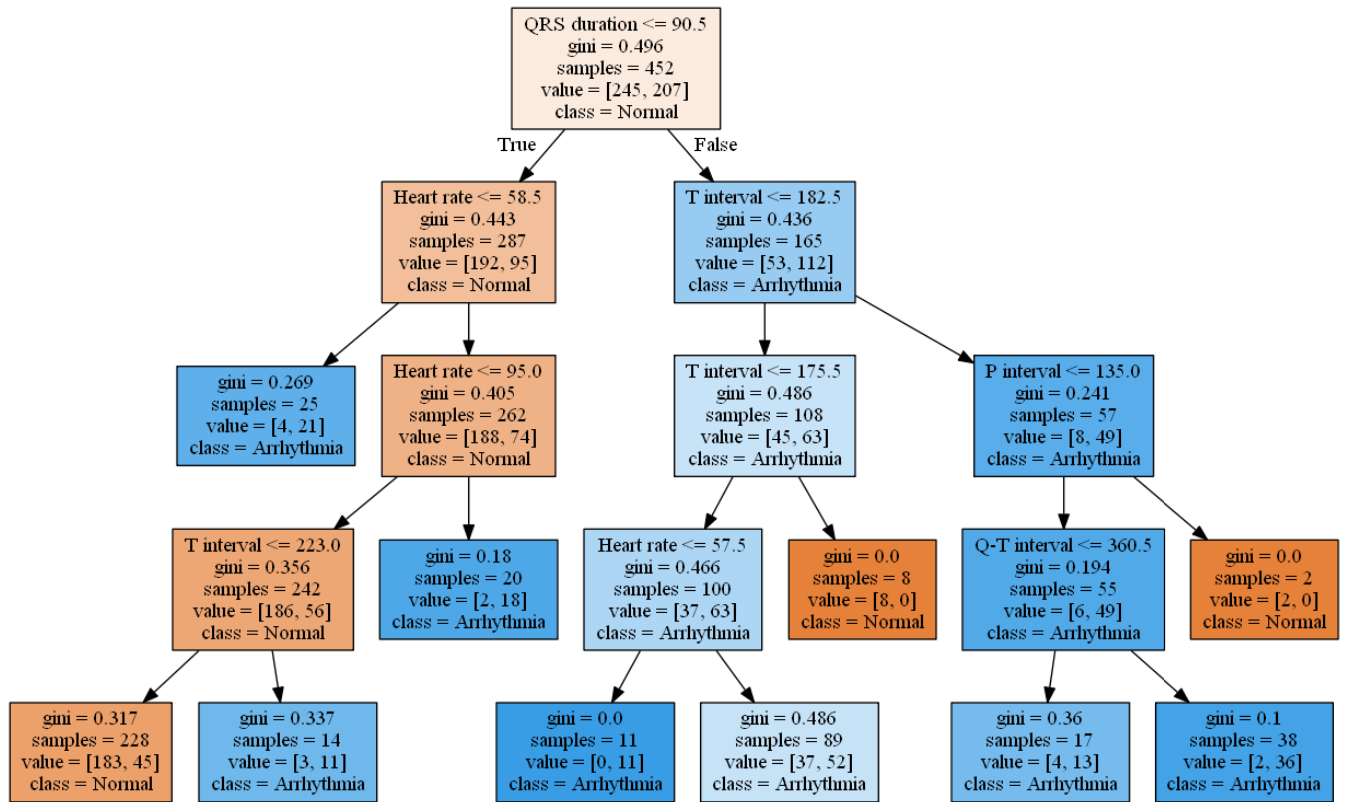


FIGURE 17. An example decision tree that classifies subjects heart beats into normal and arrhythmia based on certain features extracted from their ECG signals.

data from a continuous glucose monitor and uses a Kalman filter to predict low glucose. The authors’ goal is to analyze the behavioral differences among different age groups on the number of times they check their blood glucose levels during night time while using this PLGS system. They test their system in a clinical trial and report that the younger age groups tend to have more blood glucose checks and boluses.

Chen *et al.* [340] study personalized medicine by developing a Markov Decision Process (MDP) model that chooses appropriate therapy for breast cancer patients based on their response to previous treatments. Their model suggests to change medication in patients based on the effectiveness of their previous treatment. Their primary goal is to choose among four possible hormones for therapy instead of chemotherapy and assign rewards to the models that chooses the hormones over chemotherapy that leads to positive outcomes and reduced chemotherapy side effects. To find a solution for their model, they use Kalman Filters and compare their result with a standard breast cancer treatment. They show that their model results in higher rewards compared to standard treatment and has a much higher clinical benefit rate ($\approx 90\%$ compared to $\approx 80\%$).

C. DECISION TREES

Decision Trees [341] (DTs), also known as “Classification and Regression Trees (CART)” [222], create a tree structure based on the input data. This tree splits the data according

to different attributes at each level in a sense that the most discriminative features are placed closer to the root of the tree and the search is fine-tuned as it progresses through the branches. An example of decision tree algorithm is shown in Fig. 17 where the algorithm is applied to the heart arrhythmia database in [3]. The algorithm takes variables such as QRS duration, heart rate, T interval, P interval, and QT interval and classifies the subjects into two classes of subjects with normal heart beat and the subjects with heart beat arrhythmia. Note that Fig. 17 is simplified as in all the different types of arrhythmia are bundled together and the *depth* of the tree is fixed to 4.

One of the main benefits of DTs is that they are readable by humans. Some of the most utilized decision trees are Iterative Dichotomiser 3 (ID3) [342], c4.5 [343], c5.0, and Chi-squared Automatic Interaction Detection (CHAID).

The study in [20] utilizes DTs to predict the possibility of hypertension in people by taking biometric variables, lifestyle variables, and demographic variables as input. Authors use C5.0 and CHAID algorithms in addition to other machine intelligence algorithms to achieve their goal. In their application, CHAID yields the highest accuracy with a predictive rate of 64%.

D. ENSEMBLE ALGORITHMS (META-ALGORITHMS)

Ensemble algorithms are methods that use multiple other algorithms to achieve higher prediction accuracy.

Ensemble algorithms also reduce the chances of overfitting in machine intelligence algorithms.

1) RANDOM FORESTS

Random Forests is a meta-algorithm that utilizes multiple decision tree algorithms [344]. For a given dataset with high dimensional inputs, a random forest creates multiple subsets with lower dimensional inputs. These subsets are used to create decision trees separately and for a new entry, each decision tree makes an independent decision. Random forests take the output of all the smaller decision trees and make a final decision based on a majority vote. Ramon *et al.* [345] use First Order Random Forest (FORF) [346] to predict the survival of intensive care unit patients. They compare the results obtained from this algorithm to other algorithms such as decision trees and naive Bayes. They show that FORF has an accuracy of 82% when it is used to predict survival of a patient after one day in ICU and has comparable performance compare to other algorithms which have prediction accuracies of 79% to 88%.

Hijazi *et al.* [10] show that Random Forests and SVMs are among the most successful machine learning techniques to identify cardiac hazards (excluding Deep Learning Networks). While SVMs are good at identifying the most useful features of an ECG, Random Forests are good at classifying ECGs into certain cardiac hazard categories.

2) BOOSTING ALGORITHMS

Boosting algorithms are meta-algorithms that work with many *weaker* classifier algorithms. They take the output of a set of weaker learners into consideration and create a final output that is a *stronger* learner. One of the commonly used boosting algorithm is Adaptive Boosting (AdaBoost) [347]; in AdaBoost, the best predictive algorithm is chosen and then new algorithms are trained by putting more emphasis on the data points that are mis-predicted by previous algorithms. After training all of the algorithms, AdaBoost votes on the output of these algorithms and makes a final prediction. Mozos *et al.* [121] develop a stress detection system using AdaBoost as one of their classifier algorithms. They work with 18 subjects and achieve accuracies between 87% and 99% by using the AdaBoost algorithm.

E. ASSOCIATION RULE MINING

Association Rule Analysis [8] is designed to explore the entries of a database to find the features that appear in the data entries more *frequently*. For example, Adverse Event (AE) databases are composed of pairs of drugs and AEs, where each pair corresponds to a drug —taken by an individual— and an AE that has followed, such as a heart-burn or a heart attack. An example of this database looks like: {{Drug I, AE I}, {Drug II, AE II}, {Drug I, Drug II, AE I}, {Drug I, Drug III, AE III}, {Drug I, Drug II, AE I}, {Drug II, Drug III, AE II, AE IV}, ...}.

In this specific set, the following associations are denoted:

- {Drug I, AE I} means that Drug I is associated with Adverse Event I,
- {Drug II, AE II} means that Drug II is associated with Adverse Event II,
- {Drug I, Drug II, AE I} means that taking Drug I and Drug II together is associated with the occurrence of Adverse Event I,
- {Drug II, Drug III, AE II, AE IV} means that taking Drug II and Drug III together is associated with the occurrence of two Adverse Events, AE II and AE IV.

Association rule mining algorithms search these databases to find the data entries that appear to be correlated to each other, where in the case of AE databases, the analysis may yield the results that Drug VI and AE I appear to be correlated and show up in the data entries as pairs. There are multiple algorithms that are used for this analysis, where the *Apriori* algorithm is one of the most frequently used ones. Harpaz *et al.* [348] use the Apriori algorithm to analyze the FDA AERS (Adverse Event Reporting System) database. One of the issues with studying relations between drugs and adverse events is that these relations are only studies for single drugs and single adverse events; it is not possible to study all of the drug combinations and determine whether these combinations result in adverse events in clinical trials. The goal of this study [348] is to uncover links between unknown multi-item drug combinations and adverse events. Apriori algorithm helps the authors to analyze all drugs-AE pairs and single out some previously unknown multi-item AE associations. Note that this study does not prove that the AEs are the result of the drug combinations; rather, it narrows down the possible suspects for further analysis. For example, 67% of the drug-AE interactions —that the algorithm discovers— are already known to the medical society, although the algorithm generates another 33% as candidates for further study. While some of this newly-discovered 33% may be pure false-positives, some important interactions can also be discovered. In addition to drug-event associations, authors also discover drug-drug associations; although 91% of the drug-drug interactions are already known (78% due to being commonly prescribed together, 4% known drug-drug interactions, and 9% due to confounding), a new set of 9% is added to the repertoire of “suspect drug-drug interactions,” which may yield to useful new discoveries.

F. k-MEANS ALGORITHM

K-means algorithm is a clustering algorithm with the goal of clustering n observations into k clusters, where each observation is a member of a cluster with the closest mean. A sample result of using this algorithm is shown in Fig. 18, in which a 2 dimensional input space (x_1 and x_2) —with 3 possible outputs— is clustered using the k-means algorithm.

Sanchez-Morillo *et al.* [349] utilize the k-means algorithm in the prognosis of patients with Chronic Obstructive Pulmonary Disease (COPD). They focus on the early detection

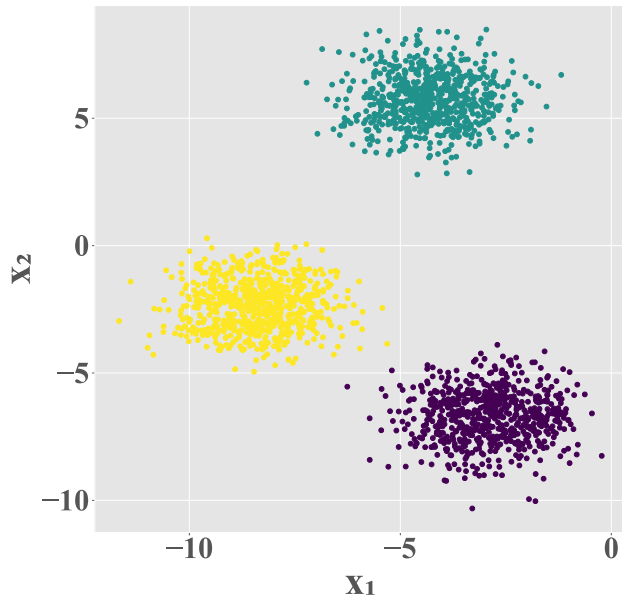


FIGURE 18. A depiction of the k-means algorithm where the input space has two axes, x_1 and x_2 , and there are three possible output classes, each shown by a different color.

of Acute Exacerbation of Respiration Symptoms (AECOPD); to achieve their goal they design a mobile health system questionnaire that asks the participants of the study about their mood and medical records on a daily basis. k-means algorithm takes this information and clusters them into different classes, where each class represents a different level of severity of symptoms. The output of the classifier is the used to predict exacerbation by checking if the symptoms are worsening for at least two consecutive days. Within 15 participants in the study, exacerbation is predicted with a 84.7% accuracy.

G. DYNAMIC PROGRAMMING

Dynamic Programming (DP) is a mathematical optimization method of solving a complex optimization problem using a simple three-step approach: (i) divide the original problem into a set of simpler sub-problems in a recursive manner, (ii) solve each of the sub-problems, and (iii) determine the global solution by carefully combining the solutions of the sub-problems. In comparison to a greedy algorithm, which selects the locally optimal solution that is not necessarily globally optimal, DP guarantees that the acquired solution will be globally optimal. We underscore that a problem needs to exhibit the optimal sub-structure property (i.e., an optimal solution to the problem can be constructed efficiently from optimal solutions of its sub-problems) to be able to use DP to solve for the optimal solution.

Problems that involve decisions over time can often be divided into smaller sub-problems and solved recursively. In fact, dynamic decision problems formulated using the sequential decision-making (SDM) models exhibit this structure and can be solved recursively via DP; examples of SDM models are Markov Decision Processes (MDPs), Partially Observable MDPs (POMDPs), and Multi-Armed

Bandits (MABs). The main idea is to divide a multi-period decision problem into a sequence of decision steps over time (Bellman's "Principle of Optimality" [183], [350]). This is achieved by defining a sequence of value functions $\bar{J}_1(z_1), \bar{J}_2(z_2), \dots, \bar{J}_n(z_n)$, where $z_k, k = 1, 2, \dots, n$ represents the information available at time step k based on which a new decision will be made. The value function at each time step k can be determined by working backwards in time (i.e., from the value function at time step $k + 1$) using the DP equation (also known as Bellman equation). This is usually achieved by optimizing a simple function (e.g., sum, product) of the gain of a specific decision at time step k and the function $\bar{J}_{k+1}(z_{k+1})$ at the new information state if this decision is made. Finally, $\bar{J}_1(z_1)$ at the initial information state corresponds to the value of the optimal solution, while the associated optimal decision variable values can be extracted by tracing back the set of performed calculations.

1) MDP STRATEGY

The optimal decision strategy of the MDP formulations given in Section VIII-D can be easily determined following the above procedure. In particular, we can recursively compute the value function for all states $x_k \in \mathcal{X}$ for an arbitrary finite horizon T as shown in Eq. (63), as shown at the bottom of the next page for $k = 0, 1, \dots, T - 1$, where $\bar{J}_k(\cdot)$ is the value function for time step k , $P(x_{k+1}|x_k, u_k)$ denotes the transition probability of moving from state x_k to state x_{k+1} when control action u_k is executed, $c_k(x_k, u_k)$ the reward/cost at time step k associated with state x_k and control action u_k and $\gamma \in (0, 1]$ is a discount factor that weighs the contribution of different states differently in time. The optimal control action for state x_k is shown in Eq. (64), as shown at the bottom of the next page.

In the case of infinite horizon, the optimal value function satisfies the fixed-point equation shown in Eq. (65), as shown at the bottom of the next page where $P(x'|x, u)$ denotes the transition probability of moving from state x to state x' when control action u is executed, and $c(x, u)$ represents the reward/cost associated with state x and control action u . To solve the above fixed-point equation, techniques such as value iteration [351], policy iteration [352], and linear programming [183] are usually employed.

The study presented in [288] uses an MDP formulation for the design of adherence-improving interventions for the cardiovascular disease management of 54,036 patients with type 2 diabetes based on individual electronic health records. The study focuses on adherence to *statin* (the most common medication for lowering cholesterol) treatment. Their Active Adherence Surveillance (AAS) system, which is created by the MDP algorithm, is compared to an Inactive Adherence Surveillance (IAS) system. AAS observes the patients and assigns a health state to them during consecutive *epochs*. These states are *adherence states*, showing the level of patient's adherence to their medication, and *absorption state*, showing the occurrence of the event the medication was supposed to prevent. The decision maker is supposed to make

a decision about intervening with the patient at each epoch. The algorithm is designed to calculate the probability of an event happening during an epoch and provide decision support for intervention. The study shows that AAS increases the life expectancy years of a male individual by 0.19 years and by 0.17 years a female individual. Furthermore, AAS reduces the cost of intervention, statin treatment, and hospitalization care for patients with cost savings of \$1800 for male and \$1700 for female patients.

2) POMDP STRATEGY

In POMDPs, the actual system state is hidden; although the same dynamic programming principles apply, the related equations are a function of the *belief state*. In particular, the value function \bar{J}_k in this case has the form shown in Eq. (66), as shown at the bottom of this page where \mathbf{p}_k represents the belief state at time step k , $\mathbf{c}(u_k) = [c(1, u_k), \dots, c(|\mathcal{X}|, u_k)]^T$ is the reward/cost vector at time step k associated with control action \mathbf{u}_k , $\mathbf{1}_{|\mathcal{X}|}$ is a column vector with $|\mathcal{X}|$ ones, $\mathbf{P}(u_k)$ is the transition probability matrix, $\Delta(y_{k+1}, u_k) = \text{diag}(f(y_{k+1}|1, u_k), \dots, f(y_{k+1}||\mathcal{X}|, u_k))$ is the diagonal matrix of observation probabilities, and $\Phi(\cdot)$ is a function that captures the evolution of the belief state over time and is described according to update rule shown in Eq. (67), as shown at the bottom of the next page. The optimal control action for belief state \mathbf{p}_k then follows Eq. (68), as shown at the bottom of this page.

The main challenge of solving a POMDP via dynamic programming is that the belief state is uncountably infinite. Fortunately, the associated value functions are piecewise linear and convex/concave [353], enabling us to determine the optimal decision strategy in finite time. A significant amount of research effort has been made in developing efficient methods for solving POMDPs [354]. Current state of

the art (e.g., point-based approaches [355], sampling techniques, and problem structure exploitation) has made the solution of POMDPs with millions of states computationally feasible [356].

Hoey *et al.* [21] study a POMDP formulation to track an individual's upper-limb reaching rehabilitation progress over time and adjust the level of difficulty based on their current abilities. Their system couples a POMDP model to a haptic robotic device and the task for the patient is reaching a target in a virtual game. The system takes inputs such as the time it takes for the patient to reach the target and the posture of the patient to adjust the goal of the game and issues break periods for the patient. Although the POMDP-based algorithm issues a higher number of break periods, its target distance decision agrees with the therapists 94% of the time.

3) MAB STRATEGY

Irrespective of the characteristics of a particular MAB formulation, the associated optimization problem can be solved using DP. For instance, in the case of a stochastic MAB problem, the DP recursion takes the form shown in Eq. (69), as shown at the bottom of this page where $r(\cdot)$ represents a bounded reward function, $x(u)$ denotes the state of arm u , K is the number of different arms (options), and $P_{x(u),k}$ denotes the transition probability associated with the state of arm u at time step k . In 1979, Gittins showed that the optimal decision strategy for this problem is equivalent to selecting the arm with the highest Gittins index at each time step [357]:

$$v(x_0(i)) = \max_{\tau > 0} \frac{\mathbb{E} \left\{ \sum_{k=0}^{\tau-1} \gamma^k r(x_k(i)) \middle| x_0(i) \right\}}{\mathbb{E} \left\{ \sum_{k=0}^{\tau-1} \gamma^k \middle| x_0(i) \right\}}, \quad (70)$$

$$\bar{J}_k(x_k) = \min_{u_k \in \mathcal{U}} \left[\sum_{x_{k+1} \in \mathcal{X}} P(x_{k+1}|x_k, u_k) c_k(x_k, u_k) + \gamma \sum_{x_{k+1} \in \mathcal{X}} P(x_{k+1}|x_k, u_k) \bar{J}_{k+1}(x_{k+1}) \right], \quad (63)$$

$$\bar{\mu}_k(x_k) = \arg \min_{u_k \in \mathcal{U}} \left[\sum_{x_{k+1} \in \mathcal{X}} P(x_{k+1}|x_k, u_k) c_k(x_k, u_k) + \gamma \sum_{x_{k+1} \in \mathcal{X}} P(x_{k+1}|x_k, u_k) \bar{J}_{k+1}(x_{k+1}) \right]. \quad (64)$$

$$\bar{J}(x) = \min_{u \in \mathcal{U}} \left[\sum_{x' \in \mathcal{X}} P(x'|x, u) c(x, u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x, u) \bar{J}(x') \right], \quad (65)$$

$$\bar{J}_k(\mathbf{p}_k) = \min_{u_k \in \mathcal{U}} \left[\mathbf{p}_k^T \mathbf{c}(u_k) + \gamma \sum_{y_k \in \mathcal{Y}} \mathbf{1}_{|\mathcal{X}|}^T \Delta(y_{k+1}, u_k) \mathbf{P}(u_k) \mathbf{p}_k \bar{J}_{k+1}(\Phi(\mathbf{p}_k, u_k, y_{k+1})) \right], \quad (66)$$

$$\mathbf{p}_{k+1} = \frac{\Delta(y_{k+1}, u_k) \mathbf{P}(u_k) \mathbf{p}_k}{\mathbf{1}_{|\mathcal{X}|}^T \Delta(y_{k+1}, u_k) \mathbf{P}(u_k) \mathbf{p}_k}. \quad (67)$$

$$\bar{\mu}_k(\mathbf{p}_k) = \arg \min_{u_k \in \mathcal{U}} \left[\mathbf{p}_k^T \mathbf{c}(u_k) + \gamma \sum_{y_k \in \mathcal{Y}} \mathbf{1}_{|\mathcal{X}|}^T \Delta(y_{k+1}, u_k) \mathbf{P}(u_k) \mathbf{p}_k \bar{J}_{k+1}(\Phi(\mathbf{p}_k, u_k, y_{k+1})) \right]. \quad (68)$$

$$\bar{J}(x(1), \dots, x(K)) = \max_{u \in \{1, \dots, K\}} \left[r(x(u)) + \gamma \sum_{k=1}^{\infty} P_{x(u),k} \bar{J}(x(1), \dots, x(u-1), k, x(u+1), \dots, x(K)) \right], \quad (69)$$

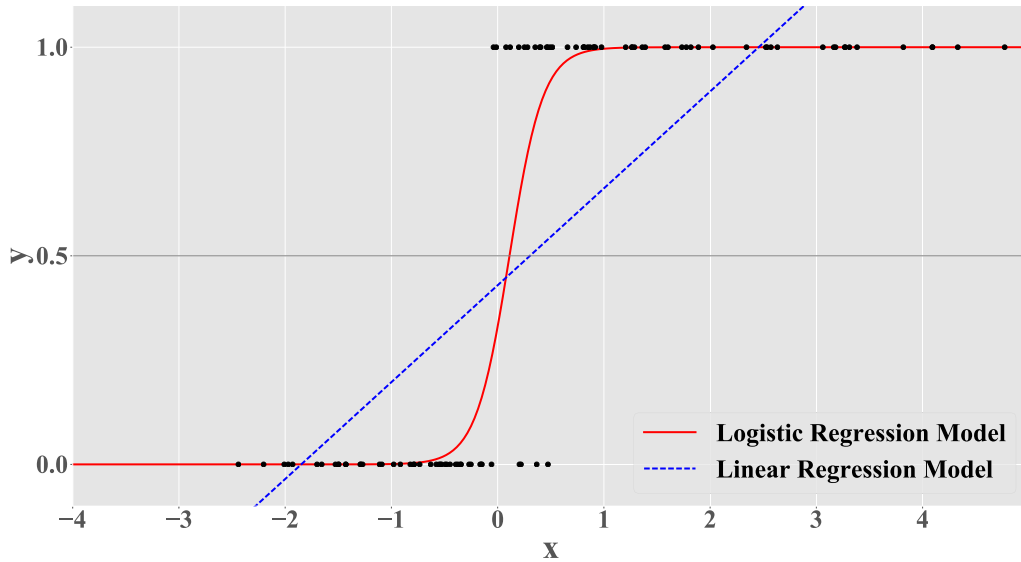


FIGURE 19. A comparison of fitting a classifier to a sample dataset using linear regression and logistic regression. The x-axis is the input space and the y-axis is the output, which can take two possible values (0 and 1). It is apparent from the figure that a logistic regression model suits this relationship better than linear regression, due to the sharp transition of the output from 0 to 1.

where $x_0(i)$ denotes the initial state of arm i . This suggests that the optimal decision strategy has a very efficient implementation using the following three steps: (i) determine $v(x_0(i))$, $i = 1, 2, \dots, K$, (ii) pull arm $\bar{\mu} = \arg \max_i v(x_0(i))$ until the minimum time that the maximum value in Eq. (70) is achieved, and (iii) repeat this process indefinitely.

