Materials Horizons

ROYAL SOCIETY OF CHEMISTRY

REVIEW

View Article Online
View Journal | View Issue

Cite this: *Mater. Horiz.*, 2025, **12**, 6587

Machine learning in biosignal analysis from wearable devices

Inhea Jeong,[ab] Won Gi Chung,[ab] Enji Kim,[ab] Wonjung Park,[ab] Hayoung Song,[ab] Jakyoung Lee,[ab] Myoungjae Oh,[ab] Eunmin Kim,[ab] Joonho Paek,[ab] Taekyeong Lee,[ab] Dayeon Kim,[ab] Seung Hyun An,[ab] Sumin Kim,[ab] Hyunjoo Cho[c] and Jang-Ung Park *[abdef]

The advancement of wearable bioelectronics has significantly improved real-time biosignal monitoring, enabling continuous health tracking and providing personalized medical insights. However, the sheer volume and complexity of biosignal data collected over extended periods, along with noise, missing values, and environmental artifacts, present significant challenges for accurate analysis. Machine learning (ML) plays a crucial role in biosignal analysis by improving processing capabilities, enhancing monitoring accuracy, and uncovering hidden patterns and relationships within datasets. Effective ML-driven biosignal analysis requires careful model selection, considering data preprocessing needs, feature extraction strategies, computational efficiency, and accuracy trade-offs. This review explores key ML algorithms for biosignal processing, providing guidelines on selecting appropriate models based on data characteristics, processing goals, computational efficiency, and accuracy requirements. We discuss data preprocessing techniques, ML models (clustering, regression, classification), and evaluation methods for assessing the accuracy and reliability of ML-driven analyses. Furthermore, we introduce ML applications in health monitoring, disease diagnosis, and prediction across neurological, cardiovascular, biochemical, and other biosignals. Finally, we discuss the integration of ML with wearable bioelectronics and its revolutionary impact on future healthcare systems.

Received 13th March 2025,
Accepted 21st May 2025

DOI: 10.1039/d5mh00451a

rsc.li/materials-horizons

Wider impact

Recent advancements in flexible and soft bioelectronics have enabled real-time and long-term health monitoring, leading to an unprecedented increase in biosignal data. This has created a growing need for efficient data processing and interpretation, positioning ML as a transformative technology in biosignal analysis. We explore how ML algorithms, such as clustering, regression, and classification, are applied to biosignals from wearable devices to enhance signal processing by improving accuracy, noise reduction, and pattern recognition. Additionally, we discuss the application of machine learning-based analysis of neural, cardiovascular, and biochemical signals in advancing health monitoring, disease diagnosis, and predictive analytics. This review provides practical guidance for selecting suitable ML algorithms based on data characteristics and processing objectives. It also discusses the broader impact of next-generation smart wearables and ML-enabled biomedical technologies on the future of healthcare.

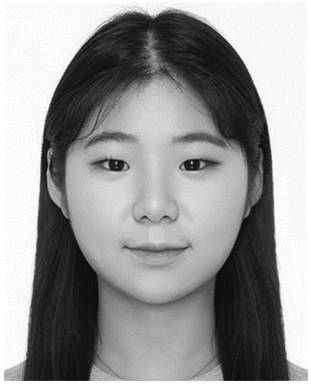
1. Introduction

The advancement of wearable bioelectronics has enabled the continuous, real-time monitoring of multiple biosignals and has significantly enhanced health monitoring capabilities.[1,2] These innovations allow for more precise tracking of physiological states, including disease progression, movement, and mental health status.[3,4] Initial biosignal monitoring relied on rigid electrodes, which were highly invasive and exhibited poor tissue compatibility, often triggering immune responses at the electrode-tissue interface. With the development of flexible and soft electrodes, minimally invasive wearable and implantable

[a] *Department of Materials Science and Engineering, Yonsei University, Seoul 03722, Republic of Korea. E-mail: jang-ung@yonsei.ac.kr*
[b] *Center for Nanomedicine, Institute for Basic Science (IBS), Yonsei University, Seoul 03722, Republic of Korea*
[c] *Department of Linguisitcs, Eberhard Karls Universität Tübingen, Keplerstraße 2, 72074 Tübingen, Germany*
[d] *Department of Neurosurgery, Yonsei University College of Medicine, Seoul 03722, Republic of Korea*
[e] *Graduate Program of Nano Biomedical Engineering (NanoBME), Advanced Science Institute, Yonsei University, Seoul 03722, Republic of Korea*
[f] *Yonsei-KIST Convergence Research Institute, Seoul 03722, Republic of Korea*

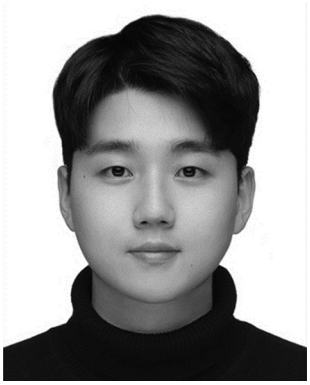
devices have emerged.[5,6] The integration of advanced structural designs, such as mesh and microfiber architectures, serpentine electrode patterns, and 2–3D deformable structures, has significantly enhanced the flexibility and stretchability of biosensors.[7–10] Additionally, soft materials including polymers, hydrogels, and liquid metals, have been introduced to improve biocompatibility and durability.[11,12] These materials exhibit mechanical properties similar to biological tissues, ensuring stable signal acquisition even under dynamic conditions.[13] These advancements minimize immune responses and allow biosensors to conform seamlessly to the body, supporting long-term, stable biosignal monitoring for both wearable and implantable applications.

Progress in bioelectronics has expanded the capabilities for monitoring electrical, physiological, and chemical biomarkers at various anatomical sites, including the brain, spinal cord, heart, blood vessels, and skin.[14,15] Beyond signal acquisition, the ability to measure diverse biosignals has not only improved comprehensive real-time health monitoring but also enabled



Inhea Jeong

Inhea Jeong received her BS degree in Physics at Konkuk University, South Korea. She is now on a PhD course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. Her research interests focus on soft electronic materials and devices for biomedical applications.

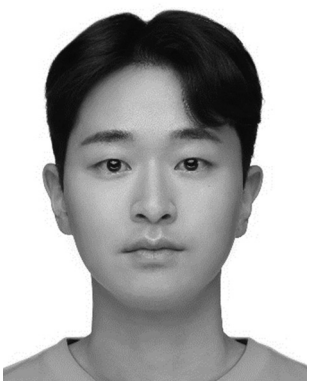


Won Gi Chung

Won Gi Chung received his PhD degree in materials science and engineering at Yonsei University, South Korea. He is now a postdoctoral fellow under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. His research interests focus on soft electronic materials and devices for biomedical applications.

Enji Kim

Enji Kim received her BS degree in materials science and engineering at Yonsei University, South Korea. She is now on a PhD course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. Her research interests focus on soft electronic materials and devices for biomedical applications.



Wonjung Park

Wonjung Park received his BS degree in optics and mechatronics engineering at Pusan National University, South Korea. He is now on a PhD course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. His research interests focus on soft electronic materials and devices for biomedical applications.

Hayoung Song

Hayoung Song received her BS degree in materials science and engineering at Kookmin University, South Korea. She is now on a PhD course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. Her research interests focus on wearable devices for biomedical applications.

Jakyoung Lee

Jakyoung Lee received her BS degree in materials science and engineering at Yonsei University, South Korea. She is now on a PhD course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. Her research interests focus on soft electronic materials and devices for biomedical applications.

the accumulation of personal bio-information, facilitating its use in personalized medicine and long-term health tracking.[16] Furthermore, large-scale biometric data collected over extended periods from diverse populations is systematically classified and statistically analyzed, enabling biomarker-based disease diagnosis, predictive analytics, and daily health management. As biosignal data grows in volume and complexity, efficient data processing and analytical tools have become essential.