Rabbi *et al.* [26] design a system named MyBehavior; they use a MAB formulation to infer an individual's physical activity and dietary behavior and suggest changes that can lead to a healthier lifestyle. The MyBehavior system takes automatic sensing and manual user input information as raw data. It includes 800 categories of activities and has over 8000 food items, which the user manually logs. The adopted MAB algorithm analyzes these inputs and suggests behavior changes for a healthier lifestyle. This approach is specifically useful, because it observes the past frequent calorie loss patterns to suggest eliminating less effective calorie loss behaviors; the goal of the algorithm is to have the user adopt these more effective patterns. The system was deployed for 14 weeks with 16 users and showed that on average, users followed 1.2 suggestions per day, walked 10 more minutes per day, burnt 42 more calories by exercise per day, and reduced their calorie intake by 56 points per meal.

H. LOGISTIC REGRESSION

Logistic Regression [358] is a classification model that provides a “probability” of a given data point belonging to a specific class. It is especially useful when a regression problem has a dichotomous (binary) dependent variable; in this case, linear regression models fail in creating good boundaries for classification. An example of how this method is used is shown in Fig. 19 where the x axis shows the input and the y axis is for the output which can take two possible values,

0 and 1. It is apparent from Fig. 19 that the logistic regression model can perform a better classification, as compared to a linear regression model.

Logistic regression follows Eq. (71) where x is the input and y is the output that can take two possible values; α and β are the components that are learned and dictate the shape of the curve. In this specific case x is only one dimensional.

$$y = \frac{1}{1 + e^{-\alpha x + \beta}} \quad (71)$$

Neuvirth *et al.* [17] study the probability for future emergency care need for a diabetic patient Their features includes lab test results, medicine, diagnosis, and medical procedures and they use LR in addition to other methods such as kNN algorithm to identify the patients-at-risk and predict their future visit to hospitals. Logistic regression achieves the highest c-index and AUC as a stand-alone algorithm with a c-index of 0.666 and an AUC of 0.713. Another study utilizing logistic regression is [28], where the authors develop a mood detection system. Authors feed various types of input data (Head movement, eye blink, pupil radius, user interactions with computers, etc.) to a logistic regression algorithm, as well as other classifiers, and classify the mood of subjects into one of three distinct states: {Positive, Neutral, Negative}. The participants of the study are 27 users and many metrics for this classification scheme are reported. F-1 score and AUC of the three states are reported as 0.84 and 0.95 for the {Negative} state, 0.59 and 0.79 for the {Neutral} state, and 0.71 an 0.91 for the {Positive} state.

I. NAÏVE BAYES

Naïve Bayes methods are algorithms that are based on the Bayes theorem and have a *naïve* assumption that features are independent from each other. This algorithm calculates the

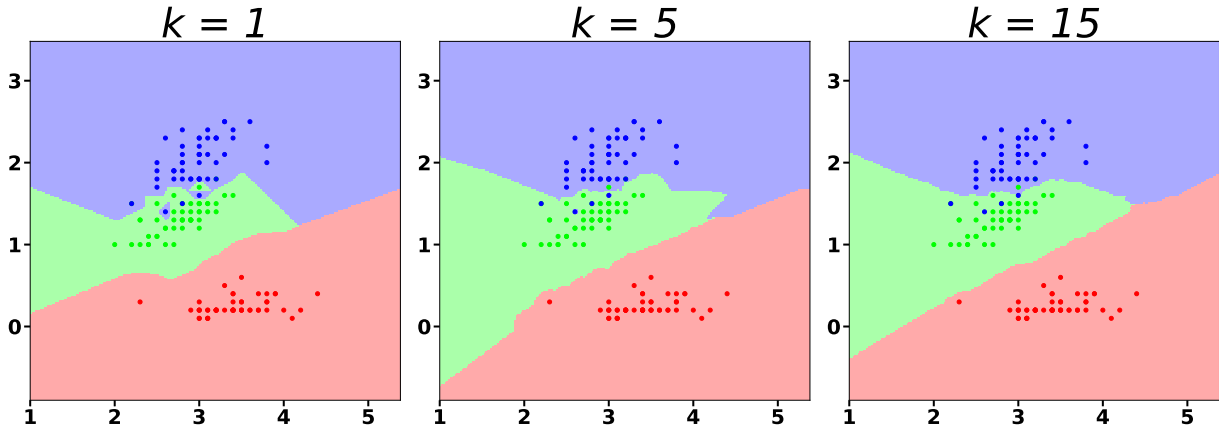


FIGURE 20. Multiple examples of classifying data points, using the k-Nearest Neighbors (kNN) algorithm. The data points on the plots are the input data points that belong to three different classes. The input space is covered by the horizontal and vertical axes and the output space (i.e., the classification) is shown by the color of each data point. The difference between the plots is the in the number of neighbors (k) that were considered for classification; based on the number of neighbors (k) considered for each point ($k = \{1, 5, 15\}$ in this specific case), the boundaries of the classes change.

probability of a given point belonging to a class based on the conditional probability of inputs given the output. The general Naïve Bayes model is given in Eq. (72) where the input vector (\vec{x}) has d dimensions and y is the output.

$$\Pr(y|\vec{x}) = \frac{\Pr(\vec{x}|y)\Pr(y)}{\Pr(\vec{x})} = \frac{\Pr(y) \prod_{i=1}^d \Pr(x_i|y)}{\Pr(\vec{x})} \quad (72)$$

Assuming that the output can take two possible values (0 and 1), we can decide that which value of y is more probable by using Eq. (72) and calculating the ratio in Eq. (73) for a given input. All of the values on the right side of Eq. (72) are calculated using the training set.

$$\frac{\Pr(y=0|\vec{x})}{\Pr(y=1|\vec{x})} = \frac{\Pr(y=0) \prod_{i=1}^d \Pr(x_i|y=0)}{\Pr(y=1) \prod_{i=1}^d \Pr(x_i|y=1)} \quad (73)$$

Using a wireless chest belt (reading ECG and respiration) and a hand sensor for skin conductance and EMG signals, Wijsman *et al.* [29] classify the mental condition of a patient into two possible categories: {**Stress, Non-stress**}. They use multiple types of classifiers, but the one with the least amount of error rate is Linear Bayes Normal with an error rate of 0.21.

J. k-NEAREST NEIGHBORS (kNN)

The k-Nearest Neighbors (kNN) algorithm [359] is a classification and regression algorithm that classifies the new input data by considering the previously-observed neighboring points of it. An example of the kNN algorithm is shown in Fig. 20, where there are 3 classes for the data points, denoted by three different colors. kNN algorithm is executed for the dataset by setting “ k ” to be 1 (a single neighbor), 5 (the datapoint and 5 of its neighbors), and 15; the new data

points that are fed into the algorithm are classified based on the shade of the color that the data entry falls on for each one of these cases. Addition of each data point makes the boundaries between pairs of classes sharper.

Lan *et al.* [360] use kNN and 2 other classifiers on EEG signals to estimate the cognitive state of subjects between 4 activities of {**slow walking, navigating and counting, communication with radio, studying mission map**}. They implement a majority vote of these classifiers to finalize their decision and achieve an 80% accuracy on their classification scheme.

K. LIKELIHOOD RATIO TEST

A Likelihood Ratio Test is a statistical hypothesis test used to determine whether a single hypothesis —out of a number of mutually exclusive alternative hypotheses— is true or not [255], [257], [361], [362]. In a typical case, we have access to a random observation vector \mathbf{y} and we want to choose among possible hypotheses; each hypothesis is described by an *a priori probability* denoted by P_i . Under each hypothesis H_i , the observation vector is probabilistically described by a known and well-defined probability distribution denoted by $P(\mathbf{y}|H_i)$. The goal is to maximize the probability of a correct decision, which is achieved by choosing the hypothesis based on maximum a posteriori (MAP) probability rule [363] as follows:

$$\hat{H} = \arg \max_i [P(H_i|\mathbf{y})], \quad (74)$$

where $P(H_i|\mathbf{y}) = \frac{P(\mathbf{y}|H_i)P_i}{P(\mathbf{y})}$. In the simple two-hypothesis case, this is equivalent to the following comparison:

$$\frac{P(\mathbf{y}|H_1)P_1}{P(\mathbf{y})} \stackrel{?}{\geq} \frac{P(\mathbf{y}|H_0)P_0}{P(\mathbf{y})}, \quad (75)$$

which indicates that H_1 holds if the left-hand-side is greater than or equal to the right-hand side and similarly for H_0

(ties are broken arbitrarily). Rearranging terms above, we get:

$$L(\mathbf{y}) \triangleq \frac{P(\mathbf{y}|H_1)}{P(\mathbf{y}|H_0)} \geq \frac{P_0}{P_1} \triangleq \eta, \tag{76}$$

where $L(\mathbf{y})$ is referred as the *likelihood ratio* and η is the threshold. Thus, the above comparison is known as the likelihood ratio test. In the multiple hypothesis case, we perform a set of binary threshold comparisons to reach a decision, i.e., for all $i, j, j > i$:

$$\frac{P(\mathbf{y}|H_j)}{P(\mathbf{y}|H_i)} \geq \frac{P_i}{P_j}. \tag{77}$$

In many situations, the cost of a wrong decision is highly asymmetric; in such a case, only the threshold η is affected and the associated test is called a Bayes test. For a detailed description of the likelihood ratio test and other related tests, the interested reader is referred to [255], [257], [361], and [362].

One example application of this algorithm is in [364], where authors predict hospitalizations due to heart diseases. The authors take the available electronic health records of patients and predict whether they will be hospitalized in the coming year. They test and compare many algorithms such as SVMs, logistic regression, and likelihood ratio test and provide ROC curves for these classifiers. They also demonstrate the features that LRT takes as significant and non-significant features; their study reports the number of emergency room visits in the previous year as “significant” and sex as “non-significant” factors.

L. SCORING SYSTEMS

In medical applications, scoring systems are classification models that help physicians make a quick risk prediction for a medical condition just by adding and subtracting the values of some physiological input parameters. The main benefit of these systems is that they do not need extensive training or a computer to be calculated. There are many scoring systems in use today such as *SAPS III* [365], which predicts the mortality of ICU patients, *QRISK2* [366] which is a prediction score for cardiovascular diseases, and *Eagle score* [367] which gives a probability for a patient dying during heart surgery.

Historically, scoring systems are developed but by experts, rather than machines. Recently, machine intelligence algorithms have been employed to develop new scoring systems in an automated fashion. An example is *Super-sparse Linear Integer Model (SLIM)* [368], which is a machine learning method for creating scoring systems. One example of using this algorithm is shown in [141] where the authors develop a scoring system for sleep apnea detection using SLIM. The authors propose a medical scoring system with 5 physiological input parameters, where the score is 0 initially and progressively increases/decreases as each

parameter is —potentially— added:

$$\text{Size-5 SLIM Score for Sleep Apnea} = \begin{cases} \text{add +4,} & \text{Age} \geq 60 \\ \text{add +4,} & \text{Hypertension} \\ \text{add +2,} & \text{BMI} \geq 30 \\ \text{add +2,} & \text{BMI} \geq 40 \\ \text{add -6,} & \text{Female.} \end{cases} \tag{78}$$

A cumulative score of greater than 1 indicates the presence of obstructive sleep apnea with False Positive Rate (FPR) of 20%. The authors also develop a more sophisticated scoring algorithm with 10 physiological input parameters including information such as smoking habits, diabetes patients (again, the score is 0 if none of the 10 parameters have been added):

$$\text{Size-10 SLIM Score for Sleep Apnea} = \begin{cases} \text{add +16,} & \text{Age} \geq 30 \\ \text{add +12,} & \text{Age} \geq 60 \\ \text{add +12,} & \text{BMI} \geq 25 \\ \text{add +2,} & \text{BMI} \geq 30 \\ \text{add +10,} & \text{BMI} \geq 35 \\ \text{add +4,} & \text{BMI} \geq 40 \\ \text{add +6,} & \text{Diabetes} \\ \text{add +4,} & \text{Hypertension} \\ \text{add +2,} & \text{Smoker} \\ \text{add -14,} & \text{Female.} \end{cases} \tag{79}$$

A total cumulative greater than 29 indicates the presence of OSA with an FPR of 20% and True Positive Rate (TPR) of 65%.

XI. ARTIFICIAL NEURAL NETWORKS

The development of Artificial Neural Networks (ANNs) were inspired from the way any living organism performs its life functions using a sophisticated network of *neurons*; the neurons and a network of their connection through *synapses* forms a Central Nervous system, which performs three important functions: **(i)** collect and pre-process the data from sensory inputs in the body, **(ii)** process the collected data in a centralized location (the brain), and **(iii)** transmit the computed motor response back to the body to activate muscles, tentacles, or other parts of the body that can generate a motion.

ANNs and their highly-popular variant Deep Learning Networks have evolved based on this bio-inspired structure of neural nets; ANNs receive computer data and pre-process it followed by a processing network and output the result in an easily-interpretable form. In healthcare computing, they have found wide-spread use due to the unparalleled prediction performance they boast.

In this section, we will investigate their computational infrastructure of ANNs (Section XI-A) and their specific usage in healthcare applications (Section XI-B). We will

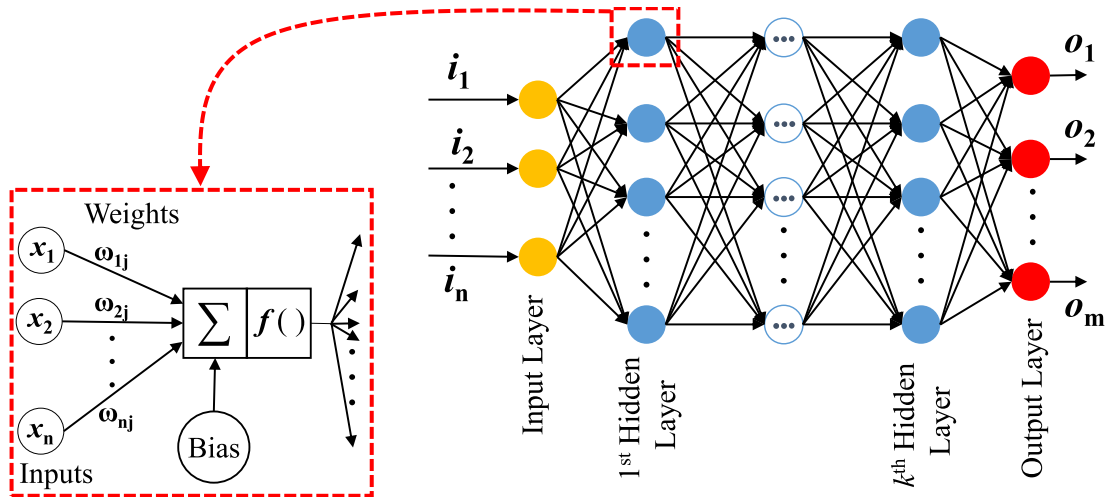


FIGURE 21. Common architecture of a fully-connected artificial neural network with n inputs, k hidden layers, and m outputs (on the right). Inner structure of a neuron used in artificial neural networks is shown on the left-side box; ω_{ij} are the weights by which inputs to the neuron (x_1, x_2, \dots, x_n) are multiplied before they are summed. “Bias” is a value by which this sum is augmented and $f()$ is the activation function, which is used to introduce a non-linear component to the output. A set of commonly used activation functions are tabulated in Table 2.

study different types of ANNs in the following subsections: Feed-forward Neural Networks (Section XI-C), Probabilistic Neural Networks (Section XI-C2), Radial Basis Function Networks (Section XI-C1), Deep Belief Networks (Section XI-D), Convolutional Neural Networks (Section XI-E), and Recurrent Neural Networks (Section XI-F).

A. COMPUTATIONAL STRUCTURE OF ANNS

As opposed to the stricter models studied in Section VII, an ANN can be thought of as being a highly-flexible model that is constructed through the connection topology of the neurons and the layers within the ANN. Because of their flexibility, ANNs are very popular in problems that do not have well-defined features, as opposed to the strict feature definitions that we investigated in Section V. The number of layers in a neural network is an indicator of the complexity of the model; problems that require complex models typically need deeper networks. Most ANN structures consist of an input layer, one or many hidden layers, and an output layer. A general structure of an artificial neural network is depicted in Fig. 21, which shows a network that has n neurons in its input layer, k hidden layers, and m outputs.

1) NEURONS

The basic building block of an ANN is a *neuron*. The inner structure of a neuron is shown in Fig. 21 (left), where the neuron collects data from multiple inputs and multiplies them by their assigned constant weights. The sum of these weighted inputs is then biased by a constant value (“Bias,” as shown in Fig. 21) and the result is passed through an “activation function.” The output of the neuron is determined by the result of the activation function.

2) ACTIVATION FUNCTIONS

A list of commonly used activation functions is provided in Table 2. The presence of activation functions is a necessity to introduce non-linearity in the functionality of a neuron and to a neural network in general. If the activation function is a linear function (e.g., the *identity function* in Table 2), the entire network becomes a linear combination of its inputs, which will not capture the nonlinear relations in the data.

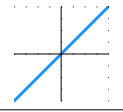

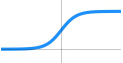
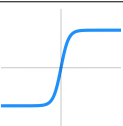
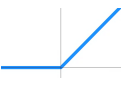
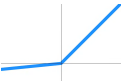
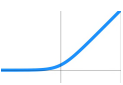
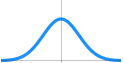
Logistic (Sigmoid) function introduces the non-linearity that is needed, while keeping the output range between 0 and 1. This output range-limiting property eliminates the danger of the internal values in the ANN spiraling out of control, which will cause convergence issues.

Step Function achieves the same non-linearity goal as the logistic function, however, the derivative of the step function is discontinuous at 0, making its use tricky in algorithms that rely heavily on derivatives.

Hyperbolic Tangent is very similar to Sigmoid. The only difference is that $\tanh()$ is an odd function and some research argues that odd functions converge faster when the network is being trained.

Rectifier Linear Unit (ReLU): ReLU activation function remedies some of the problems that arise in deeper networks, when activation functions with limits on their values are used. Activation functions such as the step function or sigmoid usually saturate when the number of layers increases; this means that adding new layers to the network increases the computational complexity of its training network without benefiting its performance, because the neurons in the early layers saturate. Introduction of ReLU solves the neuron-saturation problem in deep networks without increasing computational complexity, because it is very easy to calculate.

TABLE 2. Common activation functions for neurons, represented as $f()$ in Fig. 21.

Name	Equation	Plot
Identity	$f(x) = x$	
Step Function	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	
Logistic (Sigmoid)	$f(x) = \frac{1}{1 + e^{-x}}$	
Hyperbolic Tangent (tanh)	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \tanh(x)$	
Rectifier Linear Unit (ReLU)	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	
Parametric ReLU	$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	
SoftPlus	$f(x) = \ln(1 + e^x)$	
Radial Basis Function (RBF) - Gaussian	$\phi(X) = e^{-\beta \ X - \mu\ ^2}$	

Note that ReLU is still a non-linear function and it is sufficient to capture the nonlinearity in the data in most cases.

Parametric ReLU: Parametric ReLU is like ReLU activation function with addition of another parameter (α), which can be learned during the training phase. Having a non-zero output for negative input values proves useful for certain algorithms.

SoftPlus: SoftPlus function acts almost exactly the same way as ReLU, with the difference that it is differentiable for all the input values, as opposed to ReLU, which is not differentiable at $x = 0$.

Radial basis functions (RBFs) are a group of functions, whose values depend on their distance from some point c in the input space. In applications where the distance of the input to some given data points in the input space is important, RBFs are suitable candidates for activation functions of neurons. One of the most commonly used RBFs is the Gaussian form that is shown in Table 2.

B. CHARACTERISTICS OF ANN-BASED ALGORITHMS

ANNs include a large family of structures based on the way neurons are connected to each other and each of these structures has its unique characteristics. Different subcategories of ANNs share some common characteristics, despite being

suitable for different applications. In this section, we discuss these common characteristics, which involve how an ANN handles the input data and/or data features, as well as how the training is performed in the ANN.

1) HANDLING DATA

The dimensionality of the data determines the complexity of an ANN; increased dimensionality in the raw input data typically brings about the need to turn this raw data into features (see Section V). However, the mapping from raw data to features is not readily available for every application. One of the most exciting properties of ANNs is their ability to work with high dimensionality data for which the features are not known. To handle these different data dimensionality scenarios, ANNs could be categorized by purpose as follows:

- **Data Features as Input:** When the features are available in an application (for example, QT and RR intervals in an ECG signal), the required data input to an ANN is substantially lower, which requires much simpler ANNs. In these scenarios, the computational burden on the ANN is reduced due to existing knowledge of the application. Zhang et al. [369] create a pathological brain detection system that takes slices of brain images, extracts features from these images using a fractional Fourier transform and feed these features into an ANN to classify the brain images as pathological vs. healthy. In this specific example, the computational burden of feature extraction is shifted outside the ANN, because the readily-calculated features are the input to the ANN rather than the raw brain image data.
- **Raw Data as Input:** For cases where the data is complex and data features are not well-defined, an ANN can accept raw data as input, which will require the ANN to handle feature extraction internally. Intuitively, feature extraction implies dimensionality reduction on the data. Based on this insight, there doesn't even have to be a clear transition from raw data into features; each layer of the ANN can be taught of as being a dimensionality reducer and the entire network of the ANN is some representation of the data features. In this way, for higher dimensionality data, it is logical to expect a deeper ANN to be able to extract the features. The structural flexibility of an ANN allows it to be adapted to take image, video, audio, time series, and text data as input, with 1D, 2D, or 3D dimensions. An ANN input can be more than one type of input data (e.g. video and audio) or a combination of raw data and features. Huang et al. [370] study fetal cardiac screening videos using deep neural networks. The goal of the application is to find the angle of an ultrasound video and the location of the heart in the images. The deep neural network takes the raw ultrasound video as input, which is 3D, and is able to achieve near-human accuracies in both angle detection and heart localization.
- **Data Features as Output:** Some ANNs are used to extract features from raw data. These features can be

used as input to other algorithms that do not have the capability to work with raw data efficiently. For example, Zhang *et al.* [371] study emotion recognition in speech in a noisy environment; they use an autoencoder (which is a type of an ANN described in Section XI-D) to find and enhance the features of the input audio signal. The extracted features by the ANN are then fed into a conventional SVM to classify the emotion. They show that their enhanced features extracted by the autoencoders beat the baseline (which includes features extracted by standard digital signal processing techniques) in every case.

2) HANDLING TRAINING

Training of ANNs is typically achieved through a process called *backpropagation* using the *Gradient Descent* algorithm. Backpropagation examines the output error for a given input and propagates back from the output layers towards the input layers and by adjusting the parameters of the network to reduce the error. The number of parameters in an ANN may pose an overfitting problem, which can be overcome by using conventional as well as ANN-specific techniques. One case of the latter is the *dropout technique*, in which only half of the neurons are randomly selected and trained and the other half are ignored at each training step.

Training an ANN is usually a computationally intensive task and more complex forms of ANNs were not feasible to use until the emergence of high computation capability of GPUs in recent years [321]. The inherent parallelism of an ANN utilizes the GPU architecture in an efficient way in contrast to other general structure processing units, such as a CPU.

C. FEEDFORWARD NEURAL NETWORKS

In applications where the data is provided in a time-series, or speech recognition applications where the data consists of words in a sentence, references to previous and current data elements must be made for a neural network to produce its output. Alternatively, in a feedforward neural network, none of the connections of neurons loop back to previous ones, i.e., every neuron is front-connected; therefore, the output of a feedforward neural network at a given point in time depends only on a single input at that same time. A *fully connected* network (shown in Fig. 21) is a common architecture of feedforward neural networks, in which every neurons in one layer is connected to every neuron in the previous and next layer. Fully connected networks are best suited for shallow networks, because the output of the neurons that use sigmoid or hyperbolic tangent as their activation function saturate on deeper networks due to the limits on their output (see Section XI-A1), making the benefit from the added layers marginal or none. Furthermore, the number of parameters in a network grows rapidly, making deeper feedforward networks computationally expensive; this in turn makes training expensive and increases the probability of overfitting. Note that these problems are fixed in

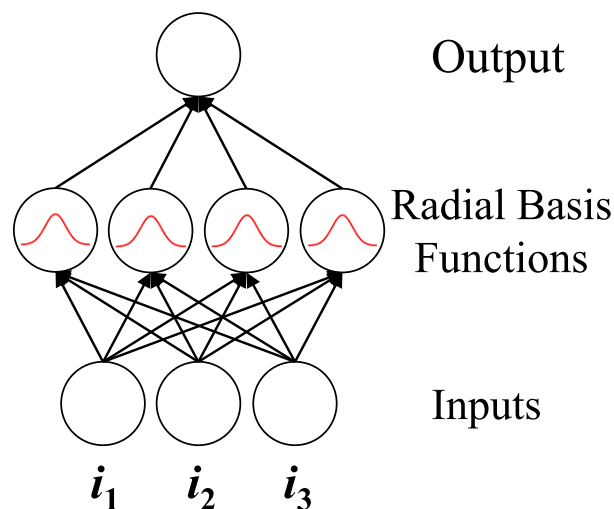


FIGURE 22. Structure of an RBF network with three input features and four neurons in the hidden layer using Gaussian Radial Basis Function as activation function.

convolutional neural networks, as will be investigated in Section XI-E.

An example application of feedforward neural networks is presented in [372], where authors design a hearing loss detection system for sensorineural hearing loss disease patients. This system extracts features from the brain tissue MRI images using signal processing algorithms. These features are then provided to a feedforward neural network with one hidden layer which detects sensorineural hearing loss with a 95% accuracy. Another application that uses a simple fully connected neural network is shown in [373], where the possibility of adverse events happening to ICU patients after their admission is studied. Input features to the network include patients' vital signs and laboratory results; the output of the network is a prediction of whether the patient will develop an adverse event in the next 4 hours or not. The system achieves a $\approx 78\%$ of positive prediction rate with an AUC of 0.92, which is more effective compared to other studies and the standard scoring system.