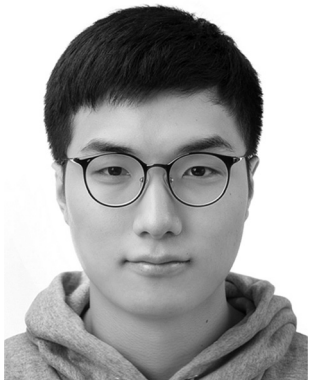
However, these vast and intricate datasets pose significant challenges for accurate analysis and meaningful interpretation. Wearable devices continuously produce large-scale, real-time data streams, but environmental factors such as motion artifacts and external interference can degrade signal quality.[17,18] These resulting noise and missing values complicate the extraction of reliable insights. Biosignals exhibit complex, nonlinear patterns, making simple statistical analyses insufficient for

Myoungjae Oh

Myoungjae Oh received his BS degree in nano science and engineering at Yonsei University, South Korea. He is now on an MS course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. His research interests focus on soft materials and devices for biomedical applications.

Eunmin Kim

Eunmin Kim received his BS degree in materials science and engineering at Yonsei University, South Korea. He is now on an MS course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. His research interests focus on soft electronic materials and devices for biomedical applications.



Joonho Paek

Joonho Paek received his bachelor's degree in nanoscience and engineering at Yonsei University, South Korea. He is now on an MS course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. His research interests focus on soft electronic materials and devices for wearable applications.

Taekyeong Lee

Taekyeong Lee received her BS degree in materials science and engineering at Yonsei University, South Korea. She is now on an MS course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. Her research interests focus on soft electronic materials and devices for electrophysiological applications.

Dayeon Kim

Dayeon Kim received her BS degree in materials science and engineering at Hallym University, South Korea. She is now on an MS course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. Her research interests focus on bioelectronics and wearable devices.

Seung Hyun An

Seung Hyun An received his BS degree in materials science and engineering at Yonsei University, South Korea. He is now on an MS course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. His research interests focus on soft electronic materials and devices for biomedical applications.

comprehensive interpretation. Therefore, optimized data processing and analytical methods are essential. Also, health status and disease progression cannot be assessed using a single biomarker but should instead be evaluated through the complex interactions among multiple biosignals.

Recently, machine learning (ML) has been increasingly integrated into biosignal analysis to address these challenges, enabling the extraction of meaningful insights. ML effectively processes large-scale biosignal data, improving monitoring accuracy and efficiency.[19] Additionally, ML facilitates the identification of hidden relationships within complex datasets, contributing to disease diagnosis, health status prediction, and decision-making. As ML applications in biosignal analysis expand, the diversity of biosignal types and characteristics necessitates the selection of appropriate ML models according to specific analytical objectives and data attributes.

As illustrated in Fig. 1a, this review examines key ML algorithms for biosignal processing. The selection and application of ML models are discussed in the context of the processing workflow, considering dataset characteristics and analytical objectives. This review also provides a guideline for choosing appropriate ML algorithms based on biosignal properties, processing goals, computational efficiency, and accuracy requirements. Also, we introduce ML applications in health monitoring, disease diagnosis, and prediction using neural, cardiovascular, biochemical, and other biosignals. Finally, we discuss the integration of ML with wearable bioelectronics, especially its impact on future healthcare systems, and potential directions for further development.

2. Machine learning algorithms for analyzing biosignals

We establish a signal processing workflow by configuring the processing steps according to the data characteristics of biosignals acquired from wearable devices and applying ML algorithms to train and evaluate each stage. Fig. 1b organizes this workflow and presents a reference guide that classifies representative ML algorithms based on their purpose and functionality within each step. Data preprocessing refers to preparing acquired data for ML algorithms by improving data quality and optimizing it for training. This process involves scaling, normalization, and dimensionality reduction, which enhance training speed, mitigate overfitting, improve accuracy, and increase computational efficiency—ultimately contributing to the overall performance of the ML model. After preprocessing, ML algorithms are employed to identify patterns and relationships within the dataset and perform prediction and classification tasks. ML algorithms are categorized according to their functional roles in biosignal analysis, namely clustering, regression, and classification. Following training and prediction, model evaluation is conducted to assess the validity of the results and ensure the reliability of predictions.

2.1 Data preprocessing algorithms

Biosignals obtained from wearable devices exist in various forms, ranging from simple low-dimensional data to high-dimensional data incorporating multiple factors. Ultimately, these signals must be processed and transformed into a format that is easily interpretable by humans. Before performing advanced processing, a preprocessing step is necessary to ensure that the data can be effectively utilized. This section

Sumin Kim

Sumin Kim received her BS degree in materials science and engineering at Hongik University, South Korea. She is now on an MS course under the supervision of Prof. Jang-Ung Park at the Department of Materials Science and Engineering in Yonsei University. Her research interests focus on soft electronic materials and devices for electrophysiological applications.

Hyunjoo Cho

Hyunjoo Cho is now on an BS course in the Department of Linguisitcs at Eberhard Karls Universität Tübingen. Her research interests focus on computational linguistics.

Jang-Ung Park

Jang-Ung Park achieved his PhD degree from the University of Illinois at Urbana-Champaign (UIUC) in 2009. After that, he served as a postdoctoral fellow at Harvard University from 2009 to 2010. He worked as an associate professor in School of Materials Science and Engineering at UNIST from 2010 to 2018. He is now a professor in the Department of Materials Science and Engineering at Yonsei University and the Department of Neurosurgery at Yonsei University College of Medicine. His current research is focused on wearable and biomedical electronics.

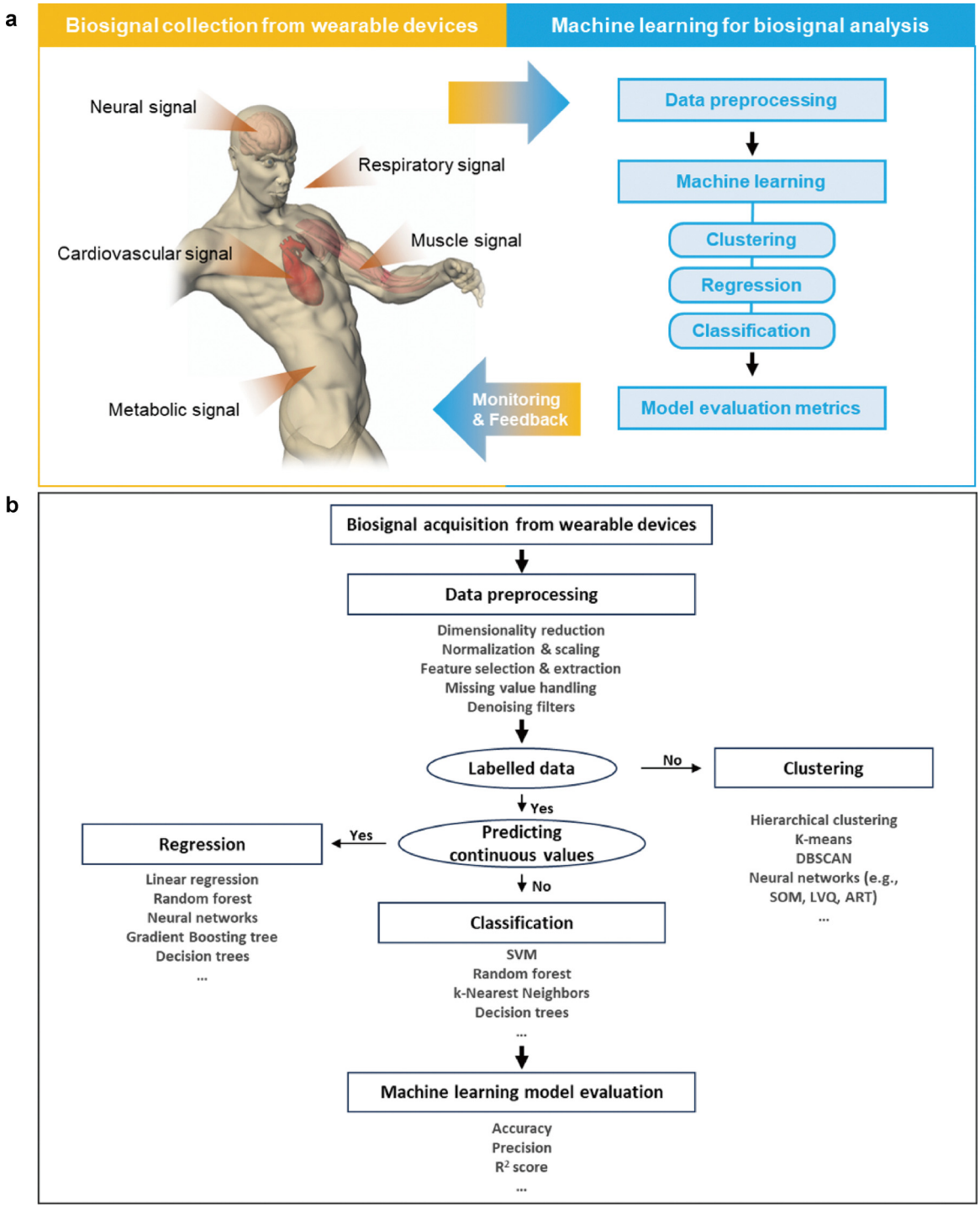


Fig. 1 (a) Overview of biosignal acquisition from wearable devices and ML-based analysis. (b) Overview of the biosignal processing workflow, integrated with a ML reference guide.

focuses on algorithms designed for preprocessing biosignals, thereby facilitating subsequent analysis (Fig. 2a). Given the diversity of biosignal properties, different preprocessing strategies are required depending on the specific challenges involved. To clarify this relationship, we provide a structured summary that connects typical biosignal characteristics to appropriate preprocessing techniques and their associated algorithms. Table 1 summarizes the recommended methods by highlighting their practical advantages in addressing common challenges in biosignal processing.

2.1.1 Dimensionality reduction. Dimensionality reduction is a crucial preprocessing technique in biosignal analysis, aimed at transforming high-dimensional data into a lower-dimensional representation while preserving essential information