1) RADIAL BASIS FUNCTION (RBF) NETWORKS

A Radial Basis Function (RBF) neural network is a special type of feedforward network with a well-defined structure [374], consisting of three layers of neurons as shown in Fig. 22. These layers are the input layer, a single hidden layer, which uses RBF as its activation function, and an output layer, which outputs the weighted sums of the neurons in the hidden layer.

A study that uses an RBF network is conducted in [375], where the authors detect tremors in Parkinson's disease patients using deep brain electrodes that gather data from the subthalamic nucleus. Frequencies in a time window of one second are fed into an RBF network as inputs and the network classifies the pattern as either tremor or non-tremor. The system achieves a $\approx 90\%$ accuracy in detecting

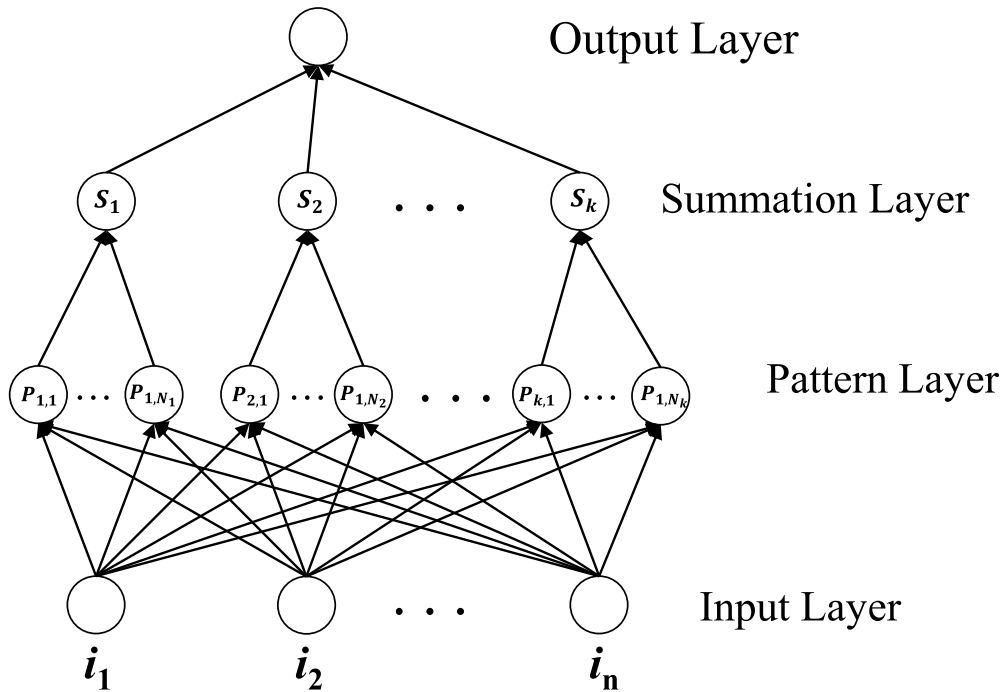


FIGURE 23. General structure of a probabilistic neural network. The network takes n dimensional inputs and has k number of classes in the data. N_1 through N_k refer to the number of data points that belong to each class (i.e. N_1 is the number of training data points that belong to class 1). This makes the number of neurons in the pattern layer equal to total number of data points in training data.

tremor patterns. In another study, researchers use an RBF network to identify cerebral vascular accidents by using CT images [376]. The input space for the RBF network consists of 51 features extracted from CT images and the output of the network is the classification of different regions of the images as normal vs. abnormal regions. Their work achieves specificity and sensitivity as high as $\approx 97.6\%$.

2) PROBABILISTIC NEURAL NETWORKS

A Probabilistic Neural Network (PNN) [377] is another type of a feedforward neural network with a well-defined structure. PNNs consist of four layers, which are called the input layer, pattern layer, summation layer, and the output (decision) layer as shown in Fig. 23. Although PNNs usually use RBF activation functions in their pattern layer, the primary difference between an RBF and a PNN network is that the number of neurons in a PNN depends on the number of the data entries as well as the number of classes in the data, whereas structure of an RBF network is independent from the input data.

Wang et al. [378] use a PNN to classify 8 classes of heartbeats, including 7 types of arrhythmia and normal heart beat. Although they extract 200 features from each heartbeat, they reduce the input space using dimensionality reduction techniques such as LDA and PCA (see Section V-A). They achieve more than 99% accuracy in classifying the heart beats into these 8 categories. Another study that uses a PNN is presented in [379], where authors use an Enhanced

Probabilistic Neural Network (EPNN) [380] to classify patients with Parkinson's disease from a set of given standard inputs. There are three classes in the study, which are subjects with Parkinson's disease, healthy controls, and patients who are diagnosed with Parkinson's disease but have normal dopaminergic functional imaging. They show that an EPNN can achieve a 92.5% classification accuracy, which is higher compared to other algorithms such as k-NN and DT.

D. DEEP BELIEF NETWORKS

Deep Belief Networks (DBNs) are a type of deep neural network mostly used for extracting features from raw data. DBNs can be used in unsupervised learning (see Section V-A9). They reconstruct the input by first building a lower-dimension version of it in the hidden layers and rebuilding a higher dimension output from these hidden layers. In this way, the hidden layers function as extracted features from the input data. DBNs are mostly composed of multiple layers of *Restricted Boltzmann Machines* [103] or *Autoencoders* [104]. These structures can be trained using a Greedy algorithm separately, which makes them a suitable candidate for deep networks by reducing the training time. When used with supervised training, DBNs can also be used as effective classification networks. A sample structure for a stacked autoencoder is shown in Fig. 24.

The study in [106] uses DBNs to extract features from multimedia (audio/video) inputs and an SVM to perform emotion recognition. Although their classification method is not an

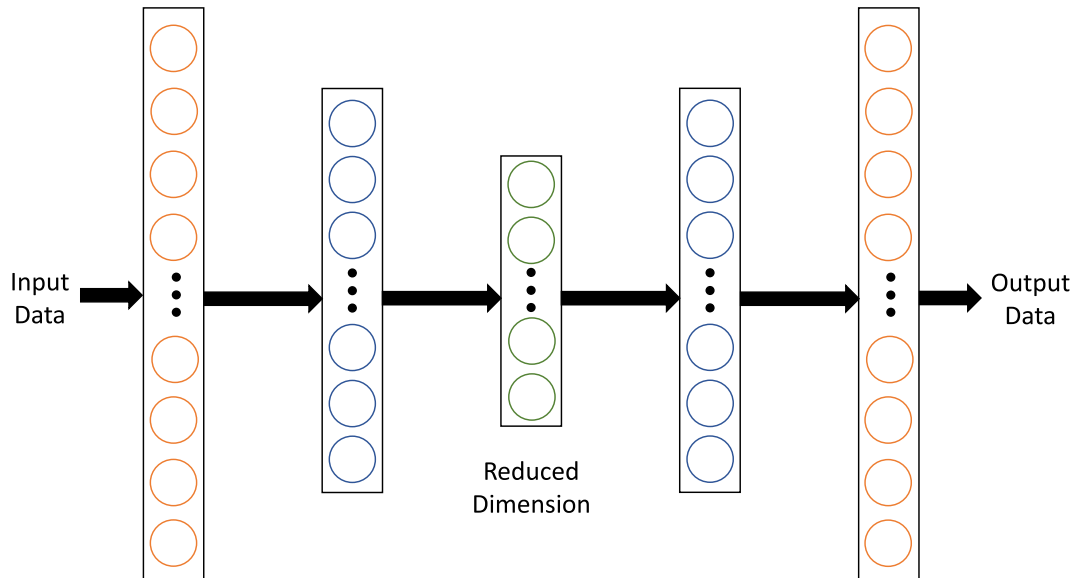


FIGURE 24. The structure of a stacked autoencoder, which is trained by replicating the input data at the output. During the training, the hidden layers form a lower-dimension representation of the data, from which the output is reconstructed. Therefore, the hidden layers function as *features* of the input data.

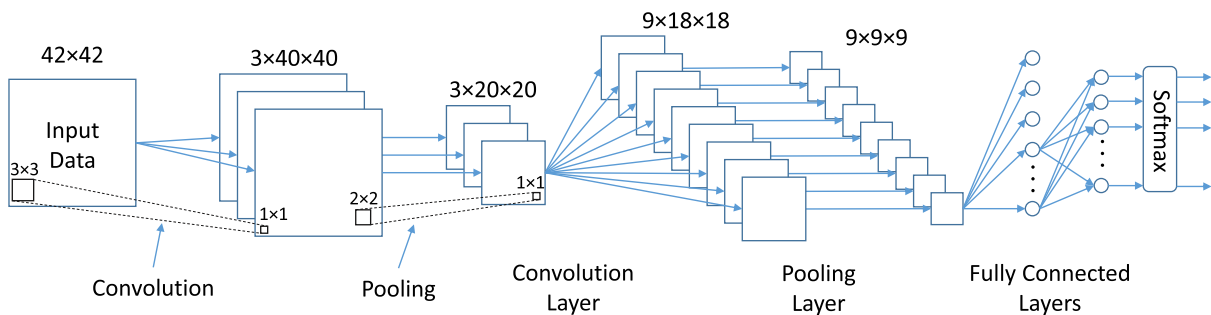


FIGURE 25. The structure of an example convolutional neural network (CNN). The network takes a 42×42 framr (of pixels) and passes it through a convolution layer that has three 3×3 filters, yielding a layer with three 40×40 frames. These frames are then passed into a pooling layer that pools 2×2 frames into a single 1×1 cell and shrinks the size of each frame from 40×40 to 20×20 . Another iteration of convolution and pooling results in nine 9×9 frames which are then fed into the fully connected layers.

ANN, their main contribution is the extraction of the features from multimedia inputs with various types of DBNs, with the eventual goal to use an SVM for the actual classification task. They show that their method for feature extraction achieves a better accuracy when compared to conventional feature selection techniques by having $\approx 4\%$ higher accurate classification. In [381], another DBN-based emotion recognition system is introduced; this study conceptualizes how the recognized emotions of a presenter in a smart classroom can be used to determine how effectively the presenter is building a rapport with the audience (i.e., the students). This information is fed back to the presenter in real time in order to allow the presenter to adjust their nonverbal behavior, such as hand gestures, body language, and intonation. This study is based on the fact that the communication among humans is heavily influenced by these nonverbal cues [382], [383] and presenters who can effectively use their nonverbal communication component can deliver a much more memorable lecture by

activating both analytical and emotional memories that are a part of the human brain [384]. Note that this separation of the two different parts of the human brain is also termed “System 1” and “System 2” [385] in psychology.

Another application that uses DBNs is presented in [386], which focuses on clustering genes that are relevant to glioblastoma (a type of brain tumor) prognosis. 1100–1400 input features are selected from the genes of patients and fed into different DBNs and their dimensionality is reduced to 100–200 features. These reduced-dimensionality features are then used as an input into a clustering algorithm, which is able to distinguish among six different subgroups of patients. Authors show that patients in each cluster show similar prognosis for their glioblastoma, while the prognosis differs with patients in other subgroups. Hu *et al.* [387] detect Alzheimer’s disease by using fMRI data. In the preprocessing phase, they capture time series signals of 90 regions of interest in the brain and convert this

data into a 90×130 matrix to generate the input data for a DBN that reduces the number of features. A softmax layer is used as a classifier to detect if the patient has Alzheimer's disease. They report an accuracy of 87.5% in detecting the disease, which is higher than other methods such as an 82.1% accuracy achieved by an SVM.

E. CONVOLUTIONAL NEURAL NETWORKS

One of most commonly used form of deep neural networks is Convolutional Neural Networks (CNNs), which have shown widespread success in analyzing images. The internal layers of a CNN are not fully connected; each neuron in a layer is only connected to neurons in its neighborhood in the previous layer. This limited connection structure helps CNNs find local features from low abstractions in the first layers (e.g., detecting horizontal or vertical lines) to higher abstractions in the final layers (e.g., detecting complex structures like faces). CNNs use activation functions with no restriction on the output value (such as ReLU), which avoid the possibility of neuron saturation in deeper networks (see Section XI-A2). CNNs are composed of two layers: the *convolution* layer, which facilitates the computation of local features and the *pooling* layer, which reduces data dimensionality in a localized region of neurons by using aggregation functions that are applied to these local features (such as max, average).

An example structure of a CNN network is shown in Fig. 25. CNNs typically have a few layers of fully connected networks followed by a softmax layer in their final layers, especially for classification problems. There are some established CNN architectures such as *Lenet-5* [388] which was introduced in 1989, *GoogLeNet* [389], *AlexNet* [390], *VGG* [391], and *Network-In-Network* [392]. Although the idea of CNNs is not new, they have received growing interest in recent years due to the increase in the computational power of computers that utilize computational accelerators such as GPUs [321], [393].

CNNs are widely used in different research topics and have shown near-human accuracies in many applications. For example, Gao *et al.* [394] analyze CT image slices to detect Interstitial Lung Disease (ILD) using CNNs. They define five classes of outputs including a healthy class and four different ILD classes. Their network achieves an AUC of 0.99 using 4 layers of convolution and pooling followed by two fully connected layers and a softmax layer at the output. Another application that takes images as inputs is presented in [156] where the CNN takes histology images as inputs and detects different regions of the image as benign or malignant glands. They achieve an F1 score of 0.9 and also report Hausdorff distance for the masks of benign and malignant glands that they create compared to the real mask.

Salvador *et al.* [395] develop a system that detects normal and malignant tissues during surgical procedures from a video feed. The nature of their problem requires analyzing a vast amount of data in real-time to provide a feedback to the surgeon about detected cancer tissues. They use deep learning techniques and a hardware accelerator in their system to

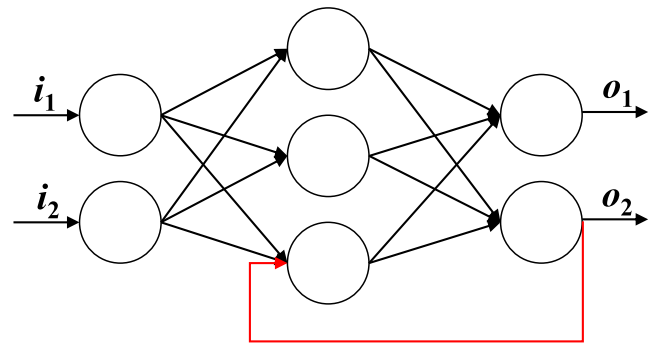


FIGURE 26. A sample recurrent neural network (RNN), in which the output of a neuron is connected to the input of a neuron in previous layers, creating a feedback cycle in the network. This property of RNNs allows them to “remember” previous data points.

achieve low latency for their processing time. They report their full system time as being between 42 to 77 seconds for different input sizes. Another application that takes input data from images in a video feed is presented in [396] where authors develop a framework that uses a video feed from ICU for non-contact vital sign monitoring. In this application, patient vital signs, such as heart rate, respiratory rate, and oxygen saturation, are estimated from skin regions in a video. The CNN that they develop analyzes the images, detects whether the patient is in the image or not, and if the patient is present, finds the regions in the image that show the skin of the patient. They study 30 infants in NICU and show 98.75% accuracy in detecting the skin regions in the images.

The structure of CNNs is flexible and they can be configured to take types of data other than images. For example, Krizhevsky *et al.* [390] detect 5 types of arrhythmia in the ECG data. Since the ECG data is a one dimensional input, they develop a novel 1-D convolutional neural network that takes ECG recordings as input. They show how the neurons are connected in their network and describe the training process. Their final system has accuracies as high as 99% in detecting different types of arrhythmia.

Another arrhythmia detection framework that uses CNNs is presented in [397], in which the dataset includes more than 60,000 ECG recordings that are annotated by clinical ECG experts into 12 different classes of heart beat arrhythmia and are used to train a 34-layer CNN to differentiate among these classes of arrhythmia. They show that their network can classify the recordings better than the cardiologists; the CNN achieve an overall F₁ score of ≈ 0.81 , while cardiologists reach an F₁ score of ≈ 0.75 .

F. RECURRENT NEURAL NETWORKS

Recurrent neural networks (RNNs) are a special type of ANN in which there are feedback paths from neurons to other neurons that are in the previous layers. This property allows RNNs to base their classification not only on continuously incoming data, but also the data that has entered the ANN (and has been computed by the network already) at a previous

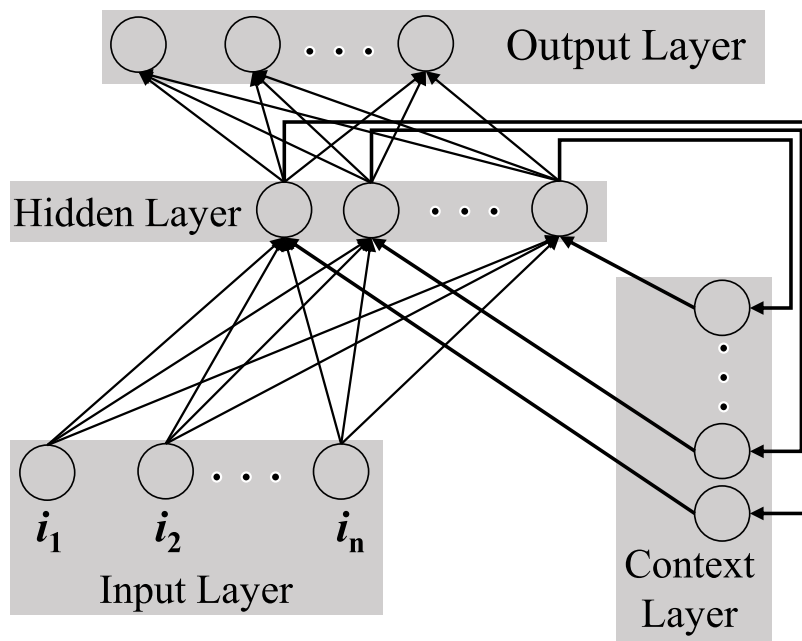


FIGURE 27. General structure of an Elman network.

time. A simple RNN is depicted in Fig. 26. RNNs have been shown to be effective in processing natural language and time series data like audio [398]. Some variants of RNNs have been applied to medical data successfully, which are briefly described below.

1) ELMAN NETWORKS

Elman Networks [399] were created by adding a layer of *context* to a feedforward neural network with one hidden layer as shown in Fig. 27. Elman networks were originally designed for language processing, but they have proved to be useful for all applications that have sequential input. For example, Chu *et al.* [400] use Elman networks to detect whether an elderly patient fell while walking. They collect their data by placing a sensor board in the patient’s pocket and using the accelerometer from the board, they detect a fall incident with high accuracy.

2) LONG SHORT-TERM MEMORY (LSTM)

LSTM [401] is a type of an RNN that has mechanisms to store data values for either long or short periods of time. The unit that remembers the values is called the LSTM unit and it stores values without applying an activation function which avoids any distortion in the data. The benefit that LSTMs provide is that they are designed to remember data for long periods of time as opposed to other models that struggle with remembering older information. An example application of LSTMs is provided in [402] where authors develop a variant of LSTMs for the purpose of subtyping patients based on their electronic health records. They create two networks; (i) one for a diabetes diagno-

sis based on hospital visit information, which achieves a 96% accuracy and (ii) another one for predicting Parkinson’s progression. They show that their prognosis analysis for Parkinson’s disease has better performance compared to other methods.

3) BIDIRECTIONAL RECURRENT NEURAL NETWORKS (BRNN)

BRNNs [403] are another type of RNN that allow access to future input data in their current state. This means that the output is influenced by both the past and the future information; this makes BRNNs a suitable candidate for the data that needs to be analyzed within a given context. For example, Ma *et al.* [404] use BRNNs to analyze electronic health records. The goal of their study is to predict the $(t + 1)^{th}$ visit to the hospital by a patient, given the information of that patient’s visits to a hospital from time 0 to time t . They test their model and show that their approach beats baseline by at least 2% more accuracy in diagnosis prediction.

4) COMBINING DIFFERENT TYPE OF ANNS

Some applications use two different types of networks simultaneously. For example, Shin *et al.* [158] develop a system that reads a chest X-ray image, detects any abnormality in the image, and adds a description to the image based on the abnormality that it finds. They combine two networks, a CNN that is responsible to classify images and an RNN that generates annotations for the images. Their reported results show that their system is able to achieve acceptable results for the annotations that it has created.

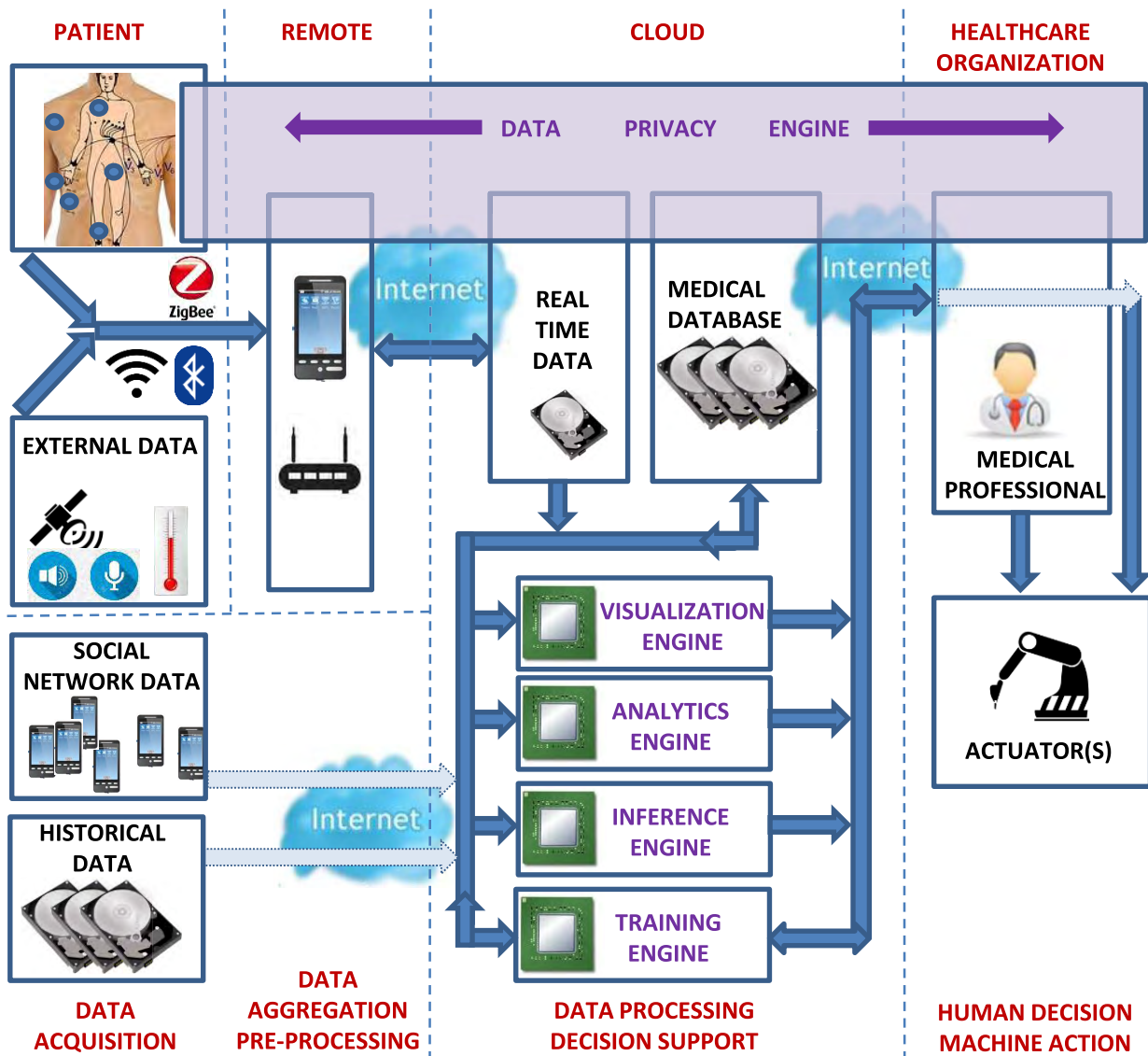


FIGURE 28. Components of a Medical Cyber Physical System (MCPS) for applications such as remote patient health monitoring. The real-time data is acquired either from the patient’s body or their environment (e.g., temperature); this data can be augmented with historical medical records from either public sources or other collaborating hospitals’ databases. The pre-processing and aggregation component allows the raw data to be turned into features before being transmitted into the cloud. The cloud uses the algorithms that are thoroughly studied in this paper, while the end result can be observed by either a human (medical personnel) or another machine (actuator). The feedback received from the human or the machine can be fed back into the training engine to further train the algorithms.

XII. MEDICAL CYBER PHYSICAL SYSTEMS (MCPS)

While the algorithms and models that are described in this paper can work on historical medical data that is stored in a database, a Medical Cyber Physical System (MCPS) — depicted in Fig. 28— can acquire and process patient health data in real time [55]. A typical application of an MCPS is remote patient health monitoring [7], in which a patient is monitored through a set of body-worn sensors and the acquired data is pre-processed and transmitted into the cloud. The execution of most of the algorithms that are described in this paper takes place in the cloud and the results are presented to a medical professional through a mobile device. However, in certain applications, a task split between mobile acquisition

devices and the cloud has been proposed, where a portion of the overall algorithm executes in the mobile devices to ease the burden in the cloud and avoid network congestion. For example, the KNOWME network [188] describes an application where a light version of the algorithm is run on the mobile device that is used to collect data from the patient and present real-time physical activity tracking information. The components of an MCPS are detailed below.

A. DATA ACQUISITION

Data acquisition in an MCPS corresponds to Box D in Fig. 1. Various data that is simultaneously gathered from a variety

of sources are fed into the machine intelligence algorithms. These sources can be:

1) PATIENT

The data that is acquired from the patient includes multiple physiological signals such as Electrocardiogram (ECG), skin temperature, surface EMG, gait (posture), blood glucose, or respiratory rate, depending on the purpose of the health monitoring application. For example, while monitoring for cardiovascular diseases benefits from obtaining ECG, gait, and respiratory rate data, Parkinson's disease patients benefit from gait monitoring, but not so much from ECG [5]. In practice, commercially-available skin-worn sensors [49] are capable of measuring a variety of physiological signals, including but not limited to ECG, EMG, acceleration, etc. and are continuously developed to incorporate more advanced bio-sensing capabilities [48], [50]. Battery-based sensors [49] suffer from the problem of continuously having to power the sensors; while this is not a problem in application where the monitoring duration is much shorter than the battery life of the sensor [5], in applications where the sensor is used a long period of time, *passive RFID sensors* is preferred [405]. Furthermore, a set of sensors that are built into today's smartphones (or other mobile devices) can be utilized for data acquisition. These types of sensors are termed *non-dedicated sensors* [406], because they are not dedicated to a single task; rather, they can be used in any application that can utilize them in a *soft-sensing* setting [407].

2) EXTERNAL DATA

In addition to the data acquired directly from the patient's body, auxiliary data, such as environmental information, may be used in context-aware applications. This auxiliary data provides information about the state of the patient at the time of the data acquisition. For instance, Bisio *et al.* [408] describe an application that monitors patients suffering from a set of mental and physical disabilities (called *co-morbidities*) at home, using smartphones that collect different parameters, such as acceleration data, audio information, WiFi connection, and the time of the day. Using these parameters, this application is able to localize the patient, identify if the patient is alone, and their current physical activity. Note that in cases where the smartphones of users other than the patient are being used, issues such as the trustworthiness of the acquired data [409] arise. Furthermore, in such scenarios, incenting other users to contribute the data from their smartphones is a challenge [410], [411].

3) SOCIAL NETWORK DATA

Rather than obtaining data from physical sensors such as GPS, temperature sensors, and barometers, social network data corresponds to information collected from cyberspace, which is generated by users in online social networking sites such as Twitter and Facebook. For example, in [412], authors address various problems, including—but not limited to—tracking and localizing illnesses over time and by geographic

region, and inferring symptoms and medication usage by analyzing user messages on Twitter. Social network data can be used to augment patient bio-signals to determine the likelihood of a remotely-monitored patient being impacted by an epidemic.

4) HISTORICAL DATA

A corpus of long-term patient health records, stored either within the hospital that houses the primary physician of the patient or other collaborating hospitals (Box DB in Fig. 1), can prove invaluable in providing training data for the machine learning algorithms. A sample medical record is available at [413], which includes an example personal health record including medications that the patient takes, history of symptoms due to various medical conditions, a summary of the doctor notes during physical exams, and lab tests reports.

B. DATA AGGREGATION AND PRE-PROCESSING

Analyzing large and complex datasets generally requires a large amount of memory and computation power. At the same time, using a large number of variables that are repetitive and non-discriminatory in nature negatively impacts the classification performance of algorithms by causing overfitting issues that lead to poor generalization. Before the acquired patient data is transmitted to the cloud, feature extraction (Box FE) and feature aggregation (Box FA) steps are performed by a nearby, computationally-capable device. Typically this device is referred to as a *concentrator* [414] within the IoT field or a *cloudlet* [415]–[418], within the mobile cloud computing field.

C. STORAGE OF HEALTH RECORDS

The large medical record database (Box DB in Fig. 1) consists of two components:

1) REAL-TIME DATA

The aggregated features that are transmitted by the cloudlet arrive in the cloud to be processed by machine intelligence algorithms. This new data serves two purposes; (i) it must be compared against the permanent database to determine if the doctor has to be warned of a newly-developing patient health condition, (ii) it must be used to appropriately “update” the training engine to eventually become a part of the permanent database, i.e., *machine knowledge*.

2) MEDICAL DATABASE

This permanent database contains not only the data about the remotely-monitored patient, but also all medical data of other patients that are being cared for by the same healthcare organization. Having a database that includes a large number of patients with similar health conditions allows machine intelligence algorithms to work more accurately [10].

D. DATA PRIVACY ENGINE

Medical information that is personalizable is termed Protected Health Information (PHI) [54]. In the USA, the privacy

of PHI is strictly mandated by Health Information Privacy and Accountability Act (HIPAA) laws [54]. To prevent violation of the HIPAA laws (and similar laws around the globe), an MCPS consists of a component that encrypts/decrypts data that is being transmitted between any two nodes of the MCPS that can be temporarily intercepted by adversaries [419], with the intention to steal or modify the acquired patient medical data. For example, an adversary can intercept the Bluetooth communication of the WBAN on the patient's body with the intention to infer the patient's health condition, even if in rough form (e.g., healthy vs. abnormal), without actually accessing the data [414]. This type of an attack is defined as a *side channel attack*. A survey of a rich set of encryption algorithms, adversary models, and side channel attack methods are surveyed in [55]. Note that there is no specific "Box" that the Data Privacy Engine corresponds to in Fig. 1; this is because *every* box along with the transmission medium between any two boxes must be protected via this engine. As a result, this engine can be thought of as a large umbrella that covers the entire MCPS.

E. VISUALIZATION ENGINE

In its raw or even pre-processed format, the amount of data that is available in the permanent medical database is beyond the processing capabilities of the human brain [37]. A visualization algorithm (considered to be a part of Box A in Fig. 1) is responsible for substantially reducing the amount of information displayed to a medical professional by using intuitive visual representations of the patient medical data. In [37] and [47], authors describe a 24-hour "ECG clock" that is capable of visualizing 24-hour patient Holter ECG recordings, as well as illustrating the "predicted" patient health condition to provide decision support to a doctor. The ECG clock reduces 100s of MB of ECG information into a single plot that allows the doctor to monitor 20–30 cardiac patients within less than a minute, without losing any critical information.

F. ANALYTICS ENGINE

Also considered to be a part of Box A in Fig. 1, this engine uses data mining techniques to find patterns that are related to the onset and evolution of diseases in the medical database. Findings of this engine can be applicable either immediately to a patient who is being monitored or not. In the former case, various fast methods (e.g., [420]–[422]) have been developed to analyze generic time-evolving sequences for (i) estimation/forecasting of missing/delayed/future values, (ii) outlier detection, and (iii) frequent values identification, while considering memory and storage requirements. In contrast, in the latter case, traditional data mining techniques (e.g., [423]–[425]) can be used to make *long term* inferences; they can be utilized to continuously learn biological characteristics of diseases, which can eventually facilitate the diagnosis and prognosis of future patients. When there is a need for computationally intensive operations, *offline* computation can be utilized for

resource efficiency, i.e. by performing computations when the demand for premium computational resource is low.

G. INFERENCE ENGINE

As the most important functionality of Box A in Fig. 1, the inference engine is responsible for making inferences in *real-time* to provide decision support to doctors for the patients that are being monitored or inform the patients about their health status. All of the surveyed machine intelligence algorithms in this paper can serve to provide this functionality. Due to its real-time requirements, this engine is characterized by significantly stricter computational requirements than the analytics engine, thereby requiring computational resources at a time when they may be expensive. The information that was extracted by the analytics engine is usually used to accelerate this *online* portion of the MCPS to reduce the associated computationally intensive operations.

H. TRAINING ENGINE

Most of the algorithms — except data mining methods — that are used in the inference engine require training to be effective. Represented as Box T in Fig. 1, the goal of the training engine is to determine the predictive relationship between *input* and *output variables* [68] by combining data from the historical database, real-time database, and feedback from the doctor or the patient. For example, the study in [10] evaluates a scenario in which a large hospital database [2] is used for the diagnosis of cardiac conditions. Features such as QT, RR, JT, QTp, JTp, which are extracted from the ECG signals of the patients in the database, are used as *input variables*. The same database provides an annotation file, which indicates whether a patient has a known cardiac condition such as long QT1 (LQT1) or long QT2 (LQT2) syndrome. This annotated information is used as *output variables* in machine intelligence algorithms. Training of the algorithms is achieved by using these input-output pairs; the accuracy of the algorithms —utilized in this study— improve steadily with an increasing number of training pairs.

I. ACTUATORS

An *actuation mechanism* is considered to be one of the observers in Box O of Fig. 1. This mechanism refers to the actuators (e.g., a robotic surgery arm) and the software that controls the actuators. In its simplest form, the actuator can be an insulin injection device with no follow-up feedback to the system [24] or a robotic surgery arm that continuously measures the location of the arm to aid in the adjustment, as well as auxiliary feedback from the doctor who is performing the surgery [41].

J. MEDICAL PROFESSIONAL

As the primary observer of the remote health monitoring results, nurses, doctors, and other medical staff are represented as Box O in Fig. 1. Their primary function is to provide feedback to the MCPS to help the machine intelligence algorithms in understanding the end points, as well

as perform in-hospital measurements of the patient vitals (e.g., blood pressure, heart rate, weight, etc). In its simplest conceptualization, medical professionals can be thought of as being the “*human*” portion of an MCPS, while all of the other components are the “*machine*.”

XIII. CHALLENGES AND OPPORTUNITIES

Despite its potential to revolutionize the medical field, utilization of machine intelligence has been at a limited scale in healthcare applications. Any notable application has been either in a research project setting or a very limited application to a highly restricted area of medicine. In this section, we provide a list of the challenges in integrating machine intelligence into healthcare applications in four different categories.

A. SYSTEM LEVEL IMPLEMENTATION CHALLENGES

One of the most important challenges in deploying a full scale Medical Cyber Physical System (MCPS) is that due to the complexity of even the simplest system, deployments have been limited to individual applications; a generalized system that collects data for a multitude of applications, aggregates them, and uses them as a unified database for multiple applications has been unrealistic. A conceptual MCPS scenario is proposed in [5], where the authors describe an MCPS that can handle three applications, namely Chronic obstructive pulmonary disease (COPD), Cardiovascular Diseases (CVD), and Parkinson’s Huntington’s Diseases (PD/HD). The proposed system collects all patient data about these three diseases (and potentially some more in the future) and saves them under a single database. The application of the machine intelligence algorithms and data analytics on this common patient database eventually allows the doctors to use the results as a *decision support* tool. While this recipe for a system sounds like the start of a new digital era in healthcare, many challenges can plague such a deployment. They are as follows:

1) COST

Cost is an important factor that needs to be considered during the system implementation phase. It is usually dictated by the amount of storage (i.e., cache memory) required on the chip, which is directly related to the area of the chip. On-chip memory needs to be optimized, while maintaining low off-chip memory bandwidth [426]. In many cases, this leads to custom designs that fit the specific application needs. For instance, Lee and Verma [427] design a custom processor that integrates a CPU with configurable accelerators for EEG-based seizure and ECG-based cardiac-arrhythmia detection. When cloud computing is being used as the execution platform of the algorithm, the cost of an algorithm is directly proportional to its runtime; as an example, Kocabas *et al.* [428] study the cost of computationally-intensive medical applications and note that the resource requirement of an application is not necessarily an “incremental” value, because all cloud service providers

(e.g., Amazon Elastic Cloud 2 [9]) rent computational resources in different “packages,” where each package implies a set of CPU, GPU, and storage resources.

2) ENERGY/POWER CONSUMPTION

Medical Cyber Physical Systems (MCPS) usually consist of a combination of battery-powered (e.g., sensors, mobile phones), passive (e.g., battery-less sensors), and outlet-powered devices (e.g., IoT concentrators). Having the potential to revolutionize health care, their sustainability is of utmost importance and mandates energy-efficient operation. To this end, it is crucial to characterize the energy/power consumption of the three fundamental processes that take place in an MCPS: (i) sensing, (ii) computation, and (iii) communication. For example, since battery-operated devices have limited power, it is necessary to decide whether data should be processed/preprocessed locally on these devices or transmitted to a more computationally-capable outlet-powered device (e.g., cloud server or a cloudlet [411]). On the other hand, the amount of power consumed for sensing and communication needs to be considered. The above considerations have a direct impact on the Quality of Service (see Section XIII-A3) for an MCPS. Zois *et al.* [35] propose an energy-efficient sensing mechanism in a WBAN equipped with a set of biometric sensors and a mobile phone for physical activity monitoring of individuals and show energy gains as high as 68%, compared to prior work.

3) QUALITY OF SERVICE

The quality of service achieved by an MCPS refers to the overall performance offered by such a system and can be quantified using the metrics: latency, availability, and reliability.

Latency is a measure of the time delay between the onset of a medical event (e.g., heart-attack, stroke, seizure) and the detection of such an event by an MCPS. Tolerance to latency is frequently dictated by the medical application of interest in conjunction with the system characteristics (e.g., communication protocols) and determine whether a specific machine intelligence algorithm is suitable for a given application. Shoeb and Gutttag [429] propose an SVM-based classifier that detects the onset of an epileptic seizure based on EEG and ECG data with a mean latency of 4.6 seconds.

Availability, on the other hand, describes the amount of time a system is functioning, and is purely dictated by system characteristics (e.g., communication protocols, software and hardware components). Still, since MCPSs perform time-sensitive tasks, it is imperative to use machine intelligence algorithms that are robust with respect to system availability.

Reliability is a measure of the ability of a system to perform its task under varying system characteristics and constraints (e.g., communication protocols, software and hardware components, channel characteristics and interference, calibration issues). Biason *et al.* [430] propose a framework that exploits both physical activity characteristics and channel state

information to perform reliable and energy-efficient physical activity detection using an energy-constrained WBAN.

4) CONSTRAINED DEPLOYMENT

MCPS technology should neither be intrusive (i.e., it should not intervene with the daily activities of the monitored individual) or cause health risks. For example, a limited number of sensors should be put on or inside the human body and custom system designs must be devised to minimize health risks (e.g., strangulation hazards, antenna radiation) during deployment. Unfortunately, these design choices directly affect the type of machine learning algorithms that can be used in practice. In [431], a micro-power EEG acquisition SoC with integrated seizure detection processor is designed to enable continuous on-scalp EEG monitoring without the use of cables that can pose a severe strangulation hazard during convulsions.

5) SECURITY & PRIVACY

Patient medical data constitutes sensitive information and inappropriately sharing and using it can significantly compromise the privacy of patients. However, data sharing between healthcare organizations, transmission of confidential medical data via wireless media and processing of such data constitute vital requirements for the wide deployment of MCPSs. In [432], a system-based on the concept of fully homomorphic encryption is proposed to enable privacy-preserving health monitoring via the extraction of relevant information from encrypted patient medical data. The authors demonstrate, through numerical simulations, that the proposed framework can be used to securely transfer and analyze ambulatory health monitoring data in real time. They use a public cloud service provider, such as Microsoft Azure or Amazon EC2, as the processing platform. While the privacy of the data is also a concern when a private cloud is being used (e.g., the datacenter of a hospital), data privacy concerns are exacerbated when a public cloud is used. Cloud service providers are required to sign a Business Associate Agreement (BAA) to serve as the cloud service provider for the storage and processing of medical data [432].

6) INTEROPERABILITY

As already discussed in Section XII, MCPSs consist of heterogeneous subsystems that exchange data using a variety of protocols. To ensure the seamless interaction of these components and the optimum overall performance of the system, it is necessary to enable (1) processing and computation across various data formats and standards and system configurations, (2) across various data transfer standards (e.g., Bluetooth, ZigBee), and (3) plug and play device interaction. In [433], a framework is proposed that can abstract different and incompatible devices in order to support interoperability in a consistent technology-independent manner and enable the seamless application of machine learning techniques to infer individual behaviors and habits without human intervention. The authors anticipate that the proposed framework

can be used to timely detect and prevent health hazards in a smart home environment, especially in the case of the elderly or individuals with chronic diseases.

7) DATA CONSISTENCY

In an MCPS, patient data is fragmented over multiple devices (i.e., mobile phones, computers, cloud) and over time. At the same time, procedures such as laboratory tests and medication orders vary by patient and are, in practice, obtained in an unscheduled manner. To make things worse, data may be missing, due to system failures or because there was probably no provision for collecting such information. Thus, one of the main challenges in an MCPS is to ensure that the performance will not be degraded due to data consistency issues. In [434], a variety of methods (i.e., mean, hot-deck and multiple imputation, multi-layer Perceptron, self-organization maps, and k-nearest neighbor) are combined with artificial neural network models to impute absent values in a breast cancer dataset in order to predict early breast cancer relapse. The authors numerically show that the machine learning-based imputation methods outperform statistical-based methods.

8) PERFORMANCE IMPROVEMENT AFTER DEPLOYMENT

In many medical applications, classifiers are trained before deployment and/or require training data to be manually added after deployment. At the same time, statistical models and distributions of features change over time due to the progression of diseases, interventions, and other factors. To maximize the performance of an MCPS, automatic ways of improving the performance of machine intelligence algorithms after deployment have been proposed. Longstaff *et al.* [308] compare active learning with three different semi-supervised learning methods and observe that the former approach leads to the highest improvement. However, democratic co-learning proves to be more appropriate in the case where initial accuracy is low and user interaction needs to be avoided.

B. DATA CHALLENGES

The amount of data that is available for certain applications is far beyond the processing capability of traditional processing platforms; this *Big Data* problem was initially described using 3 V's [407]:

- *Volume* of data denotes the scale,
- *Velocity* of data implies the speed at which new data is arriving (e.g., streaming), and
- *Variety* of data signifies the amount of different types of data available.

This was recently expanded to 5 V's by adding:

- *Veracity*, which implies the uncertainty (accuracy) of existing data, and
- *Value*, which implies its ability to provide statistically-meaningful information.

The true potential of the machine intelligence algorithms can only be harnessed when a significant amount of high quality data is available to train them. For certain

applications, it is very hard (and expensive) to collect data, since they require expensive equipment to acquire it; for example, capturing functional MRI (fMRI) data requires expensive MRI devices, which are only available in large HCOs. The available data also contains missing or unreliable samples; for example, continuous ECG data collection using Holter devices is known to have segments in the data, where one or more electrodes are completely disconnected. This not only complicates algorithm design by requiring appropriate mechanisms to address such reliability issues, but also reduces the effectiveness of the data collection subsystem. In certain applications, having two sets of data that provide complementary pieces of information cannot be obtained in a practical setting. One such example is the combined informational value of EEG and fMRI data; while EEG provides high temporal but poor spatial resolution, fMRI has exactly the opposite characteristics by providing high spatial but poor temporal resolution. Therefore, one would expect the combination of the two to provide a more complete source of information for algorithms use for determining brain function. However, obtaining both EEG and fMRI is impractical in a regular healthcare setting. Even though abundance of data can be problematic as we already discussed, not having sufficient amount of data is also an issue. Last but not least, noise models for the existing data are generally unknown, which reduces the usefulness of the data.

C. ALGORITHMIC CHALLENGES

While obtaining a sufficient amount of reliable data presents the problems mentioned in Section XIII-B, designing algorithms that are accurate enough to be used in decisions that involve human lives is a challenge of its own, even when the required data is available. Furthermore, algorithms or models for certain scenarios do not exist, such as models to capture a patient's health over time. Existing algorithms must be very accurate to provide support for the decisions that healthcare officials make. If the outcome of the algorithms cannot provide any additional insight to the doctors, they will not be adopted, because the experience of a doctor, combined with the incredible ability of the human mind to make associations in the existing data creates a formidable "competitor" for any machine intelligence algorithm. The only way an algorithm can be useful is when it can use its advantage in applications that require sophisticated computations on a vast amount of data, potentially gathered from different sources. From a practical standpoint, effective use of these algorithms can include simple and intuitive visualizations to the doctors that are based on the processing of vast amount of data (e.g., the ECG visualization scheme in [37]), while enable them to make accurate, quick and efficient decisions. Furthermore, since most doctors are typically specialized in one area (e.g., oncology), algorithms can provide an advantage by incorporating domain knowledge from multiple domains. Last but not least, sequential decision making algorithms can automate certain medical processes and significantly improve patient's health when doctors are not available.

From the application point of view, one of the most important challenges is *personalization* of the algorithms; it is very challenging to tune an algorithm using one patient's data and be applicable to another patient without any modification. Due to the highly sophisticated inter-related processes in a human body, every person may have different sensitivities to different algorithmic parameters. This implies that different bio-markers may have different significance for different patients. As a consequence, on one hand, incorporating a wide array of biomarkers can improve the personalization aspect while significantly increasing dimensionality; on the other hand, using a limited set of input parameters can improve general algorithmic accuracy by eliminating issues such as over-training, however, may not work as well for different patients. This creates a very important algorithmic challenge: how to design algorithms that use the least amount of input parameters (whether biomarkers acquired from a patient or environmental data) and the least amount of sensitivity for patient-to-patient variability.

One last challenge with designing algorithms is the dependence on the dataset when testing their functionality. When an algorithm is designed, it is typically tested with a single database. However, it may not produce the same results when tested with a completely different dataset, because standardized methods for creating the database do not exist or some databases may include data from different sources. As an example, the THEW ECG database [2] contains patient records from multiple countries in multiple decade time span. Patients in different geographic regions may be exposed to different environmental factors, highly variable diets, and physical activity habits. Testing an algorithm with such diverse data (although the data is restricted to ECG recordings) creates interesting challenges for algorithm design; determining which data points to include/exclude may mean the difference between eliminating invaluable data that would have otherwise yielded much more generalizable results versus eliminating data that would unnecessarily create distant (or even outlier) data points.

D. LEGAL AND POLICY CHALLENGES

Most of the challenges described in Sections XIII-A–XIII-C also present legal problems for HCOs, in many aspects. The HIPAA laws in the US require careful handling of protected health information (PHI) [432] to avoid the exposure of the personal medical records to unauthorized parties, or even adversaries [55]. Protecting data privacy of PHI becomes challenging for the following reasons:

- (i) Data acquisition is typically performed by sensing devices that have severe power limitations, which makes it difficult for them to implement strong encryption; this in turn provides a weak link for potential attackers [5].
- (ii) Even if the data acquisition is secure, the transmission of the acquired data requires communication protocols that incorporate strong security; battery power-limited sensing devices cannot implement strong security measures from the acquisition to the cloud.

- (iii) If an HCO uses public cloud services, the data is exposed in case of an attack on the cloud service provider (e.g., Amazon EC2).
- (iv) Although advanced encryption algorithms (e.g., Fully Homomorphic Encryption) allow the processing of the data in public clouds, without the cloud service provider being able to observe the data, these algorithms are impractical [6], [435].

In addition to data privacy, the aforementioned 5V problem exacerbates the situation. The sheer volume of the acquired data, in the case of remote health monitoring, implies that storing the raw data is simply impractical. If the HCO stores just the data features, they may not be as useful for the algorithms that are introduced in the future, which require a different set of features to operate. Furthermore, storing only a part of the information presents a potential legal issue in the case of medical malpractice.

One potential alternative to turning the data into features at the time of acquisition (i.e., *pre-processing*) is to save all of the raw data and perform all of the necessary computations at the time the data is needed in the future (i.e., *post-processing*). This significantly increases the requirement for storage resources and shifts the computation burden to the future. To deal with this significant computational requirement, one potential solution is for different HCOs to share a pooled infrastructure, which includes storage and computational resources. However, this also creates a legal concern: who will be the responsible HCO in case there is a malpractice arising from potentially incorrect data? One possibility is that there will be businesses in the future, which focus on these shared database infrastructures [7]; such business establishments can actually help make the algorithms work better, because algorithms can benefit from data coming from different data sources, potentially at different geographic locations.

XIV. CONCLUSIONS

In this survey, a comprehensive discussion of machine intelligence algorithms, which are used in healthcare applications and medical cyber physical systems (MCPS), is provided. This survey centers around describing a conceptual diagram that unifies every healthcare application of interest; furthermore, it enable us to divide a medical cyber physical system into functional units and provide summarized information in a concise manner. These units are **D**: data source, **FE**: feature extraction, **FA**: feature aggregation, **M**: modeling, **A**: algorithms, **T**: training, **O**: observer, and **DB**: database. Units **D**, **FE**, and **FA** exist in every application scenario, while the source of the data being used in the algorithms or models (**D**) usually come from a variety of sources, including but not limited to existing databases, remote health monitoring, and social networks. The goal of units **FE** and **FA** is to identify useful information for further processing, while substantially reducing the amount of information that the algorithms need to process. The modeling step (**M**) and the algorithms (**A**)

constitute the heart of a medical cyber physical system, where the former step enable us to represent the structure and characteristics of a healthcare application while the latter step is used for discovery of new knowledge, classification and estimation tasks as well as automated decision making. A necessary step for both models and algorithms is Training (**T**), although unsupervised learning algorithms do not require this step. Last but not least, the observer (**O**) represents the end user (e.g., a doctor), while the database (**DB**) represents a vast amount of already available medical data. While discussing the different functional units, we have attempted to provide a summarized tutorial and an overview of the huge literature that relates to the functionality of each unit. In an effort to promote further research, a list of opportunities and challenges are provided.