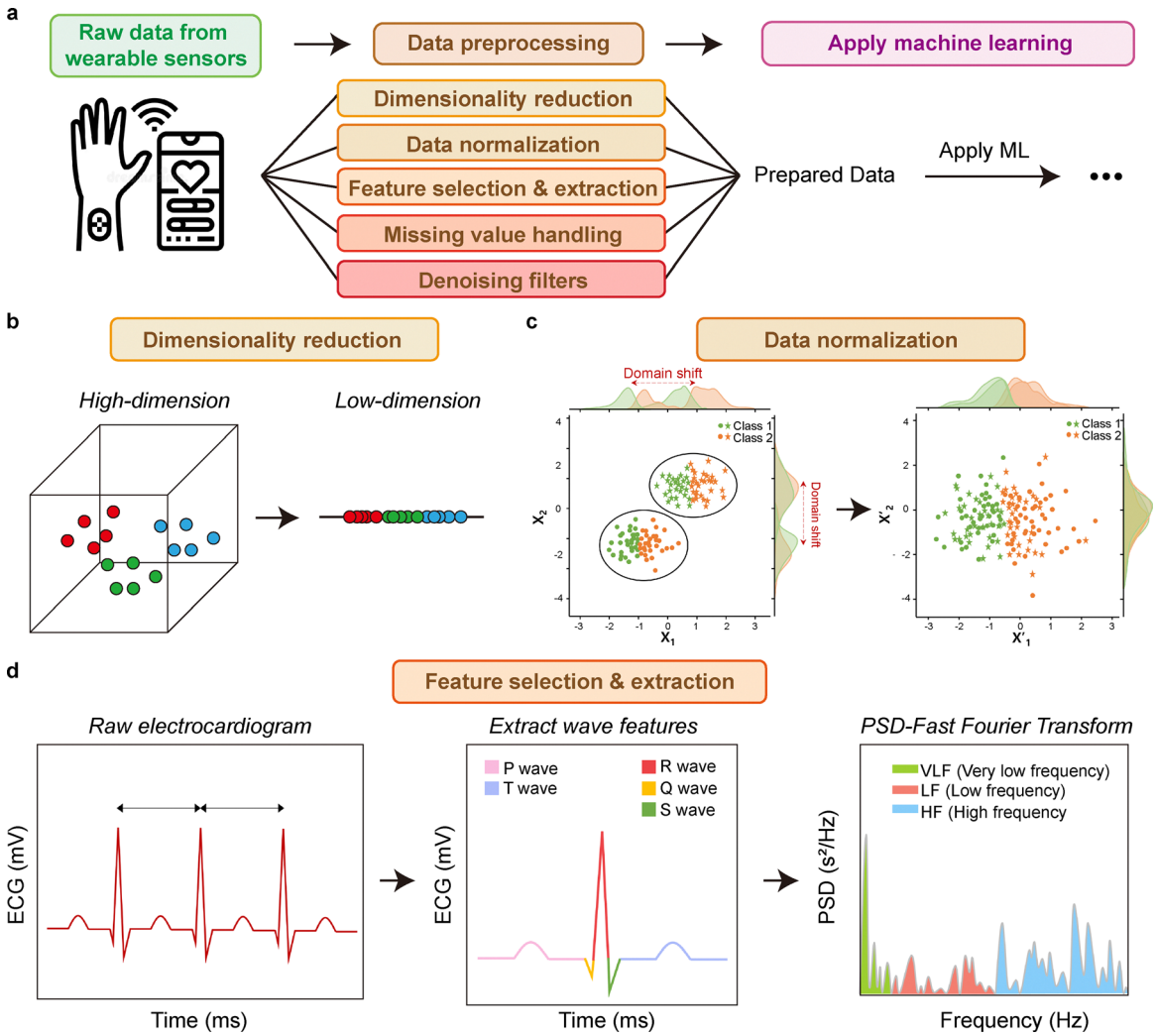


Fig. 2 Data preprocessing. (a) Pipeline showing representative preprocessing strategies to process raw data before applying ML. (b) Dimensionality reduction converting high-dimensional data into a lower-dimensional space. (c) Data normalization improving class separation. (d) Feature extraction from ECG signals includes wave detection and frequency analysis using fast Fourier transform (FFT).

Table 1 Summary of data preprocessing techniques for biosignal analysis

Preprocessing technique	Associated algorithms	Advantages for biosignal properties	Ref.
Dimension reduction	PCA, ICA, t-SNE, UMAP	Reducing noise, handling data efficiently, and enhancing visualization	20–26
Data normalization	Z-score, Min–Max	Domain shift alleviation, generalization in ML, reduction of computational cost and runtime, improvement of classification accuracy	27–31
Feature selection & extraction	STFT, wavelet, PCA, filter/wrapper/ embedded methods	Uncovering new feature, prevention of overfitting, enhancement of model performance and interpretability	32,33
Missing value handling	Mean/median imputation, KNN imputation, multiple imputation	Achieving unbiased results, ensuring data integrity, and improving the robustness of ML models	34,35
Denoising filters	High/low-pass filtering, cubic spline interpolation, EMD, CNN	Achieving discrete signals, enhancing reliability and performance of algorithms, facilitating data interpretation	36–39

(Fig. 2b).[40–42] Biosignals obtained from biological systems, such as electroencephalography (EEG) and electrocardiography (ECG), are inherently complex and often contain multiple factors, including time, amplitude, and phase. The high dimensionality of these signals poses challenges in terms of computational efficiency and interpretability. Directly processing raw biosignals without reducing dimensionality can lead to increased computational