APPENDIX A HEALTHCARE DATASETS

Having a sufficient amount of high quality data is crucially important in developing and testing new machine intelligence algorithms. This motivated the development of publicly-shared medical databases both for educational and research purposes. It is also necessary that these databases contain data that conforms to known data storage standards to reduce the probability of misinterpretation when the data is being shared among different research, professional, or educational entities. Some of these standards are:

(i) *Health Level 7 (HL7)*, which is a set of international standards used when medical data is transferred between different applications and has many related standards to it such as

- *Clinical Document Architecture (CDA)* [436],
- *Continuity of Care Document (CCD)*, and
- *Structured Product Labeling (SPL)*.

(ii) *Digital Imaging and Communications in Medicine (DICOM)* [437], which is a standard for multidimensional data ranging from time series to four-dimensional data, and

(iii) *Continuity of Care Record (CCR)*, which provides electronic summaries of patients' health, especially when it is meant to be transferred to another healthcare organization.

Some databases use file formats that are not standardized (like the formats shown above); for this reason, their use is typically limited. However, they are open-source and documentation about their file structure is readily available. Some examples of these file formats are shown below:

- (i) The *ISHNE* [44] format, which is for storing ECG data that is acquired by Holter devices,
- (ii) The *BCI2000* [438] format is for storing EEG data, and
- (iii) the *General Data Format for Biomedical Signals (GDF)* [439] format is for storing generic time series biomedical signals.

The following are some representative medical datasets that span a wide range of healthcare applications (in alphabetical order):

1) **Alzheimer's Disease Neuroimaging Initiative (ADNI):**

➤ <http://www.adni-info.org/>

ADNI is a multi-center study focused on Alzheimer's disease (AD). The study is Designed to track the progression of AD and define the course of the disease by using various biomarkers such as lab results and providing MR images in addition to PET images for the patients.

2) **Adverse event report datasets:**

These databases contain adverse events, which are related to drugs and medications; they can be used for discovering previously unknown effect of drugs or side effects due to the interaction of —two or more of— these drugs.

- **FDA's Adverse Event Reporting System (AERS):**

➤ <https://open.fda.gov/data/faers/>

The Food and Drug Administration Adverse Event Reporting System (FAERS) collects reports related to adverse events, errors in medication, and complaints on the quality of medicinal products that are submitted to the FDA. The data is submitted voluntarily from healthcare professionals and consumers and is publicly available.

- **Canada Vigilance Adverse Reaction Online Database:**

➤ <http://hc-sc.gc.ca/dhp-mps/medeff/databasdon/index-eng.php>

The Canada Vigilance Adverse Reaction Online Database is designed to collect suspected side effects of health-related products. These reports are submitted by consumers, healthcare professionals, and manufacturers.

- **European database of suspected adverse drug reaction reports:**

➤ <http://www.adrreports.eu/>

European Economic Area (EEA) also keeps track of suspected side effects of medicine and provides access to them publicly.

3) **BCI Competition [440]–[442]:**

➤ <http://www.bbci.de/competition/>

Multiple Brain-Computer Interface (BCI) datasets are available thorough BCI competitions which were held to validate signal processing methods and classification algorithms for the interface between human brain and machine. The datasets provided for the competitions include various types of data such as EEG or Magnetoencephalography (MEG) and have different applications such as controlling a speller, motor imagery, and finger movements.

4) **BioGPS [443]:**

➤ <http://biogps.org/>

BioGPS is a portal that aggregates gene annotation resources into a centralized database. Different gene portals have annotations for genes based on their specialized focus and BioGPS aggregates these annotations to provide an exhaustive description for genomics data.

5) **ClinicalTrials.gov:**

➤ <https://clinicaltrials.gov/>

ClinicalTrials.gov is a web-based tool that keeps records of clinical studies conducted on volunteer human subjects on different diseases and health conditions. These records include the disease, the medical procedure, location of the trial, description of the study, outcome of the study, and other relevant information. The database is public and is maintained by the US national Library of medicine, which is a part of NIH.

6) **DICOM Library:**

➤ <http://www.dicomlibrary.com/>

DICOM Library is an online sharing platform for medical signals, images, and videos. Users can upload anonymized data in the DICOM format through the website and share it with other healthcare professionals and researchers. This medium is free to use and is funded by the European Union.

7) **Digital Database for Screening Mammography (DDSM) [444], [445]:**

➤ <http://marathon.csee.usf.edu/Mammography>

DDSM is a mammography database with the goal of algorithm development for screening and diagnosis of breast cancer. The data includes images of breasts in addition to information of the patient associated with the mammography images. Data is categorized into different classes such as normal, benign, and cancer and is publicly available.

8) **The Federal Interagency Traumatic Brain Injury Research (FITBIR):**

➤ <https://fitbir.nih.gov/>

FITBIR is an informatics system that shares traumatic brain injury data among researchers and is built by a partnership between NIH and DoD. The data submitted to FITBIR includes imaging data in addition to biomarkers and other features such as outcome assessments from traumatic brain injury patients.

9) **Genomic Data Commons (GDC):**

➤ <https://gdc.cancer.gov/>

The Genomic Data Commons (GDC) provides DNA and RNA sequencing data targeting cancer genomic studies. This program is supported by the National Cancer Institute (NCI) and is a sharing platform for data from various cancer research programs.

10) **HCUPnet:**

➤ <https://hcupnet.ahrq.gov/>

The Healthcare Cost and Utilization Project (HCUP) provides a query system that can be used to research

hospital-related data such as utilization, access, expenses, results, and quality. This platform is created by Agency for Healthcare Research and Quality (AHRQ) and has hospital in-patient, surgery, and emergency department data in addition to healthcare data at the county level.

11) **The Interactive Emotional Dyadic Motion Capture (IEMOCAP)** [446]:

➤ <http://sail.usc.edu/iemocap/index.html>

IEMOCAP is an audio-visual database of emotional expressions. The database includes acted videos accompanied by speech, motion capture of face, and text transcripts, which are annotated and classified into different emotional categories.

12) **International Skin Imaging Collaboration (ISIC)**:

➤ <http://isdis.net/isic-project/>

Melanoma project of the International Skin Imaging Collaboration has an open-source public access archive that includes dermatology images for skin lesion diagnosis purposes. Dermatology images lack standards; for this reason, creating a collection of dermatologic images under the ISIC Archive helps with both developing standards in skin imaging and material for developing clinical decision support algorithms.

13) **Japanese Society of Radiological Technology (JSRT) Database** [447]:

➤ <http://www.jsrt.or.jp/data/english/>

JSRT database is a chest image database, which includes both images with chest nodules and images without chest nodules. The images are accompanied by other data such as the age and gender of the patient, and the location of the nodule.

- **SCR database: Segmentation in Chest Radiographs** [448]:

➤ <http://www.isi.uu.nl/Research/Databases/SCR/>

SCR database is based on the JSRT database and contains chest radiograph images for segmentation purposes. The images available in the JSRT database are presented with their anatomical structures, which are broken down into separate sections.

14) **MEDLINE®/PubMed®**:

➤ <https://www.ncbi.nlm.nih.gov/pubmed>

Medical Literature Analysis and Retrieval System Online (MEDLINE) is a database of biomedical literature and life sciences that gathers their journal citations and abstracts and is maintained by United States National Library of Medicine. PubMed is the search engine that is based on MEDLINE and links the users to full text of literature when available.

15) **mPower: Mobile Parkinson Disease Study** [449]:

➤ <http://parkinsonmpower.org/>

mPower is a dataset for Parkinson's disease, which gathers its information through a mobile application by taking surveys from participants and recording mobile

phones' sensor data. The participants in the database are volunteers and anyone can join the study.

16) **National Biomedical Imaging Archive (NBIA)**:

➤ <https://imaging.nci.nih.gov/ncia/>

NBIA is a repository that is maintained by the National Cancer Institute. It gathers in vivo images in DICOM format; it allows searches to be performed on the images.

17) **National Database for Autism Research (NDAR)** [450]:

➤ <https://ndar.nih.gov/>

NDAR is a database that is maintained by the National Institute of Health. It contains clinical and image data on patients with autism. This database collects data both from labs and research papers and provides tools for users to perform searches.

18) **NeuroVault** [451]:

➤ <http://www.neurovault.org/>

NeuroVault is a repository for MRI and PET studies, which stores datasets that are created in different studies. The goal of the database is collecting and sharing statistical maps related to the human brain. The website also provides tools that helps researchers process/manipulate with the MRI and PET images.

19) **Open Access Series of Imaging Studies (OASIS)** [452]:

➤ <http://www.oasis-brains.org/>

The goal of the OASIS platform is to make brain MRI images freely available. The two datasets available include "MRI Data in Nondemented and Demented Older Adults" and "MRI Data in Young, Middle Aged, Nondemented and Demented Older Adults."

20) **Open fMRI** [453]:

➤ <https://openfmri.org/>

Open fMRI is a platform that collects and shares free raw magnetic resonance images in addition to EEG time series signals. Users can get data from the database free of charge and the project accepts new submitted datasets.

21) **Open-i (Open Access Biomedical Image Search Engine)** [454]:

➤ <https://openi.nlm.nih.gov/>

Open-i is a service provided by National Library of Medicine that is able to search and retrieve medical images from all available open source literature. The images, including charts, clinical images, graphs, etc, can both be searched with a text or an image query.

22) **Open PHACTS** [455]:

➤ <http://www.openphacts.org/>

Open PHACTS is a platform that links multiple pharmacological datasets and provides them in a unified medium. These datasets indicate the relationship between the compounds of different drugs, their targets, the diseases that they are used for, etc. The data can be used to discover new drugs or side effects of the existing drugs.

- 23) **Pedro Hispano Hospital (PH²) dataset** [456]:
 > <https://www.fc.up.pt/addi/ph2%20database.html>
 PH² database is collection of dermoscopic images of melanoma patients. The dermoscopic parameters of each image are assessed and annotated by an expert.
- 24) **PhysioNet** [457]:
 > <http://physionet.org/>
 PhysioNet is a collection of recorded physiological signals such as ECG data, gait and balance signals, image databases, and other forms of signals. The PhysioNet collection contains many subsets that are frequently used in the literature:
- **MIMIC-III** [458]:
 > <https://mimic.physionet.org/>
 Medical Information Mart for Intensive Care III is a database containing health-related data for patients who stayed in Critical Care Units.
 - **The Apnea – ECG database** [459]:
 A dataset of ECG recordings during sleep from sleep apnea patients. Waveforms in the database are annotated with apnea marks.
- 25) **Parkinson's Progression Markers Initiative (PPMI)** [460]:
 > <http://www.ppmi-info.org/>
 PPMI is a study that collects clinical and imaging data in addition to biological samples that are used to track the progression of Parkinson's disease. This data includes motor assessment, MRI imaging, DNA testing, plasma and urine collection, etc. It is available to academic researchers at no cost.
- 26) **SpineWeb**:
 > <http://spineweb.digitalimaginggroup.ca/>
 SpineWeb is a collection of datasets related to spinal images. This collection includes datasets with various types of images such as MRI or CT. SpineWeb also has tools related to the spinal images and publicly available at no cost.
- 27) **The Cancer Image Archive (TCIA)** [461]:
 > <http://www.cancerimagingarchive.net/>
 This cancer image archive is a collection of multiple cancer related image databases for public download. The datasets images are mostly in DICOM format and supporting data for the images such as treatment details and outcomes are accompanied by them. This platform is supported by the Fredrick National Laboratory for Cancer Research.
- 28) **Telemetric and Holter ECG Warehouse (THEW)** [2]:
 > <http://thew-project.org/>
 THEW is a database that has multiple sets of ECG data recordings related to different cardiovascular diseases. The datasets include patients with LQTS, chest pain, acute myocardial infarction, exercise test, etc.
- 29) **UCI machine learning repository** [3]:
 > <http://archive.ics.uci.edu/ml/>

UCI machine learning repository is a collection of datasets in different fields; some of these datasets are related to healthcare. Most notable medical datasets are as follows:

- **Diabetes 130-US Hospitals for Years 1999-2008 Data Set** [78]:
 A clinical care dataset of diabetic patients collected throughout 10 years within various US hospitals. The data is related to patients who have stayed in hospital for a duration of 1–14 days.
- **Thyroid Disease Data Set**:
 A combination of multiple datasets related to thyroid disease.
- **Parkinson's Telemonitoring Data Set** [462]:
 A dataset of voice measurements from patients with early-stage Parkinson's disease recorded through a six month trial by telemonitoring devices that captures the progression of the disease.
- **Diabetic Retinopathy Debrecen Data Set (The Messidor Database)** [463], [464]:
 > <http://www.adcis.net/en/Download-Third-Party/Messidor.html#download-en.php>
 A set of images focusing on diabetic retinopathy including eye fundus color images.
- **Parkinson Speech Dataset with Multiple Types of Sound Recordings Data Set** [147]:
 A collection of recorded sounds of patients with Parkinson's disease and healthy subjects.
- **Mammographic Mass Data Set** [465]:
 A dataset on mammographic masses images divided into benign and malignant cases.
- **Breast Cancer Wisconsin (Original) Data Set** [466]:
 699 of clinical cases for breast cancer.
- **Thoracic Surgery Data Data Set** [467]:
 A dataset related to post-operative life expectancy of patients with lung cancer, indicating whether the patient survived one year after their operation or not.
- **Heart Disease Data Set** [468]:
 920 records of data related to heart disease patients.

APPENDIX B MACHINE INTELLIGENCE TOOLS, PROGRAMS, AND LIBRARIES

Due to the significant amount of research that has been conducted in machine learning approaches in the past few decades, a rich set of open-source machine learning tools and libraries are available for testing newly-developed ideas in healthcare. A list of these tools along with a short description for each tools are provided in this section. While a large portion of these tools are applicable to a much wider range of applications, they can be used in health-care applications either right out of the box or with minor modifications.

- 1) **Caffe** [469]:
 - <http://caffe.berkeleyvision.org/>

Caffe is a deep learning framework written in C++. For computing platforms that incorporate GPUs, CUDA [321] version is also available. It has complete bindings to Python and MATLAB and is widely used in research projects that involve deep learning.
- 2) **DIGITS (NVIDIA Deep Learning GPU Training System)**:
 - <https://developer.nvidia.com/digits>

DIGITS is a deep neural networks (DNN) design tool for image classification and object detection tasks. DIGITS is an interactive tool, so there is no need for programming or debugging. It is an open source project and can be customized and extended to suit healthcare applications.
- 3) **ELKI** [470]:
 - <http://elki.dbs.ifi.lmu.de/>

Environment for Developing Knowledge Discovery in Database (KDD)-Applications Supported by Index-Structures (ELKI) is an open source software for data mining and is implemented in Java. The emphasis of ELKI is on clustering and anomaly detection.
- 4) **IBM SPSS Software**:
 - <http://www.ibm.com/analytics/us/en/technology/spss/>

IBM Social Package for the Social Sciences (SPSS) software is a platform that provides statistical analysis and machine learning algorithms in addition to other applications such as text analytics. Although it was at first designed for social sciences, it is now widely used in the healthcare industry. This tool is not free of charge, but provides discounted versions for academic purposes.
- 5) **LIBSVM** [471]:
 - <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

LIBSVM is an integrated software for various machine intelligence algorithms available in C++ and Java. It also has extensions in many programming languages such Python, R, Matlab, etc. It has many different Support Vector Machine (SVM) formulations implemented and provides a graphic interface to its users.
- 6) **Python Libraries**:

Some machine learning packages for the Python programming language are as follows:

 - **scikit-learn** [472]:
 - <http://scikit-learn.org/>

A machine learning library for Python, which has many implemented algorithms for tasks such as classification, regression, clustering, etc.
 - **PyBrain** [473]:
 - <http://pybrain.org/>

Python-Based Reinforcement Learning, Artificial Intelligence and Neural Network (PyBrain) library

is an open source easy-to-use library for Python that is best suited for building neural networks.
- **Orange** [474]:
 - <http://orange.biolab.si/>

Orange is a machine learning tool based on Python that has a visual programming front-end with interactive data visualization. This makes Orange a useful tool for smaller datasets that are plotted easily and also a useful tool for teaching.
- **PyMVPA** [475]:
 - <http://www.pymvpa.org/>

MultiVariate Pattern Analysis (MVPA) is machine intelligence Python package that is mostly suited to be used in the neuroimaging domain.
- **Theano** [476]:
 - <http://deeplearning.net/software/theano/>

Theano is a Python library for deep learning frameworks that can both utilize CPUs and GPUs. It provides support for optimized mathematical expressions used in machine intelligence algorithms, especially expressions with multi-dimensional arrays. Major developments for Theano have ceased.
- 7) **SHOGUN** [477]:
 - <http://www.shogun-toolbox.org/>

Shogun is an open source machine learning toolbox implemented in C++, which can provide efficient implementations of ML algorithms and can interface with many programming languages. It also runs natively under all major operating systems and provides APIs for most of standard algorithms including classifiers, regressors, and neural networks.
- 8) **SVM^{light}** [478]:
 - <http://svmlight.joachims.org/>

SVM^{light} implements Support Vector Machines (SVMs) in the C programming language and is free for scientific use. This tool only focuses on SVMs and provides a variety of support for them.

 - **SVM^{perf}** [479]:
 - http://www.cs.cornell.edu/People/tj/svm_light/svm_perf.html

SVM^{perf} is based on SVM^{light} and provides much faster training on large datasets.
- 9) **TensorFlow** [480]:
 - <https://www.tensorflow.org/>

TensorFlow is an open source library for data computation; it can implement neural networks efficiently. TensorFlow was originally developed by Google and has C, C++, and Python APIs. It works on various hardware structures such as CPUs, GPUs, or even mobile devices.

 - **SkFlow**:
 - <https://github.com/tensorflow/skflow>

SkFlow (Scikit Flow) is a combination of the scikit-learn library for Python and TensorFlow.

It provides a simplified interface for for TensorFlow and it is now integrated as a part of TensorFlow.

10) WEKA [481]:

➤ <http://www.cs.waikato.ac.nz/ml/weka>

The Waikato Environment for Knowledge Analysis (WEKA) is an open source suite written in Java that supports several standard data mining tasks, such as clustering, classification, regression, and feature selection.

REFERENCES

- [1] J. Manyika *et al.*, “*Big Data: The Next Frontier for Innovation, Competition, and Productivity*.” New York, NY, USA: McKinsey Global Institute, 2011.
- [2] J.-P. Couderc, “The telemetric and Holter ECG warehouse initiative (THEW): A data repository for the design, implementation and validation of ECG-related technologies,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., IEEE Eng. Med. Biol. Soc. Conf.*, 2010, p. 6252.
- [3] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [4] N. Council, *Frontiers in Massive Data Analysis*. 2013.
- [5] M. Hassanaliheragh, A. Page, T. Soyata, G. Sharma, and M. Aktas, “Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges,” in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2015, pp. 285–292.
- [6] O. Kocabas, T. Soyata, J.-P. Couderc, M. Aktas, J. Xia, and M. Huang, “Assessment of cloud-based health monitoring using homomorphic encryption,” in *Proc. IEEE 31st Int. Conf. Comput. Design (ICCD)*, Oct. 2013, pp. 443–446.
- [7] A. Page, S. Hijazi, D. Askan, B. Kantarci, and T. Soyata, “Research directions in cloud-based decision support systems for health monitoring using Internet-of-Things driven data acquisition,” *Int. J. Services Comput.*, vol. 4, no. 4, pp. 18–34, 2016.
- [8] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, 1993.
- [9] AmazonEC2. (2017). *Amazon Elastic Compute Cloud*. [Online]. Available: <https://aws.amazon.com/ec2/>
- [10] S. Hijazi, A. Page, B. Kantarci, and T. Soyata, “Machine learning in cardiac health monitoring and decision support,” *IEEE Comput. Mag.*, vol. 49, no. 11, pp. 38–48, Nov. 2016.
- [11] N. V. Thakor and Y.-S. Zhu, “Applications of adaptive filtering to ECG analysis: Noise cancellation and arrhythmia detection,” *IEEE Trans. Biomed. Eng.*, vol. 38, no. 8, pp. 785–794, Aug. 1991.
- [12] M. Bsoul, H. Minn, and L. Tamil, “Apnea MedAssist: Real-time sleep apnea monitor using single-lead ECG,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 3, pp. 416–427, May 2011.
- [13] H. S. Mousavi, V. Monga, G. Rao, and A. U. K. Rao, “Automated discrimination of lower and higher grade gliomas based on histopathological image analysis,” *J. Pathol. Inform.*, vol. 6, no. 1, p. 15, 2015.
- [14] A. Karargyris, O. Karargyris, and A. Pantelopoulou, “DERMA/care: An advanced image-processing mobile application for monitoring skin cancer,” in *Proc. IEEE 24th Int. Conf. Tools Artif. Intell.*, Nov. 2012, pp. 1–7.
- [15] S. Patel *et al.*, “Monitoring motor fluctuations in patients with Parkinson’s disease using wearable sensors,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 864–873, Nov. 2009.
- [16] S. Klöppel *et al.*, “Automatic classification of MR scans in Alzheimer’s disease,” *Brain*, vol. 131, no. 3, pp. 681–689, 2008.
- [17] H. Neuvirth *et al.*, “Toward personalized care management of patients at risk: The diabetes case study,” in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 395–403.
- [18] N. Koutsouleris *et al.*, “Use of neuroanatomical pattern classification to identify subjects in at-risk mental states of psychosis and predict disease transition,” *Arch. Gen. Psychiatry*, vol. 66, no. 7, pp. 700–712, 2009.
- [19] Y. Tamaki *et al.*, “Construction of a dental caries prediction model by data mining,” *J. Oral Sci.*, vol. 51, no. 1, pp. 61–68, 2009.
- [20] Y. M. Chae, S. H. Ho, K. W. Cho, D. H. Lee, and S. H. Ji, “Data mining approach to policy analysis in a health insurance domain,” *Int. J. Med. Inform.*, vol. 62, nos. 2–3, pp. 103–111, 2001.
- [21] J. Hoey, C. Boutilier, P. Poupart, P. Olivier, A. Monk, and A. Mihailidis, “People, sensors, decisions: Customizable and adaptive technologies for assistance in healthcare,” *ACM Trans. Interact. Intell. Syst.*, vol. 2, no. 4, Dec. 2012, Art. no. 20.
- [22] A. Mihailidis, J. N. Boger, T. Craig, and J. Hoey, “The COACH prompting system to assist older adults with dementia through handwashing: An efficacy study,” *BMC Geriatrics*, vol. 8, no. 1, p. 28, 2008.
- [23] I. P. Ktistakis and N. G. Bourbakis, “Assistive intelligent robotic wheelchairs,” *IEEE Potentials*, vol. 36, no. 1, pp. 10–13, Jan./Feb. 2017.
- [24] K. Turksoy, S. Samadi, J. Feng, E. Littlejohn, L. Quinn, and A. Cinar, “Meal detection in patients with type 1 diabetes: A new module for the multivariable adaptive artificial pancreas control system,” *IEEE J. Biomed. Health Inform.*, vol. 20, no. 1, pp. 47–54, Jan. 2016.
- [25] E. S. Sazonov, G. Fulk, N. Sazonova, and S. Schuckers, “Automatic recognition of postures and activities in stroke patients,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Sep. 2009, pp. 2200–2203.
- [26] M. Rabbi, M. H. Aung, M. Zhang, and T. Choudhury, “MyBehavior: Automatic personalized health feedback from user behaviors and preferences using smartphones,” in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 707–718.
- [27] D. S. Zois, M. Levorato, and U. Mitra, “Active classification for POMDPs: A Kalman-like state estimator,” *IEEE Trans. Signal Process.*, vol. 62, no. 23, pp. 6209–6224, Dec. 2014.
- [28] D. Zhou *et al.*, “Tackling mental health by integrating unobtrusive multimodal sensing,” in *Proc. AAAI*, 2015, pp. 1401–1409.
- [29] J. Wijsman, B. Grundlehner, H. Liu, H. Hermens, and J. Penders, “Towards mental stress detection using wearable physiological sensors,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2011, pp. 1798–1801.
- [30] N. P. Tatonetti *et al.*, “Detecting drug interactions from adverse-event reports: Interaction between paroxetine and pravastatin increases blood glucose levels,” *Clin. Pharmacol. Therapeutics*, vol. 90, no. 1, pp. 133–142, 2011.
- [31] B. Han, “Building a better disease detective,” *IEEE Spectr.*, vol. 52, no. 10, pp. 46–51, Oct. 2015.
- [32] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, “Detecting influenza epidemics using search engine query data,” *Nature*, vol. 457, no. 7232, pp. 1012–1014, 2009.
- [33] V. Chandola, S. R. Sukumar, and J. C. Schryver, “Knowledge discovery from massive healthcare claims data,” in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1312–1320.
- [34] A. Page, M. Hassanaliheragh, T. Soyata, M. K. Aktas, B. Kantarci, and S. Andreescu, “Conceptualizing a real-time remote cardiac health monitoring system,” in *Enabling Real-Time Mobile Cloud Computing through Emerging Technologies*, T. Soyata, Ed. Hershey, PA, USA: IGI Global, 2015, ch. 1, pp. 1–34.
- [35] D. S. Zois, M. Levorato, and U. Mitra, “Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks,” *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1581–1594, Apr. 2013.
- [36] M. Pouryazdan, B. Kantarci, T. Soyata, and H. Song, “Anchor-assisted and vote-based trustworthiness assurance in smart city crowdsensing,” *IEEE Access*, vol. 4, pp. 529–541, 2016.
- [37] A. Page, T. Soyata, J. P. Couderc, M. Aktas, B. Kantarci, and S. Andreescu, “Visualization of health monitoring data acquired from distributed sensors for multiple patients,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–7.
- [38] A. Page, M. K. Aktas, T. Soyata, W. Zareba, and J.-P. Couderc, “‘QT clock’ to improve detection of QT prolongation in long QT syndrome patients,” *Heart Rhythm*, vol. 13, no. 1, pp. 190–198, Jan. 2016.
- [39] M. Li *et al.*, “Multimodal physical activity recognition by fusing temporal and cepstral information,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 4, pp. 369–380, Aug. 2010.
- [40] J.-P. Y. Couderc, M. K. AKTAS, T. Soyata, and A. T. Page, “ECG clock electrocardiogram based diagnostic device and method,” U.S. Patent Appl. 15/368 587, Dec. 3, 2016.
- [41] R. H. Taylor, A. Menciassi, G. Fichtinger, P. Fiorini, and P. Dario, “Medical robotics and computer-integrated surgery,” in *Springer Handbook of Robotics*. Springer, 2008, pp. 1199–1222.

- [42] Retention. (2017). *Medical Record Retention and Media Formats for Medical Records*. [Online]. Available: <https://www.cms.gov/Outreach-and-Education/Medicare-Learning-Network-MLN/MLNMattersArticles/downloads/SE1022.pdf>
- [43] I. S. Ockene and N. H. Miller, "Cigarette smoking, cardiovascular disease, and stroke," *Circulation*, vol. 96, no. 9, pp. 3243–3247, 1997.
- [44] F. Badilini, "The ISHNE Holter standard output file format," *Ann. Non-invasive Electrocardiol.*, vol. 3, pp. 263–266, 1998.
- [45] W.-H. Weng, K. B. Waghlikar, A. T. McCray, P. Szolovits, and H. C. Chueh, "Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach," *BMC Med. Inform. Decis. Making*, vol. 17, p. 155, Dec. 2017.
- [46] M. K. Breitenstein, H. Liu, K. N. Maxwell, J. Pathak, and R. Zhang, "Electronic health record phenotypes for precision medicine: Perspectives and caveats from treatment of breast cancer at a single institution," *Clin. Transl. Sci.*, vol. 11, no. 1, pp. 85–92, 2018.
- [47] A. Page, T. Soyata, J.-P. Couderc, and M. K. Aktas, "An open source ECG clock generator for visualization of long-term cardiac monitoring data," *IEEE Access*, vol. 3, pp. 2704–2714, Dec. 2015.
- [48] S. Xu et al., "Soft microfluidic assemblies of sensors, circuits, and radios for the skin," *Science*, vol. 344, no. 6179, pp. 70–74, Apr. 2014.
- [49] D. Son et al., "Multifunctional wearable devices for diagnosis and therapy of movement disorders," *Nature Nanotechnol.*, vol. 9, pp. 397–404, Mar. 2014.
- [50] D.-H. Kim, R. Ghaffari, N. Lu, and J. A. Rogers, "Flexible and stretchable electronics for biointegrated devices," *Annu. Rev. Biomed. Eng.*, vol. 14, pp. 113–128, Aug. 2012.
- [51] E. G. Duffin, "Implantable monitor," U.S. Patent 6230059, May 8, 2001. [Online]. Available: <https://www.google.com/patents/US6230059>
- [52] M. Mitchell, C. Meyers, A.-I. A. Wang, and G. Tyson, "ContextProvider: Context awareness for medical monitoring applications," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug./Sep. 2011, pp. 5244–5247.
- [53] H.-T. Chu, C.-C. Huang, Z.-H. Lian, and J. J. P. Tsai, "A ubiquitous warning system for asthma-inducement," in *Proc. IEEE Int. Conf. Sensor Netw., Ubiquitous, Trustworthy Comput.*, vol. 2, Jun. 2006, pp. 186–191.
- [54] HIPAA. (1996). *Health Information Privacy*. [Online]. Available: <https://www.hhs.gov/hipaa/index.html>
- [55] O. Kocabas, T. Soyata, and M. K. Aktas, "Emerging security mechanisms for medical cyber physical systems," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 3, pp. 401–416, Jun. 2016.
- [56] S. N. Murphy and H. C. Chueh, "A security architecture for query tools used to access large biomedical databases," in *Proc. AMIA Symp.*, 2002, p. 552.
- [57] H. Gudbjartsson and S. Patz, "The Rician distribution of noisy MRI data," *Mag. Reson. Med.*, vol. 34, no. 6, pp. 910–914, 1995.
- [58] H. Lu, X. Li, I.-T. Hsiao, and Z. Liang, "Analytical noise treatment for low-dose CT projection data by penalized weighted least-square smoothing in the K-L domain," *Proc. SPIE*, vol. 4682, pp. 146–152, May 2002.
- [59] I. I. Christov and I. K. Daskalov, "Filtering of electromyogram artifacts from the electrocardiogram," *Med. Eng. Phys.*, vol. 21, no. 10, pp. 731–736, 1999.
- [60] P. He, G. Wilson, and C. Russell, "Removal of ocular artifacts from electro-encephalogram by adaptive filtering," *Med. Biol. Eng. Comput.*, vol. 42, no. 3, pp. 407–412, 2004.
- [61] A. Eklund, T. E. Nichols, and H. Knutsson, "Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 28, pp. 7900–7905, 2016.
- [62] C. L. Kimberlin and A. G. Winterstein, "Validity and reliability of measurement instruments used in research," *Amer. J. Health-Syst. Pharmacy*, vol. 65, no. 23, pp. 2276–2284, 2008.
- [63] A. Burton and D. G. Altman, "Missing covariate data within cancer prognostic studies: A review of current reporting and proposed guidelines," *Brit. J. Cancer*, vol. 91, no. 1, pp. 4–8, 2004.
- [64] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*, vol. 81. Hoboken, NJ, USA: Wiley, 2004.
- [65] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Mijsch, and B. Kavsek, "Informal identification of outliers in medical data," in *Proc. 5th Int. Workshop Intell. Data Anal. Med. Pharmacol.*, vol. 1, 2000, pp. 20–24.
- [66] G. Thatte et al., "Optimal time-resource allocation for energy-efficient physical activity detection," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1843–1857, Apr. 2011.
- [67] M. Swan, "Emerging patient-driven health care models: An examination of health social networks, consumer personalized medicine and quantified self-tracking," *Int. J. Environ. Res. Public Health*, vol. 6, no. 2, pp. 492–525, 2009.
- [68] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2001.
- [69] B. Logan et al., "Mel frequency cepstral coefficients for music modeling," in *Proc. Int. Symp. Music Inf. Retr.*, 2000, pp. 1–13.
- [70] A. Sano and R. W. Picard, "Stress recognition using wearable sensors and mobile phones," in *Proc. IEEE Humane Assoc. Conf. Affect. Comput. Intell. Interact. (ACII)*, Sep. 2013, pp. 671–676.
- [71] B. Xie and H. Minn, "Real-time sleep apnea detection by classifier combination," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 3, pp. 469–477, May 2012.
- [72] T. Pawar, S. Chaudhuri, and S. P. Duttgupta, "Body movement activity recognition for ambulatory cardiac monitoring," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 5, pp. 874–882, May 2007.
- [73] D. Álvarez, R. Hornero, D. Abásolo, F. del Campo, and C. Zamarrón, "Nonlinear characteristics of blood oxygen saturation from nocturnal oximetry for obstructive sleep apnea detection," *Physiol. Meas.*, vol. 27, no. 4, p. 399, 2006.
- [74] D. Sow, A. Biem, M. Blount, M. Ebling, and O. Verscheure, "Body sensor data processing using stream computing," in *Proc. Int. Conf. Multimedia Inf. Retr.*, 2010, pp. 449–458.
- [75] W.-J. Kang, J.-R. Shiu, C.-K. Cheng, J.-S. Lai, H.-W. Tsao, and T. S. Kuo, "The application of cepstral coefficients and maximum likelihood method in EMG pattern recognition," *IEEE Trans. Biomed. Eng.*, vol. 42, no. 8, pp. 777–785, Aug. 1995.
- [76] R. M. Rangayyan and Y. Wu, "Analysis of vibroarthrographic signals with features related to signal variability and radial-basis functions," *Ann. Biomed. Eng.*, vol. 37, no. 1, pp. 156–163, 2009.
- [77] B. Lei, S. A. Rahman, and I. Song, "Content-based classification of breath sound with enhanced features," *Neurocomputing*, vol. 141, pp. 139–147, Oct. 2014.
- [78] B. Strack et al., "Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records," *BioMed Res. Int.*, vol. 2014, Apr. 2014, Art. no. 781670.
- [79] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer-Verlag, 2006.
- [80] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature Extraction: Foundations and Applications*, vol. 207. Berlin, Germany: Springer, 2008.
- [81] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, no. 2, pp. 164–171, Jul. 2000.
- [82] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2002.
- [83] R. J. Martis, U. R. Acharya, K. M. Mandana, A. K. Ray, and C. Chakraborty, "Application of principal component analysis to ECG signals for automated diagnosis of cardiac health," *Expert Syst. Appl.*, vol. 39, no. 14, pp. 11792–11800, 2012.
- [84] L. Lukas et al., "Brain tumor classification based on long echo proton MRS signals," *Artif. Intell. Med.*, vol. 31, no. 1, pp. 73–89, 2004.
- [85] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [86] M. M. López et al., "SVM-based CAD system for early detection of the Alzheimer's disease using kernel PCA and LDA," *Neurosci. Lett.*, vol. 464, no. 3, pp. 233–238, 2009.
- [87] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, nos. 3–4, pp. 321–377, 1936.
- [88] X. Chen, Y. Wang, M. Nakanishi, X. Gao, T.-P. Jung, and S. Gao, "High-speed spelling with a noninvasive brain-computer interface," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 44, pp. E6058–E6067, 2015.
- [89] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [90] K. Fukunaga, *Introduction to Statistical Pattern Classification*. Cambridge, MA, USA: Academic, 1990.
- [91] C.-H. Jen, C.-C. Wang, B. C. Jiang, Y.-H. Chu, and M.-S. Chen, "Application of classification techniques on development an early-warning system for chronic illnesses," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 8852–8858, 2012.
- [92] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, no. 10, pp. 2385–2404, 2000.

- [93] B. M. Asl, S. K. Setarehdan, and M. Mohebbi, "Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal," *Artif. Intell. Med.*, vol. 44, no. 1, pp. 51–64, 2008.
- [94] A. Güven, K. Polat, S. Kara, and S. Güneş, "The effect of generalized discriminant analysis (GDA) to the classification of optic nerve disease from VEP signals," *Comput. Biol. Med.*, vol. 38, no. 1, pp. 62–68, 2008.
- [95] S. Güneş, K. Polat, and Ş. Yosunkaya, "Efficient sleep stage recognition system based on EEG signal using k -means clustering based feature weighting," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7922–7928, 2010.
- [96] S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski, "Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features," in *Proc. 5th IEEE Int. Symp. Biomed. Imag. (ISBI)*, May 2008, pp. 496–499.
- [97] T. F. Cox and M. A. Cox, "Multidimensional scaling," in *Handbook of Data Visualization*. London, U.K.: Chapman & Hall, 1994.
- [98] A. D. Cliff, P. Haggett, M. R. Smallman-Raynor, D. F. Stroup, and G. D. Williamson, "The application of multidimensional scaling methods to epidemiological data," *Stat. Methods Med. Res.*, vol. 4, no. 2, pp. 102–123, 1995.
- [99] I. V. E. Carlier, R. D. Lamberts, and B. P. R. Gersons, "The dimensionality of trauma: A multidimensional scaling comparison of police officers with and without posttraumatic stress disorder," *Psychiatry Res.*, vol. 97, no. 1, pp. 29–39, 2000.
- [100] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
- [101] G. Hamarneh, C. McIntosh, and M. S. Drew, "Perception-based visualization of manifold-valued medical images using distance-preserving dimensionality reduction," *IEEE Trans. Med. Imag.*, vol. 30, no. 7, pp. 1314–1327, Jul. 2011.
- [102] D. H. Ye, K. M. Pohl, and C. Davatzikos, "Semi-supervised pattern classification: Application to structural MRI of Alzheimer's disease," in *Proc. IEEE Int. Workshop Pattern Recognit. Neuroimag. (PRNI)*, May 2011, pp. 1–4.
- [103] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [104] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
- [105] J. Ge *et al.*, "Computer aided detection of clusters of microcalcifications on full field digital mammograms," *Med. Phys.*, vol. 33, no. 8, pp. 2975–2988, 2006.
- [106] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3687–3691.
- [107] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. ICML*, Washington, DC, USA, Aug. 2003, pp. 856–863.
- [108] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.
- [109] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997.
- [110] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 121–129.
- [111] T. Marill and D. Green, "On the effectiveness of receptors in recognition systems," *IEEE Trans. Inf. Theory*, vol. 9, no. 1, pp. 11–17, Jan. 1963.
- [112] S. Yu and L. Guan, "A CAD system for the automatic detection of clustered microcalcifications in digitized mammogram films," *IEEE Trans. Med. Imag.*, vol. 19, no. 2, pp. 115–126, Feb. 2000.
- [113] J. Wagner, J. Kim, and E. André, "From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2005, pp. 940–943.
- [114] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Comput.*, vol. C-20, no. 9, pp. 1100–1103, Sep. 1971.
- [115] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Univ. Waikato, Hamilton, New Zealand, 1999.
- [116] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. ICML*, vol. 3, 2003, pp. 856–863.
- [117] R. F. Walker, P. Jackway, B. Lovell, and I. D. Longstaff, "Classification of cervical cell nuclei using morphological segmentation and textural feature extraction," in *Proc. Austral. New Zealand Intell. Inf. Syst. Conf. (ANZIS)*, Nov./Dec. 1994, pp. 297–301.
- [118] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.
- [119] Y. Sun, C. F. Babbs, and E. J. Delp, "A comparison of feature selection methods for the detection of breast cancers in mammograms: Adaptive sequential floating search vs. genetic algorithm," in *Proc. 27th Annu. IEEE Int. Conf. Eng. Med. Biol. Soc. (EMBS)*, Jan. 2006, pp. 6532–6535.
- [120] H. Ganster, P. Pinz, R. Rohrer, E. Wildling, M. Binder, and H. Kittler, "Automated melanoma recognition," *IEEE Trans. Med. Imag.*, vol. 20, no. 3, pp. 233–239, Mar. 2001.
- [121] O. M. Mozos *et al.*, "Stress detection using wearable physiological and sociometric sensors," *Int. J. Neural Syst.*, vol. 27, no. 2, p. 1650041, 2017.
- [122] T. H. Cheng, C. P. Wei, and V. S. Tseng, "Feature selection for medical data mining: Comparisons of expert judgment and automatic approaches," in *Proc. 19th IEEE Int. Symp. Comput.-Based Med. Syst. (CBMS)*, Jun. 2006, pp. 165–170.
- [123] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, nos. 2–3, pp. 95–99, 1988.
- [124] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Intell. Syst. Appl.*, vol. 13, no. 2, pp. 44–49, Mar. 1998.
- [125] M. Gletsos, S. G. Mouggiakakou, G. K. Matsopoulos, K. S. Nikita, A. S. Nikita, and D. Kelekis, "A computer-aided diagnostic system to characterize CT focal liver lesions: Design and optimization of a neural network classifier," *IEEE Trans. Inf. Technol. Biomed.*, vol. 7, no. 3, pp. 153–162, Sep. 2003.
- [126] A. P. Dhawan, Y. Chitre, and C. Kaiser-Bonasso, "Analysis of mammographic microcalcifications using gray-level image structure features," *IEEE Trans. Med. Imag.*, vol. 15, no. 3, pp. 246–259, Jun. 1996.
- [127] Z. Ori, G. Monir, J. Weiss, X. Sayhouni, and D. H. Singer, "Heart rate variability. frequency domain analysis," *Cardiol. Clin.*, vol. 10, no. 3, pp. 499–537, 1992.
- [128] V. Srinivasan, C. Eswaran, and A. Sriraam, "Artificial neural network based epileptic detection using time-domain and frequency-domain features," *J. Med. Syst.*, vol. 29, no. 6, pp. 647–660, Dec. 2005.
- [129] B. Zheng, W. Qian, and L. P. Clarke, "Digital mammography: Mixed feature neural network with spectral entropy decision for detection of microcalcifications," *IEEE Trans. Med. Imag.*, vol. 15, no. 5, pp. 589–597, Oct. 1996.
- [130] J. Krajewski and B. J. Kröger, "Using prosodic and spectral characteristics for sleepiness detection," in *Proc. INTERSPEECH*, 2007, pp. 1841–1844.
- [131] A. V. Oppenheim and R. W. Schaffer, "From frequency to quefrequency: A history of the cepstrum," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 95–106, Sep. 2004.
- [132] H. C. Bazett, "An analysis of the time-relations of electrocardiograms," *Heart*, vol. 7, no. 2, pp. 353–370, 1920.
- [133] L. S. Fridericia, "Die systolendauer im elektrokardiogramm bei normalen menschen und bei herzkranken," *Acta Med. Scandinavica*, vol. 53, no. 1, pp. 469–486, 1920.
- [134] A. Subasi, "EEG signal classification using wavelet feature extraction and a mixture of expert model," *Expert Syst. Appl.*, vol. 32, no. 4, pp. 1084–1093, 2007.
- [135] A. Quetelet, *Sur L'homme et le Développement de ses Facultés ou Essai de Physique Sociale*, vol. 1. Paris, France: Bachelier, 1835.
- [136] F. Rohrer, *The Index Corpulence as Measure Nutritional State*. Munich, Germany: Münchner Med, 1921.
- [137] N. Ng, R. A. Gabriel, J. McAuley, C. Elkan, and Z. C. Lipton. (2017). "Predicting surgery duration with neural heteroscedastic regression." [Online]. Available: <https://arxiv.org/abs/1702.05386>
- [138] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization," *Future Generat. Comput. Syst.*, vol. 66, pp. 30–35, Jan. 2017.
- [139] A. Kushki, A. Khan, J. Brian, and E. Anagnostou, "A Kalman filtering framework for physiological detection of anxiety-related arousal in children with autism spectrum disorder," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 3, pp. 990–1000, Mar. 2015.
- [140] J. A. Swets, "The relative operating characteristic in psychology," *Science*, vol. 182, no. 4116, pp. 990–1000, 1973.

- [141] B. Ustun, M. B. Westover, C. Rudin, and M. T. Bianchi, "Clinical prediction models for sleep apnea: The importance of medical history over symptoms," *J. Clin. Sleep Med. Off. Publication Amer. Acad. Sleep Med.*, vol. 12, no. 2, pp. 161–168, 2016.
- [142] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. 4th Int. Conf. Mach. Learn.*, vol. 97. Nashville, TN, USA, Jul. 1997, pp. 179–186.
- [143] C. T. Su, P. C. Wang, Y. C. Chen, and L. F. Chen, "Data mining techniques for assisting the diagnosis of pressure ulcer development in surgical patients," *J. Med. Syst.*, vol. 36, no. 4, pp. 2387–2399, 2012.
- [144] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
- [145] S. S. M. Salehi, D. Erdogmus, and A. Gholipour. (2017). "Auto-context convolutional neural network (auto-net) for brain extraction in magnetic resonance imaging." [Online]. Available: <https://arxiv.org/abs/1703.02083>
- [146] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta-Protein Struct.*, vol. 405, no. 2, pp. 442–451, Oct. 1975.
- [147] B. E. Sakar et al., "Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 4, pp. 828–834, Jul. 2013.
- [148] J. Cohen, "A coefficient of agreement for nominal scales," *Edu. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.
- [149] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," *J. Neural Eng.*, vol. 14, no. 1, p. 016003, 2017.
- [150] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [151] I. Kurt, M. Ture, and A. T. Kurum, "Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 366–374, 2008.
- [152] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "EEG data classification using wavelet features selected by Wilcoxon statistics," *Neural Comput. Appl.*, vol. 26, no. 5, pp. 1193–1202, 2015.
- [153] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure–activity relationships," *J. Chem. Inf. Model.*, vol. 55, no. 2, pp. 263–274, 2015.
- [154] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L. J. Wei, "On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data," *Statist. Med.*, vol. 30, no. 10, pp. 1105–1117, 2011.
- [155] A. Khosla, Y. Cao, C. C. Y. Lin, H. K. Chiu, J. Hu, and H. Lee, "An integrated machine learning approach to stroke prediction," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 183–192.
- [156] H. Chen, X. Qi, L. Yu, and P. A. Heng. (2016). "DCAN: Deep contour-aware networks for accurate gland segmentation." [Online]. Available: <https://arxiv.org/abs/1604.02677>
- [157] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
- [158] H. C. Shin, K. Roberts, L. Lu, D. Demner-Fushman, J. Yao, and R. M. Summers. (2016). "Learning to read chest X-rays: Recurrent neural cascade model for automated image annotation." [Online]. Available: <https://arxiv.org/abs/1603.08486>
- [159] NICHD Early Child Care Research Network, "Child-care structure → process → outcome: Direct and indirect effects of child-care quality on young children's development," *Psychol. Sci.*, vol. 13, no. 3, pp. 2016.
- [160] S. Lemeshow and D. W. Hosmer, "A review of goodness of fit statistics for use in the development of logistic regression models," *Amer. J. Epidemiol.*, vol. 115, no. 1, pp. 92–106, 1982.
- [161] K. R. Hansen et al., "Predictors of pregnancy and live-birth in couples with unexplained infertility after ovarian stimulation–intrauterine insemination," *Fertility Sterility*, vol. 105, no. 6, pp. 1575–1583, 2016.
- [162] F. J. Massey, Jr., "The Kolmogorov–Smirnov test for goodness of fit," *J. Amer. Statist. Assoc.*, vol. 46, no. 253, pp. 68–78, 1951.
- [163] T. J. Chen, K. S. Chuang, W. Wu, and Y. R. Lu, "Compressed medical image quality determination using the Kolmogorov–Smirnov test," *Current Med. Imag. Rev.*, vol. 13, no. 2, pp. 204–209, 2017.
- [164] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Predicting healthcare trajectories from medical records: A deep learning approach," *J. Biomed. Inform.*, vol. 69, pp. 218–229, May 2017.
- [165] N. K. Francis et al., "The use of artificial neural networks to predict delayed discharge and readmission in enhanced recovery following laparoscopic colorectal cancer surgery," *Techn. Coloproctol.*, vol. 19, no. 7, pp. 419–428, 2015.
- [166] E. Yom-Tov, "Predicting drug recalls from Internet search engine queries," *IEEE J. Transl. Eng. Health Med.*, vol. 5, 2017, Art. no. 4400106
- [167] P. Schulam and S. Saria, "A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 748–756.
- [168] D. Conforti, F. Guerriero, R. Guido, M. M. Cerinic, and M. L. Conforti, "An optimal decision making model for supporting week hospital management," *Health Care Manage. Sci.*, vol. 14, no. 1, pp. 74–88, 2011.
- [169] Q. Li, R. G. Mark, and G. D. Clifford, "Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter," *Physiol. Meas.*, vol. 29, no. 1, p. 15, 2007.
- [170] G. C. Sharp, S. B. Jiang, S. Shimizu, and H. Shirato, "Prediction of respiratory tumour motion for real-time image-guided radiotherapy," *Phys. Med. Biol.*, vol. 49, no. 3, p. 425, 2004.
- [171] L. Wang, B. Li, and L. F. Tian, "EGGDD: An explicit dependency model for multi-modal medical image fusion in shift-invariant shearlet transform domain," *Inf. Fusion*, vol. 19, pp. 29–37, Sep. 2014.
- [172] O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts, "Determining the acceptance of cadaveric livers using an implicit model of the waiting list," *Oper. Res.*, vol. 55, no. 1, pp. 24–36, 2007.
- [173] B. Courcelle, "Graph rewriting: An algebraic and logic approach," in *Handbook of Theoretical Computer Science*. Amsterdam, The Netherlands: Elsevier, 1990, pp. 193–242.
- [174] M. R. Sleep, M. J. Plasmeyjer, and M. C. van Eekelen, *Semagraph: The Theory and Practice of Term Graph Rewriting*. Chichester, U.K.: Wiley, 1993.
- [175] H. Ehrig and G. Rozenberg, *Handbook of Graph Grammars and Computing by Graph Transformation*, vol. 3. Singapore: World Scientific, 1999.
- [176] R. Heckel, "Graph transformation in a nutshell," *Electron. Notes Theor. Comput. Sci.*, vol. 148, no. 1, pp. 187–198, 2006.
- [177] W. S. Hlavacek, J. R. Faeder, M. L. Blinov, R. G. Posner, M. Hucka, and W. Fontana, "Rules for modeling signal-transduction systems," *Science*, vol. 2006, no. 344, p. re6, 2006.
- [178] A. F. M. Smith and M. West, "Monitoring renal transplants: An application of the multiprocess Kalman filter," *Biometrics*, vol. 39, no. 4, pp. 867–878, 1983.
- [179] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm," *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 45–57, Jan. 2001.
- [180] A. Wittek, K. Miller, R. Kikinis, and S. K. Warfield, "Patient-specific model of brain deformation: Application to medical image registration," *J. Biomech.*, vol. 40, no. 4, pp. 919–929, 2007.
- [181] A. R. Cassandra, "A survey of POMDP applications," in *Proc. Work. Notes AAAI Fall Symp. Planning Partially Observable Markov Decision Process.*, vol. 1724, 1998, pp. 1–9.
- [182] D. J. White, *Markov Decision Processes*. Hoboken, NJ, USA: Wiley, 1993.
- [183] D. P. Bertsekas, *Dynamic Systems Optimal Control*, vol. 1. Belmont, MA, USA: Athena Scientific, 2005.
- [184] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.
- [185] V. Krishnamurthy, *Partially Observed Markov Decision Processes*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
- [186] A. Mahajan and D. Teneketzis, *Multi-Armed Bandit Problems*. New York, NY, USA: Springer, 2008, pp. 121–151.
- [187] J. Gittins, K. Glazebrook, and R. Weber, *Multi-Armed Bandit Allocation Indices*. Hoboken, NJ, USA: Wiley, 2011.
- [188] U. Mitra et al., "KNOWME: A case study in wireless body area sensor network design," *IEEE Commun. Mag.*, vol. 50, no. 5, pp. 116–125, May 2012.
- [189] P. Paredes, R. Gilad-Bachrach, M. Czerwinski, A. Roseway, K. Rowan, and J. Hernandez, "PopTherapy: Coping with stress through pop-culture," in *Proc. 8th Int. Conf. Pervasive Comput. Technol. Healthcare (PervasiveHealth)*, 2014, pp. 109–117.