costs, longer processing times, and difficulties in extracting meaningful patterns. Therefore, dimensionality reduction techniques are employed to preprocess the raw data for efficient data handling, by reducing the noise of the signal and improving the performance of subsequent ML models.

Dimensionality reduction methods can be broadly categorized into linear and nonlinear techniques, depending on the nature of the data and the relationships among its features.[43] Linear methods assume that high-dimensional data lie in a lower-dimensional subspace and transform the original data accordingly to facilitate the ML process after. One of the most widely used linear techniques is principal component analysis (PCA), which identifies the directions, or principal components, that capture the maximum variance in the data and projects the data onto these directions.[20,21] By retaining only the most significant components, PCA effectively reduces dimensionality while preserving essential information. Another commonly used linear method is linear discriminant analysis (LDA), which is particularly useful in classification tasks as it maximizes the separation between different classes in the data.[22,23] Additionally, independent component analysis (ICA) is frequently employed in biosignal processing to separate overlapping sources, such as distinguishing different signal components in EEG and ECG recordings.[24]

In cases where data exhibit complex, nonlinear relationships, nonlinear dimensionality reduction techniques are more effective.[25] Methods such as uniform manifold approximation and projection (UMAP) and t-distributed stochastic neighbor embedding (t-SNE) are widely used for visualizing high-dimensional biosignals in a lower-dimensional space while preserving local structures.[26] These techniques are particularly useful for exploratory data analysis and clustering applications.