- [190] H. Yu, "Approximate solution methods for partially observable Markov and semi-Markov decision processes," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2006.
- [191] M. Hauskrecht and H. Fraser, "Planning treatment of ischemic heart disease with partially observable Markov decision processes," *Artif. Intell. Med.*, vol. 18, no. 3, pp. 221–244, 2000.
- [192] M. Zabel, B. Acar, T. Klingenheben, M. R. Franz, S. H. Hohnloser, and M. Malik, "Analysis of 12-lead T-wave morphology for risk stratification after myocardial infarction," *Circulation*, vol. 102, no. 11, pp. 1252–1257, 2000.
- [193] L. Ljung, "System identification," in *Signal Analysis and Prediction*. Boston, MA, USA: Springer, 1998, pp. 163–173.
- [194] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
- [195] L. Ljung, "Perspectives on system identification," *Annu. Rev. Control*, vol. 34, no. 1, pp. 1–12, 2010.
- [196] O. Nelles, *Nonlinear System Identification From Classical Approaches to Neural Networks and Fuzzy Models*. Berlin, Germany: Springer, 2013.
- [197] Y. Li, H.-L. Wei, S. A. Billings, and P. G. Sarrigiannis, "Identification of nonlinear time-varying systems using an online sliding-window and common model structure selection (CMSS) approach with applications to EEG," *Int. J. Syst. Sci.*, vol. 47, no. 11, pp. 2671–2681, 2016.
- [198] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.
- [199] V. Kontis, J. E. Bennett, C. D. Mathers, G. Li, K. Foreman, and M. Ezzati, "Future life expectancy in 35 industrialised countries: Projections with a Bayesian model ensemble," *Lancet*, vol. 389, no. 10076, pp. 1323–1335, 2017.
- [200] B. Settles, "Active learning," *Synth. Lect. Intell. Mach. Learn.*, vol. 6, no. 1, pp. 1–114, 2012.
- [201] M. Kholghi, L. Sitbon, G. Zuccon, and A. Nguyen, "Active learning: A step towards automating medical concept extraction," *J. Amer. Med. Inform. Assoc.*, vol. 23, no. 2, pp. 289–296, 2015.
- [202] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "GRAM: Graph-based attention model for healthcare representation learning," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 787–795.
- [203] F. Pukelsheim, *Optimal Design of Experiments*. Philadelphia, PA, USA: SIAM, 2006.
- [204] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Stat. Sci.*, vol. 10, no. 3, pp. 273–304, 1995.
- [205] A. Wald, "Sequential tests of statistical hypotheses," *Ann. Math. Statist.*, vol. 16, no. 2, pp. 117–186, 1945.
- [206] H. Chernoff, *Sequential Analysis and Optimal Design*. Philadelphia, PA, USA: SIAM, 1972.
- [207] T. L. Lai, *Sequential Analysis*. Hoboken, NJ, USA: Wiley, 2001.
- [208] G. C. Goodwin and R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*. Cambridge, MA, USA: Academic, 1977.
- [209] E. Walter and L. Pronzato, *Identification of Parametric Models From Experimental Data*. New York, NY, USA: Springer-Verlag, 1997.
- [210] M. C. Priess, J. Choi, C. Radcliffe, J. M. Popovich, N. P. Cholewicki, and J. and Reeves, "Time-domain optimal experimental design in human seated postural control testing," *J. Dyn. Syst., Meas., Control*, vol. 137, no. 5, p. 054501, 2015.
- [211] N. R. Draper and H. Smith, *Applied Regression Analysis*. Hoboken, NJ, USA: Wiley, 2014.
- [212] S. Chatterjee and A. S. Hadi, *Regression Analysis by Example*. Hoboken, NJ, USA: Wiley, 2015.
- [213] D. A. Freedman, *Statistical Models: Theory and Practice*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [214] J. S. Armstrong, "Illusions in regression analysis," *Int. J. Forecasting*, vol. 3, no. 28, pp. 689–694, 2012.
- [215] G. A. F. Seber and A. J. Lee, *Linear Regression Analysis*, vol. 936. Hoboken, NJ, USA: Wiley, 2012.
- [216] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, vol. 821. Hoboken, NJ, USA: Wiley, 2012.
- [217] D. M. Bates and D. G. Watts, *Nonlinear Regression Analysis and its Applications*, vol. 2. Hoboken, NJ, USA: Wiley, 1988.
- [218] N. Cressie, "The origins of kriging," *Math. Geol.*, vol. 22, no. 3, pp. 239–252, 1990.
- [219] J. Wolberg, *Data Analysis Using the Method of Least Squares: Extracting the Most Information from Experiments*. Berlin, Germany: Springer, 2006.
- [220] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk, *A Distribution-Free Theory Nonparametric Regression*. New York, NY, USA: Springer, 2006.
- [221] M. R. Segal, "Tree-structured methods for longitudinal data," *J. Amer. Stat. Assoc.*, vol. 87, no. 418, pp. 407–418, 1992.
- [222] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.
- [223] J. A. Mitchell *et al.*, "Physical activity and pediatric obesity: A quantile regression analysis," *Med. Sci. Sports Exerc.*, vol. 49, no. 3, pp. 466–473, 2017.
- [224] G. Wei *et al.*, "Lung nodule classification using local kernel regression models with out-of-sample extension," *Biomed. Signal Process. Control*, vol. 40, pp. 1–9, Feb. 2018.
- [225] A. Agresti, *An Introduction to Categorical Data Analysis*, vol. 135. New York, NY, USA: Wiley, 1996.
- [226] B. G. Tabachnick, L. S. Fidell, and S. J. Osterlind, *Using Multivariate Statistics*. Boston, MA, USA: Allyn & Bacon, 2001.
- [227] A. Christin, C. Akre, A. Berchtold, and J. C. Suris, "Parent-adolescent relationship in youths with a chronic condition," *Child, Care, Health Develop.*, vol. 42, no. 1, pp. 36–41, 2016.
- [228] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected Papers of Hirotugu Akaike*. New York, NY, USA: Springer, 1998, pp. 199–213.
- [229] C. Vigen *et al.*, "Validation of self-reported comorbidity status of breast cancer patients with medical records: The California breast cancer survivorship consortium (CBCSC)," *Cancer Causes Control*, vol. 27, no. 3, pp. 391–401, 2016.
- [230] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.
- [231] K. Ide, Y. Kawasaki, H. Yamada, and N. Masaki, "Regional differences in hepatitis C treatment with peginterferon and ribavirin in Japan: A retrospective cohort study," *Drug Des., Develop. Therapy*, vol. 10, pp. 1217–1223, Mar. 2016.
- [232] H. Mamiya, K. Schwartzman, A. Verma, C. Jauvin, M. Behr, and D. Buckeridge, "Towards probabilistic decision support in public health practice: Predicting recent transmission of tuberculosis from patient attributes," *J. Biomed. Inform.*, vol. 53, pp. 237–242, Feb. 2015.
- [233] D. J. Spiegelhalter, N. G. Best, B. P. Carlin, and A. van der Linde, "The deviance information criterion: 12 years on," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 76, no. 3, pp. 485–493, 2014.
- [234] I. M. Blake, P. Chenoweth, H. Okayasu, C. A. Donnelly, R. B. Aylward, and N. C. Grassly, "Faster detection of poliomyelitis outbreaks to support polio eradication," *Emerg. Infectious Diseases*, vol. 22, no. 3, p. 449, 2016.
- [235] G. Claeskens and N. L. Hjort, "The focused information criterion," *J. Amer. Stat. Assoc.*, vol. 98, no. 464, pp. 900–916, 2003.
- [236] H. Yang, Y. Liu, and H. Liang, "Focused information criterion on predictive models in personalized medicine," *Biometrical J.*, vol. 57, no. 3, pp. 422–440, 2015.
- [237] R. E. Kass and A. E. Raftery, "Bayes factors," *J. Amer. Statist. Assoc.*, vol. 90, no. 430, pp. 773–795, 1995.
- [238] A. R. Kaup *et al.*, "Trajectories of depressive symptoms in older adults and risk of dementia," *JAMA Psychiatry*, vol. 73, no. 5, pp. 525–531, 2016.
- [239] C. P. Nelson *et al.*, "Association analyses based on false discovery rate implicate new loci for coronary artery disease," *Nature Genet.*, vol. 49, no. 9, pp. 1385–1391, 2017.
- [240] C. L. Mallows, "Some comments on C_p ," *Technometrics*, vol. 15, no. 4, pp. 661–675, 1973.
- [241] A. Luzak *et al.*, "Which early life events or current environmental and lifestyle factors influence lung function in adolescents?—Results from the GINIplus & LISAPlus studies," *Respiratory Res.*, vol. 18, no. 1, p. 138, 2017.
- [242] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.
- [243] R. A. Stine, "Model selection using information theory and the MDL principle," *Sociol. Methods Res.*, vol. 33, no. 2, pp. 230–260, 2004.
- [244] R. J. van Sloun, L. Demi, A. W. Postema, J. J. de la Rosette, H. Wijkstra, and M. Mischi, "Ultrasound-contrast-agent dispersion and velocity imaging for prostate cancer localization," *Med. Image Anal.*, vol. 35, pp. 610–619, Jan. 2017.
- [245] C. S. Wallace and D. M. Boulton, "An information measure for classification," *Comput. J.*, vol. 11, no. 2, pp. 185–194, 1968.

- [246] S. Ameli, F. Naghdy, D. Stirling, G. Naghdy, and M. Aghmesheh, "Objective clinical gait analysis using inertial sensors and six minute walking test," *Pattern Recognit.*, vol. 63, pp. 246–257, Mar. 2017.
- [247] V. N. Vapnik and A. J. Chervonenkis, *Theory of Pattern Recognition*. Moscow, Russia: Nauka, 1974.
- [248] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," in *Measures of Complexity*. Cham, Switzerland: Springer, 2015, pp. 11–30.
- [249] L. Wang, S. W. Su, B. G. Celler, G. S. Chan, T. M. Cheng, and A. V. Savkin, "Assessing the human cardiovascular response to moderate exercise: feature extraction by support vector regression," *Physiol. Meas.*, vol. 30, no. 3, pp. 227–244, 2009.
- [250] M. A. Grandner and J. W. Winkelman, "Nocturnal leg cramps: Prevalence and associations with demographics, sleep disturbance symptoms, medical conditions, and cardiometabolic risk factors," *PLoS ONE*, vol. 12, no. 6, p. e0178465, 2017.
- [251] T. P. Hettmansperger and J. W. McKean, *Robust Nonparametric Statistical Methods*. Boca Raton, FL, USA: CRC Press, 2010.
- [252] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*. Hoboken, NJ, USA: Wiley, 2013.
- [253] S. M. Kay, *Fundamentals of Statistical Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [254] J. M. Bernardo and A. F. Smith, *Bayesian Theory*. Hoboken, NJ, USA: Wiley, 1994.
- [255] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control* (Signal Processing Series). Upper Saddle River, NJ, USA: Prentice-Hall, 1995.
- [256] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.
- [257] H. L. Van Trees, *Detection, Estimation, and Filtering Theory*. Hoboken, NJ, USA: Wiley, 2004.
- [258] C. H. Van Houtven and E. C. Norton, "Informal care and health care use of older adults," *J. Health Econ.*, vol. 23, no. 6, pp. 1159–1180, 2004.
- [259] C. Poletto, C. Pelat, D. Levy-Bruhl, Y. Yazdanpanah, P. Y. Boelle, and V. Colizza, "Assessment of the Middle East respiratory syndrome coronavirus (MERS-CoV) epidemic in the Middle East and risk of international spread using a novel maximum likelihood analysis approach," *Eurosurveillance*, vol. 19, no. 23, p. 3, 2014.
- [260] P. Schulam, F. Wigley, and S. Saria, "Clustering longitudinal clinical marker trajectories from electronic health data: Applications to phenotyping and endotype discovery," in *Proc. AAAI*, 2015, pp. 2956–2964.
- [261] B. M. Marlin, D. C. Kale, R. G. Khemani, and R. C. Wetzel, "Unsupervised pattern discovery in electronic health care data using probabilistic clustering models," in *Proc. 2nd ACM SIGHT Int. Health Inform. Symp.*, 2012, pp. 389–398.
- [262] X. Wang, W. Xu, and Z. Jin, "A hidden Markov model based dynamic scheduling approach for mobile cloud telemonitoring," in *Proc. IEEE EMBS Int. Conf. Biomed. Health Inform. (BHI)*, Feb. 2017, pp. 273–276.
- [263] R. L. Stratonovich, "Conditional Markov processes," *Theory Probab. Appl.*, vol. 5, no. 2, pp. 156–178, 1960.
- [264] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 1966.
- [265] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [266] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*, vol. 29. New York, NY, USA: Springer, 2008.
- [267] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [268] J. Hu, M. K. Brown, and W. Turin, "HMM based online handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 1039–1045, Oct. 1996.
- [269] J. Söding, "Protein homology detection by HMM–HMM comparison," *Bioinformatics*, vol. 21, no. 7, pp. 951–960, 2004.
- [270] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Comput.*, vol. 9, no. 1, pp. 48–53, Jan./Mar. 2010.
- [271] Y. Qin, W. Qian, N. Shojaati, and N. Osgood, "Identifying smoking from smartphone sensor data and multivariate hidden Markov models," in *Proc. Int. Conf. Social Comput., Behav.-Cultural Modeling Prediction Behav. Represent. Modeling Simulation*. Cham, Switzerland: Springer, 2017, pp. 230–235.
- [272] J. Son, P. F. Brennan, and S. Zhou, "Correlated gamma-based hidden Markov model for the smart asthma management based on rescue inhaler usage," *Statist. Med.*, vol. 36, no. 10, pp. 1619–1637, 2017.
- [273] P.-N. Yu, S. A. Naini, C. N. Heck, C. Y. Liu, D. Song, and T. W. Berger, "A sparse Laguerre-Volterra autoregressive model for seizure prediction in temporal lobe epilepsy," in *Proc. IEEE 38th Annu. Int. Conf. Eng. Med. Biol. Soc. (EMBC)*, Aug. 2016, pp. 1664–1667.
- [274] Y. Zhang, B. Liu, X. Ji, and D. Huang, "Classification of EEG signals based on autoregressive model and wavelet packet decomposition," *Neural Process. Lett.*, vol. 45, no. 2, pp. 365–378, 2017.
- [275] B. Williams, P. S. Lacy, P. Yan, C. N. Hwee, C. Liang, and C. M. Ting, "Development and validation of a novel method to derive central aortic systolic pressure from the radial pressure waveform using an N-point moving average method," *J. Amer. College Cardiol.*, vol. 57, no. 8, pp. 951–961, 2011.
- [276] P. Whittle, *Prediction and Regulation by Linear Least-Square Methods*. Oxford, U.K.: English Univ. Press, 1963.
- [277] E. J. Hannan, *Multiple Time Series Models*, vol. 38. Hoboken, NJ, USA: Wiley, 2009.
- [278] S. M. Imaduddin and T. Heldt, "Model-based estimation of radial artery blood pressure from recordings of the Nexfin monitor," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 1692–1695.
- [279] S. Revels, S. A. P. Kumar, and O. Ben-Assuli, "Predicting obesity rate and obesity-related healthcare costs using data analytics," *Health Policy Technol.*, vol. 6, no. 2, pp. 198–207, 2017.
- [280] Y. Lyu et al., "Dynamic evaluation model of coronary heart disease for ubiquitous healthcare," *Comput. Ind.*, vol. 69, pp. 35–44, May 2015.
- [281] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*. New York, NY, USA: Springer, 2013.
- [282] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.
- [283] M. H. In, S. Y. Lee, T. S. Park, T. S. Kim, M. H. Cho, and Y. B. Ahn, "Ballistocardiogram artifact removal from EEG signals using adaptive filtering of EOG signals," *Physiol. Meas.*, vol. 27, no. 11, pp. 1227–1240, 2006.
- [284] E. J. Knobbe and B. Buckingham, "The extended Kalman filter for continuous glucose monitoring," *Diabetes Technol. Therapeutics*, vol. 7, no. 1, pp. 15–27, 2005.
- [285] K. Punithakumar, I. B. Aayed, A. Islam, I. G. Ross, and S. Li, "Regional heart motion abnormality detection via information measures and unscented Kalman filtering," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervent.* Berlin, Germany: Springer, 2010, pp. 409–417.
- [286] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," McMaster University, Hamilton, ON, Canada, Tech. Rep., 2003.
- [287] P. Magni, S. Quaglioni, M. Marchetti, and G. Barosi, "Deciding when to intervene: A Markov decision process approach," *Int. J. Med. Inform.*, vol. 60, no. 3, pp. 237–253, 2000.
- [288] J. Mason. (2013). *Using Electronic Health Records to Monitor and Improve Adherence to Medication*. [Online]. Available: http://people.virginia.edu/~jem4yb/Papers/Mason_2013.pdf
- [289] A. Eitan, *Constrained Markov Decision Processes*, vol. 7. Boca Raton, FL, USA: CRC Press, 1999.
- [290] Y. Wang, B. Krishnamachari, Q. Zhao, and M. Annavaram, "Markov-optimal sensing policy for user state estimation in mobile devices," in *Proc. ACM 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2010, pp. 268–278.
- [291] Y. Wang, B. Krishnamachari, and M. Annavaram, "Semi-Markov state estimation and policy optimization for energy efficient mobile sensing," in *Proc. 9th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw. (SECON)*, Jun. 2012, pp. 533–541.
- [292] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, Mar. 1985.
- [293] J. Langford and T. Zhang, "The epoch-greedy algorithm for multi-armed bandits with side information," in *Proc. NIPS*, 2008, pp. 817–824.
- [294] J. Mickova, "Stochastic scheduling with multi-armed bandits," Ph.D. dissertation, Dept. Elect. Electron. Eng., Univ. Melbourne, Parkville, VIC, Australia, 2000.
- [295] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and non-stochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

- [296] J. K. Pritchard, M. Stephens, and P. Donnelly, "Inference of population structure using multilocus genotype data," *Genetics*, vol. 155, no. 2, pp. 945–959, 2000.
- [297] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
- [298] C. Xiao, P. Zhang, W. A. Chaowalitwongse, J. Hu, and F. Wang, "Adverse drug reaction prediction with symbolic latent Dirichlet allocation," in *Proc. AAAI*, 2017, pp. 1590–1596.
- [299] D. R. Cox, "Regression models and life-tables," in *Breakthroughs in Statistics*. New York, NY, USA: Springer, 1992, pp. 527–541.
- [300] C. Lusivika-Nzinga, H. Selinger-Leneman, S. Grabar, D. Costagliola, and F. Carrat, "Performance of the marginal structural Cox model for estimating individual and joined effects of treatments given in combination," *BMC Med. Res. Methodol.*, vol. 17, no. 1, p. 160, 2017.
- [301] T. Käkilehto, S. Salo, and M. Larmas, "Data mining of clinical oral health documents for analysis of the longevity of different restorative materials in Finland," *Int. J. Med. Inform.*, vol. 78, no. 12, pp. e68–e74, 2009.
- [302] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2009, pp. 485–585.
- [303] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [304] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proc. ACM 23rd Int. Conf. Mach. Learn.*, 2006, pp. 161–168.
- [305] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.
- [306] B. Schölkopf and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [307] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, 2009.
- [308] B. Longstaff, S. Reddy, and D. Estrin, "Improving activity classification for health applications on mobile devices using active and semi-supervised learning," in *Proc. IEEE 4th Int. Conf. Pervasive Comput. Technol. Healthcare (PervasiveHealth)*, Mar. 2010, pp. 1–7.
- [309] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proc. ACM 23rd Int. Conf. Mach. Learn.*, 2006, pp. 417–424.
- [310] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.
- [311] R. S. Sutton and A. G. Barto, *Reinforcement Learning—An Introduction*, vol. 1. Cambridge, MA, USA: MIT Press, 1998.
- [312] D. Ernst, G.-B. Stan, J. Goncalves, and L. Wehenkel, "Clinical data based optimal STI strategies for HIV: A reinforcement learning approach," in *Proc. 45th IEEE Conf. Decis. Control*, Dec. 2006, pp. 667–672.
- [313] A. Trotman, "Learning to rank," *Inf. Retr.*, vol. 8, no. 3, pp. 359–381, 2005.
- [314] T.-Y. Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retr.*, vol. 3, no. 3, pp. 225–331, 2009.
- [315] F. Pedregosa, E. Cauvet, G. Varoquaux, C. Pallier, B. Thirion, and A. Gramfort, "Learning to rank from medical imaging data," in *Proc. Int. Workshop Mach. Learn. Med. Imag.* Berlin, Germany: Springer, 2012, pp. 234–241.
- [316] G. Bakir, *Predicting Structured Data*. Cambridge, MA, USA: MIT Press, 2007.
- [317] Y. Xu, K. Hong, J. Tsujii, and E. C. I. Chang, "Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries," *J. Amer. Med. Inform. Assoc.*, vol. 19, no. 5, pp. 824–832, 2012.
- [318] L. Massotier and S. Casciaro, "A new fully automatic and robust algorithm for fast segmentation of liver tissue and tumors from CT scans," *Eur. Radiol.*, vol. 18, no. 8, pp. 1658–1665, 2008.
- [319] A. Aggarwal and J. Vitter, "The input/output complexity of sorting and related problems," *Commun. ACM*, vol. 31, no. 9, pp. 1116–1127, 1988.
- [320] A. C.-C. Yao, "Some complexity questions related to distributive computing (Preliminary Report)," in *Proc. 11th STOC*, 1979, pp. 209–213.
- [321] T. Soyata, *GPU Parallel Program Development Using CUDA*. New York, NY, USA: Taylor & Francis, 2018.
- [322] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervent.* Berlin, Germany: Springer, 2013, pp. 411–418.
- [323] J. Zhou, F. Wang, J. Hu, and J. Ye, "From micro to macro: Data driven phenotyping by densification of longitudinal electronic medical records," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 135–144.
- [324] P. J. Schwartz *et al.*, "Prolongation of the QT interval and the sudden infant death syndrome," *New England J. Med.*, vol. 338, no. 24, pp. 1709–1714, 1998.
- [325] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi, "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance," *Neural Netw.*, vol. 21, nos. 2–3, pp. 427–436, 2008.
- [326] A. T. Özdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors*, vol. 14, no. 6, pp. 10691–10708, 2014.
- [327] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. ACM 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.
- [328] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [329] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.
- [330] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.
- [331] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, D, J. Basic Eng.*, vol. 82, pp. 35–45, 1960.
- [332] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng.*, vol. 83, no. 1, pp. 95–108, 1961.
- [333] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Sep. 2006.
- [334] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Partical Filters for Tracking Applications, vol. 685. Boston, MA, USA: Artech House, 2004.
- [335] M. S. Grewal and A. P. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]," *IEEE Control Syst.*, vol. 30, no. 3, pp. 69–78, Jun. 2010.
- [336] S. Kuindersma *et al.*, "Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot," *Auton. Robots*, vol. 40, no. 3, pp. 429–455, 2016.
- [337] T. S. Davis *et al.*, "Restoring motor control and sensory feedback in people with upper extremity amputations using arrays of 96 microelectrodes implanted in the median and ulnar nerves," *J. Neural Eng.*, vol. 13, no. 3, p. 036001, 2016.
- [338] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
- [339] L. H. Messer *et al.*, "In-home nighttime predictive low glucose suspend experience in children and adults with type 1 diabetes," *Pediatric Diabetes*, vol. 18, no. 5, pp. 332–339, 2017.
- [340] X. Chen, R. D. Shachter, A. W. Kurian, and D. L. Rubin, "Dynamic strategy for personalized medicine: An application to metastatic breast cancer," *J. Biomed. Inform.*, vol. 68, pp. 50–57, Apr. 2017.
- [341] E. B. Hunt, J. Marin, and P. J. Stone, *Experiments in Induction*. New York, NY, USA: Academic, 1966.
- [342] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
- [343] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Amsterdam, The Netherlands: Elsevier, 2014.
- [344] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [345] J. Ramon *et al.*, "Mining data from intensive care patients," *Adv. Eng. Inform.*, vol. 21, no. 3, pp. 243–256, 2007.
- [346] C. Vens, A. Van Assche, H. Blockeel, and S. Džeroski, "First order random forests with complex aggregates," in *Proc. Int. Conf. Inductive Logic Program.* Berlin, Germany: Springer, 2004, pp. 323–340.
- [347] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.
- [348] R. Harpaz, H. S. Chase, and C. Friedman, "Mining multi-item drug adverse effect associations in spontaneous reporting systems," *BMC Bioinf.*, vol. 11, no. 9, p. S7, 2010.

- [349] D. Sanchez-Morillo, M. A. Fernandez-Granero, and A. L. Jiménez, "Detecting COPD exacerbations early using daily telemonitoring of symptoms and k-means clustering: A pilot study," *Med. Biol. Eng. Comput.*, vol. 53, no. 5, pp. 441–451, 2015.
- [350] R. E. Bellman, *Dynamic Programming*. Mineola, NY, USA: Courier Corporation, 2013.
- [351] R. E. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.
- [352] R. A. Howard, *Dynamic Programming and Markov Processes*. Cambridge, MA, USA: MIT Press, 1960.
- [353] E. J. Sondik, "The optimal control of partially observable Markov decision processes," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 1971.
- [354] D. Aberdeen, "A (revised) survey of approximate methods for solving partially observable Markov decision processes," Nat. ICT Australia, Sydney, NSW, Australia, Tech. Rep., 2003.
- [355] J. Pineau, G. Gordon, and S. Thrun, "Anytime point-based approximations for large POMDPs," *J. Artif. Intell. Res.*, vol. 27, pp. 335–380, 2006.
- [356] P. Poupart and C. Boutilier, "VDCBPI: An approximate scalable algorithm for large POMDPs," in *Proc. NIPS*, 2004, pp. 1081–1088.
- [357] J. C. Gittins, "Bandit processes and dynamic allocation indices," *J. Roy. Statist. Soc. B, Methodol.*, vol. 41, no. 2, pp. 148–177, 1979.
- [358] D. W. Hosmer, Jr., and S. Lemeshow, *Applied Logistic Regression*. Hoboken, NJ, USA: Wiley, 2004.
- [359] K. Fukunaga and L. Hostetler, "Optimization of k nearest neighbor density estimates," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 3, pp. 320–326, May 1973.
- [360] T. Lan, A. Adami, D. Erdogmus, and M. Pavel, "Estimating cognitive state using EEG signals," in *Proc. IEEE 13th Eur. Signal Process. Conf.*, Sep. 2005, pp. 1–4.
- [361] G. Casella and R. L. Berger, *Statistical Inference*, vol. 2. Boston, MA, USA: Cengage Learning, 2002.
- [362] G. Bohm and G. Zech, *Introduction to Statistics and Data Analysis for Physicists*. Hamburg, Germany: Deutsches Elektronen-Synchrotron, 2010.
- [363] M. H. DeGroot, *Optimal Statistical Decisions*, vol. 82. Hoboken, NJ, USA: Wiley, 2005.
- [364] W. Dai, T. S. Brisimi, W. G. Adams, T. Mela, V. Saligrama, and I. C. Paschalidis, "Prediction of hospitalization due to heart diseases by supervised learning methods," *Int. J. Med. Inform.*, vol. 84, no. 3, pp. 189–197, 2015.
- [365] R. P. Moreno et al., "SAPS 3—From evaluation of the patient to evaluation of the intensive care unit. Part 2: Development of a prognostic model for hospital mortality at ICU admission," *Intensive Care Med.*, vol. 31, no. 10, pp. 1345–1355, 2005.
- [366] J. Hippisley-Cox, C. Coupland, Y. Vinogradova, J. Robson, and P. Brindle, "The performance of the QRISK cardiovascular risk prediction algorithm in an external UK sample of patients from general practice: A validation study," *Heart*, vol. 94, no. 1, pp. 34–39, 2008.
- [367] J. L. Gilbert et al., "Development and validation of a Bayesian model for perioperative cardiac risk assessment in a cohort of 1,081 vascular surgical candidates," *J. Amer. College Cardiol.*, vol. 27, no. 4, pp. 779–786, 1996.
- [368] B. Ustun and C. Rudin, "Supersparse linear integer models for optimized medical scoring systems," *Mach. Learn.*, vol. 102, no. 3, pp. 349–391, 2016.
- [369] Y. Zhang, Y. Sun, P. Phillips, G. Liu, X. Zhou, and S. Wang, "A multilayer perceptron based smart pathological brain detection system by fractional Fourier entropy," *J. Med. Syst.*, vol. 40, no. 7, p. 173, 2016.
- [370] W. Huang, C. P. Bridge, J. A. Noble, and A. Zisserman, "Temporal HeartNet: Towards human-level automatic analysis of fetal cardiac screening video," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2017, pp. 341–349.
- [371] Z. Zhang, F. Ringeval, J. Han, J. Deng, E. Marchi, and B. Schuller, "Facing realism in spontaneous emotion recognition from speech: Feature enhancement by autoencoder with LSTM neural networks," in *Proc. 17th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2016, pp. 3593–3597.
- [372] S. Wang et al., "Hearing loss detection in medical multimedia data by discrete wavelet packet entropy and single-hidden layer neural network trained by adaptive learning-rate back propagation," in *Proc. Int. Symp. Neural Netw.*, 2017, pp. 541–549.
- [373] S. B. Hu, D. J. L. Wong, A. Correa, N. Li, and J. C. Deng, "Prediction of clinical deterioration in hospitalized adult patients with hematologic malignancies using a neural network model," *PLoS ONE*, vol. 11, no. 8, p. e0161401, 2016.
- [374] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Roy. Signals Radar Establishment, Malvern, U.K., Tech. Rep. RSRE-MEMO-4148, 1988.
- [375] D. Wu et al., "Prediction of Parkinson's disease tremor onset using a radial basis function neural network based on particle swarm optimization," *Int. J. Neural Syst.*, vol. 20, no. 2, pp. 109–116, 2010.
- [376] M. G. Ruano, E. Hajimani, and A. E. Ruano, "A radial basis function classifier for the automatic diagnosis of cerebral vascular accidents," in *Proc. IEEE Global Med. Eng. Phys. Exchanges/Pan Amer. Health Care Exchanges (GMEPE/PAHCE)*, Apr. 2016, pp. 1–4.
- [377] D. F. Specht, "Probabilistic neural networks," *Neural Netw.*, vol. 3, no. 1, pp. 109–118, 1990.
- [378] J.-S. Wang, W.-C. Chiang, Y.-L. Hsu, and Y.-T. C. Yang, "ECG arrhythmia classification using a probabilistic neural network with a feature reduction method," *Neurocomputing*, vol. 116, pp. 38–45, Sep. 2013.
- [379] T. J. Hirschauer, H. Adeli, and J. A. Buford, "Computer-aided diagnosis of Parkinson's disease using enhanced probabilistic neural network," *J. Med. Syst.*, vol. 39, no. 11, p. 179, 2015.
- [380] M. Ahmadlou and H. Adeli, "Enhanced probabilistic neural network with local decision circles: A robust classifier," *Integr. Comput.-Aided Eng.*, vol. 17, no. 3, pp. 197–210, 2010.
- [381] Y. Kim, T. Soyata, and R. F. Behnagh, "Towards emotionally aware AI smart classroom: Current issues and directions for engineering and education," *IEEE Access*, vol. 6, pp. 5308–5331, Jan. 2018.
- [382] A. Mehrabian, *Nonverbal Communication*. Piscataway, NJ, USA: Transaction Publishers, 1972.
- [383] A. Mehrabian, *Silent Messages*, vol. 8. Belmont, CA, USA: Wadsworth, 1971.
- [384] S. Blackmore, *Consciousness: An Introduction*. Abingdon, U.K.: Routledge, 2013.
- [385] D. Kahneman and P. Egan, *Thinking, Fast and Slow*. London, U.K.: Macmillan, 2011.
- [386] J. D. Young, C. Cai, and X. Lu, "Unsupervised deep learning reveals prognostically relevant subtypes of glioblastoma," *BMC Bioinf.*, vol. 18, no. 11, p. 381, 2017.
- [387] C. Hu, R. Ju, Y. Shen, P. Zhou, and Q. Li, "Clinical decision support for Alzheimer's disease based on deep learning and brain network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
- [388] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
- [389] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
- [390] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [391] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [392] M. Lin, Q. Chen, and S. Yan. (2013). "Network in network." [Online]. Available: <https://arxiv.org/abs/1312.4400>
- [393] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
- [394] M. Gao, Z. Xu, L. Lu, A. P. Harrison, R. M. Summers, and D. J. Mollura. (2017). "Holistic interstitial lung disease detection using deep convolutional neural networks: Multi-label learning and unordered pooling." [Online]. Available: <https://arxiv.org/abs/1701.05616>
- [395] R. Salvador et al., "HELICoID: Interdisciplinary and collaborative project for real-time brain cancer detection," in *Proc. Comput. Frontiers Conf.*, 2017, pp. 313–318.
- [396] S. Chaichulee et al., "Multi-task convolutional neural network for patient detection and skin segmentation in continuous non-contact vital sign monitoring," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 266–272.
- [397] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng. (2017). "Cardiologist-level arrhythmia detection with convolutional neural networks." [Online]. Available: <https://arxiv.org/abs/1707.01836>
- [398] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [399] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.

- [400] C.-T. Chu, C.-H. Chang, T.-J. Chang, and J.-X. Liao, "Elman neural network identify elders fall signal base on second-order train method," in *Proc. 6th Int. Symp. Next Gener. Electron. (ISNE)*, May 2017, pp. 1–4.
- [401] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [402] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware LSTM networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 65–74.
- [403] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [404] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao, "Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1903–1911.
- [405] T. Soyata, L. Copeland, and W. Heinzelman, "RF energy harvesting for embedded systems: A survey of tradeoffs and methodology," *IEEE Circuits Syst. Mag.*, vol. 16, no. 1, pp. 22–57, Feb. 2016.
- [406] M. Habibzadeh, Z. Qin, T. Soyata, and B. Kantarci, "Large scale distributed dedicated- and non-dedicated smart city sensing systems," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7649–7658, Dec. 2017.
- [407] M. Habibzadeh, A. Boggio-Dandry, Z. Qin, T. Soyata, B. Kantarci, and H. T. Mouftah, "Soft sensing in smart cities: Handling 3Vs using recommender systems, machine intelligence, and data analytics," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 78–86, Feb. 2018.
- [408] I. Bisio, F. Lavagetto, M. Marchese, and A. Sciarone, "Smartphone-centric ambient assisted living platform for patients suffering from comorbidities monitoring," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 34–41, Jan. 2015.
- [409] M. Pouryazdan, B. Kantarci, T. Soyata, L. Foschini, and H. Song, "Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowd-sensing," *IEEE Access*, vol. 5, pp. 1382–1397, Jan. 2017.
- [410] M. Pouryazdan, C. Fiandrino, B. Kantarci, T. Soyata, D. Kliazovich, and P. Bouvry, "Intelligent gaming for mobile crowd-sensing participants to acquire trustworthy big data in the Internet of Things," *IEEE Access*, vol. 5, pp. 22209–22223, Dec. 2017.
- [411] M. Pouryazdan, C. Fiandrino, B. Kantarci, D. Kliazovich, T. Soyata, and P. Bouvry, "Game-theoretic recruitment of sensing service providers for trustworthy cloud-centric Internet-of-Things (IoT) applications," in *Proc. Globecom Workshops (GC Wkshps)*, Washington, DC, USA, Dec. 2016, pp. 1–6.
- [412] M. J. Paul and M. Dredze, "You are what you tweet: Analyzing Twitter for public health," in *Proc. ICWSM*, vol. 20, Jul. 2011, pp. 265–272.
- [413] AHRQ. (2011). *Practice Facilitation Handbook*. [Online]. Available: <https://www.ahrq.gov/professionals/prevention-chronic-care/improve/system/pfhandbook/mod8appbmonicalatte.html>
- [414] G. Honan, A. Page, O. Kocabas, T. Soyata, and B. Kantarci, "Internet-of-everything oriented implementation of secure digital health (D-Health) systems," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Messina, Italy, Jun. 2016, pp. 718–725.
- [415] N. Powers *et al.*, "The cloudlet accelerator: Bringing mobile-cloud face recognition into real-time," in *Proc. Globecom Workshops (GC Wkshps)*, San Diego, CA, USA, Dec. 2015, pp. 1–7.
- [416] T. Soyata, *Enabling Real-Time Mobile Cloud Computing Through Emerging Technologies*. Hershey, PA, USA: IGI Global, Aug. 2015.
- [417] T. Soyata, H. Ba, W. Heinzelman, M. Kwon, and J. Shi, "Accelerating mobile cloud computing: A survey," in *Communication Infrastructures for Cloud Computing*, H. T. Mouftah and B. Kantarci, Eds. Hershey, PA, USA: IGI Global, Sep. 2013, ch. 8, pp. 175–197.
- [418] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-Vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. 17th IEEE Symp. Comput. Commun. (ISCC)*, Cappadocia, Turkey, Jul. 2012, pp. 59–66.
- [419] S. Ames, M. Venkatasubramaniam, A. Page, O. Kocabas, and T. Soyata, "Secure health monitoring in the cloud using homomorphic encryption: A branching-program formulation," in *Enabling Real-Time Mobile Cloud Computing Through Emerging Technologies*, T. Soyata, Ed. IGI Global, 2015, ch. 4, pp. 116–152.
- [420] B. K. Yi, N. D. Sidiropoulos, T. Johnson, H. V. Jagadish, C. Faloutsos, and A. Biliris, "Online data mining for co-evolving time sequences," in *Proc. 16th Int. Conf. Data Eng.*, Feb./Mar. 2000, pp. 13–22.
- [421] J. H. Chang and W. S. Lee, "Finding recent frequent itemsets adaptively over online data streams," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 487–492.
- [422] N. D. Phung, M. M. Gaber, and U. Rohm, "Resource-aware online data mining in wireless sensor networks," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Apr. 2007, pp. 139–146.
- [423] C. C. Aggarwal, *Data Mining: The Textbook*. Cham, Switzerland: Springer, 2001.
- [424] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2005.
- [425] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2011.
- [426] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, and Z. Zhang. (2016). "Hardware for machine learning: Challenges and opportunities." [Online]. Available: <https://arxiv.org/abs/1612.07625>
- [427] K. H. Lee and N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *IEEE J. Solid-State Circuits*, vol. 48, no. 7, pp. 1625–1637, Jul. 2013.
- [428] O. Kocabas, R. Gyampoh-Vidogah, and T. Soyata, "Operational cost of running real-time mobile cloud applications," in *Enabling Real-Time Mobile Cloud Computing through Emerging Technologies*, T. Soyata, Ed. Hershey, PA, USA: IGI Global, 2015, ch. 10, pp. 294–321.
- [429] A. H. Shoeb and J. V. Gutttag, "Application of machine learning to epileptic seizure detection," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 975–982.
- [430] A. Biason, U. Mitra, and M. Zorzi, "Improved active sensing performance in wireless sensor networks via channel state information," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 2469–2473.
- [431] N. Verma, A. Shoeb, J. V. Gutttag, and A. P. Chandrakasan, "A micro-power EEG acquisition SoC with integrated seizure detection processor for continuous patient monitoring," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2009, pp. 62–63.
- [432] A. Page, O. Kocabas, T. Soyata, M. K. Aktas, and J. Couderc, "Cloud-based privacy-preserving remote ECG monitoring and surveillance," *Ann. Noninvasive Electrocardiol.*, vol. 20, no. 4, pp. 328–337, 2014.
- [433] V. Miori and D. Russo, "Anticipating health hazards through an ontology-based, IoT domestic environment," in *Proc. 6th Int. IEEE Conf. Innov. Mobile Internet Services Ubiquitous Comput. (IMIS)*, Jul. 2012, pp. 745–750.
- [434] J. M. Jerez *et al.*, "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," *Artif. Intell. Med.*, vol. 50, no. 2, pp. 105–115, 2010.
- [435] O. Kocabas and T. Soyata, "Utilizing homomorphic encryption to implement secure and private medical cloud computing," in *Proc. IEEE 8th Int. Conf. Cloud Comput. (CLOUD)*, New York, NY, USA, Jun. 2015, pp. 540–547.
- [436] R. H. Dolin *et al.*, "HL7 clinical document architecture, release 2," *J. Amer. Med. Inform. Assoc.*, vol. 13, no. 1, pp. 30–39, 2006.
- [437] *Digital Imaging and Communications in Medicine (DICOM)*, NEMA, Rosslyn, Virginia, 2009.
- [438] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "BCI2000: A general-purpose brain-computer interface (BCI) system," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1034–1043, Jun. 2004.
- [439] A. Schlögl. (2006). "GDF—A general dataformat for biosignals." [Online]. Available: <https://arxiv.org/abs/cs/0608052>
- [440] P. Sajda, A. Gerson, K. R. Muller, B. Blankertz, and L. Parra, "A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 2, pp. 184–185, Jun. 2003.
- [441] B. Blankertz *et al.*, "The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1044–1051, Jun. 2004.
- [442] B. Blankertz *et al.*, "The BCI competition III: Validating alternative approaches to actual BCI problems," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 153–159, Jun. 2006.
- [443] C. Wu *et al.*, "BioGPS: An extensible and customizable portal for querying and organizing gene annotation resources," *Genome Biol.*, vol. 10, no. 11, p. R130, 2009.
- [444] K. Bowyer *et al.*, "The digital database for screening mammography," in *Proc. 3rd Int. Workshop Digit. Mammography*, vol. 58, 1996, p. 27.
- [445] M. Heath *et al.*, "Current status of the digital database for screening mammography," in *Digital Mammography*. Dordrecht, The Netherlands: Springer, 1998, pp. 457–460.

- [446] C. Busso et al., "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang. Resour. Eval.*, vol. 42, no. 4, pp. 335–359, 2008.
- [447] J. Shiraishi et al., "Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists' detection of pulmonary nodules," *Amer. J. Roentgenol.*, vol. 174, no. 1, pp. 71–74, 2000.
- [448] B. van Ginneken, M. B. Stegmann, and M. Loog, "Segmentation of anatomical structures in chest radiographs using supervised methods: A comparative study on a public database," *Med. Image Anal.*, vol. 10, pp. 19–40, Feb. 2007.
- [449] B. M. Bot et al., "The mPower study, Parkinson disease mobile data collected using ResearchKit," *Sci. Data*, vol. 3, Mar. 2016, Art. no. 160011.
- [450] D. Hall, M. F. Huerta, M. J. McAuliffe, and G. K. Farber, "Sharing heterogeneous data: the national database for autism research," *Neuroinformatics*, vol. 10, no. 4, pp. 331–339, 2012.
- [451] K. J. Gorgolewski et al., "NeuroVault.org: A Web-based repository for collecting and sharing unthresholded statistical maps of the human brain," *Frontiers Neuroinform.*, vol. 9, p. 8, Apr. 2015.
- [452] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults," *J. Cogn. Neurosci.*, vol. 19, no. 9, pp. 1498–1507, 2007.
- [453] R. A. Poldrack et al., "Toward open sharing of task-based fMRI data: the OpenfMRI project," *Frontiers Neuroinform.*, vol. 7, p. 12, Jul. 2013.
- [454] D. Demner-Fushman, S. Antani, M. Simpson, and G. R. Thoma, "Design and development of a multimodal biomedical information retrieval system," *J. Comput. Sci. Eng.*, vol. 6, no. 2, pp. 168–177, 2012.
- [455] A. J. Williams et al., "Open PHACTS: Semantic interoperability for drug discovery," *Drug Discovery Today*, vol. 17, nos. 21–22, pp. 1188–1198, 2012.
- [456] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. S. Marcal, and J. Rozeira, "PH²—A dermoscopic image database for research and benchmarking," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 5437–5440.
- [457] A. L. Goldberger et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000. [Online]. Available: <http://circ.ahajournals.org/cgi/content/full/101/23/e215>, doi: 10.1161/01.CIR.101.23.e215.
- [458] A. E. W. Johnson et al., "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, p. 160035, May 2016.
- [459] T. Penzel, G. B. Moody, R. G. Mark, A. L. Goldberger, and J. H. Peter, "The apnea-ECG database," in *Proc. Comput. Cardiol.*, Sep. 2000, pp. 255–258.
- [460] K. Marek et al., "The parkinson progression marker initiative (PPMI)," *Prog. Neurobiol.*, vol. 95, no. 4, pp. 629–635, 2011.
- [461] K. Clark et al., "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *J. Digit. Imag.*, vol. 26, no. 6, pp. 1045–1057, 2013.
- [462] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, "Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 4, pp. 884–893, Apr. 2010.
- [463] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy," *Knowl.-Based Syst.*, vol. 60, pp. 20–27, Apr. 2014.
- [464] E. Decencière et al., "Feedback on a publicly distributed image database: The Messidor database," *Image Anal. Stereol.*, vol. 33, no. 3, pp. 231–234, 2014.
- [465] M. Elter, R. Schulz-Wendtland, and T. Wittenberg, "The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process," *Med. Phys.*, vol. 34, no. 11, pp. 4164–4172, 2007.
- [466] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," *Proc. Nat. Acad. Sci. USA*, vol. 87, no. 23, pp. 9193–9196, 1990.
- [467] M. Ziba, M. Lubicz, J. Witek, and J. M. Tomczak, "Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients," *Appl. Soft. Comput.*, vol. 14, pp. 99–108, Jan. 2014.
- [468] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano, "Heart disease data set," in *Proc. UCI KDD Arch.*, 1988.
- [469] Y. Jia et al. (2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: <https://arxiv.org/abs/1408.5093>
- [470] E. Schubert, A. Koos, T. Emrich, A. Züfle, K. A. Schmid, and A. Zimek, "A framework for clustering uncertain data," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 1976–1979, 2015. [Online]. Available: <http://www.vldb.org/pvldb/vol8/p1976-schubert.pdf>
- [471] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011.
- [472] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
- [473] T. Schaul et al., "PyBrain," *J. Mach. Learn. Res.*, vol. 11, pp. 743–746, Feb. 2010.
- [474] J. Demšar et al., "Orange: Data mining toolbox in Python," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2349–2353, 2013.
- [475] M. Hanke, Y. O. Halchenko, P. B. Sederberg, S. J. Hanson, J. V. Haxby, and S. Pollmann, "PyMVPA: A Python toolbox for multivariate pattern analysis of fMRI data," *Neuroinformatics*, vol. 7, no. 1, pp. 37–53, 2009.
- [476] The Theano Development Team et al. (2016). "Theano: A Python framework for fast computation of mathematical expressions." [Online]. Available: <https://arxiv.org/abs/1605.02688>
- [477] S. Sonnenburg et al., "The SHOGUN machine learning toolbox," *J. Mach. Learn. Res.*, vol. 11, pp. 1799–1802, Jun. 2010.
- [478] T. Joachims, "SVM: Support vector machines," Tech. Univ. Dortmund, Dortmund, Germany, Tech. Rep., 1999. [Online]. Available: <http://svmlight.joachims.org/>
- [479] T. Joachims, "A support vector method for multivariate performance measures," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 377–384.
- [480] M. Abadi et al. (2016). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: <https://arxiv.org/abs/1603.04467>
- [481] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.



OMID RAJABI SHISHVAN (S'17) received the B.Sc. degree in EE from the Sharif University of Technology in 2012 and the M.Sc. degree in ECE from the University of Rochester, Rochester, NY, USA, in 2015. He is currently pursuing the Ph.D. degree with the ECE Department, SUNY Albany, Albany, NY, USA, under the supervision of Dr. T. Soyata in the field of digital health.



DAPHNEY-STAVROULA ZOIS (M'14) received the B.S. degree in computer engineering and informatics from the University of Patras, Patras, Greece, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, CA, USA. Previous appointments include the University of Illinois, Urbana-Champaign, IL, USA. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, SUNY Albany, Albany, NY, USA. Her research is in decision making in uncertain environments. She received the Viterbi Dean's and Myronis Graduate fellowships.



TOLGA SOYATA (M'08–SM'16) received the B.S. degree in ECE from Istanbul Technical University in 1988, the M.S. degree in ECE from Johns Hopkins University, Baltimore, MD, USA, in 1992, and the Ph.D. degree in ECE from the University of Rochester, Rochester, NY, USA, in 2000. He is currently an Associate Professor with the ECE Department, SUNY Albany, Albany, NY, USA. His teaching interests include CMOS VLSI ASIC design, and FPGA- and GPU-based parallel computation. His research interests include autonomous systems, cyber physical systems, and digital health. He is a Senior Member of ACM.