One key application of dimensionality reduction is noise reduction, where methods like PCA and ICA help remove unwanted noise while retaining meaningful signal components. Additionally, dimensionality reduction facilitates feature extraction, which will be introduced in the latter section, by identifying the most relevant features, which enhances the performance of ML models. In real-time biosignal processing applications, such as wearable sensor systems, reducing dimensionality allows for faster computation and more efficient data handling. Moreover, visualization techniques enabled by dimensionality reduction help researchers and clinicians interpret high-dimensional biosignal data more effectively.

Despite its advantages, dimensionality reduction presents several challenges. One major concern is the loss of information, as reducing dimensions inevitably discards some data, potentially affecting downstream analysis.[44] Selecting an optimal number of dimensions is crucial to maintaining a balance between data simplification and information retention. Additionally, the interpretability of dimensionality reduction results varies depending on the method used. While PCA provides clear principal components, methods like autoencoders generate abstract latent representations that may be difficult to interpret. Nonlinear techniques such as t-SNE and UMAP are primarily useful for visualization rather than preserving mathematical relationships within the data. Another consideration is computational complexity, as certain methods, such as t-SNE, require significant computational resources, which can be a limitation for real-time applications.

2.1.2 Normalization & scaling. Typical biosignals, such as EEGs, exhibit non-stationary characteristics, meaning that their statistical properties dynamically change over time.[45] Consequently, even when recorded from the same subject under similar conditions, the statistical distribution of EEG signals is not constant across different periods due to inherent neural variability and external influences (*e.g.*, user emotion and electrode placement).[46,47] Additionally, EEG signals exhibit significant inter-subject variability, as Saha *et al.* show that EEG patterns vary across individuals during motor imagery tasks.[48] This is referred to as domain shift and limits the ability of ML models, which were trained on a sample dataset (*i.e.*, source domain), to generalize effectively to signals from target subjects, thereby reducing the reliability of model predictions.[27]

To address this challenge, various domain adaptation techniques have been developed to mitigate variance across datasets, ranging from simple normalization methods to complex deep learning algorithms. Data normalization and scaling are fundamental pre-processing techniques that reduce variability across datasets, facilitating domain adaptation (Fig. 2c).[28] By ensuring consistency in input feature distributions, these methods help ML models learn more robust patterns. Specifically, transforming data scales ensure that all features contribute equivalently, minimizing the biases in model training. This process is crucial in alleviating the domain shift, a challenge that occurs when statistical discrepancies between training and test datasets hinder model generalization.[29] Addressing the domain shift is particularly important in biomedical signal analysis, where variations in data acquisition conditions can significantly impact model performance.[30]

Data normalization & scaling are essential not only for addressing domain shifts generated from a single source, but also for processing recorded data from various sources in wearable health monitoring devices.[49] For effective health monitoring, recording multiple biosignals simultaneously enables comprehensive health state evaluation. Therefore, techniques have been developed toward multimodal wearable devices to facilitate the comparison and analysis of data from various sources. However, depending on the type of signal source, data can vary significantly in unit and scale, requiring appropriate normalization for meaningful comparison. If data analysis with multiple biosignals is performed without normalization and scaling, computational cost and processing time increase, and features with larger units dominate those with smaller units, potentially distorting the learning process.[31] Therefore, data normalization and scaling can enhance classification performance across various applications, including medical data analysis, multimodal biometric systems, and industrial fault detection. The most widely used methods for data normalization and scaling comprise Z-score normalization and Min–Max normalization.[50]

Z-score normalization is a method for normalizing data by transforming them into a dataset with a zero mean and a unit variance. This method transforms each data point $x_{i,n}$ in the dataset into its normalized value $x'_{i,n}$ according to the following equation (eqn (1)):

$$x'_{i,n} = \frac{x_{i,n} - \mu_i}{\sigma_i} \quad (1)$$

Here, μ and σ denote the mean and standard deviation, respectively. It is effective in reducing the influence of outliers.[51] However, since the mean and standard deviation of the biosignals, whose values change over time, their effectiveness in maintaining consistent scaling across data with the same unit diminishes.[52] On the other hand, Min–Max normalization preserve the relative position of raw data, as it is applied uniformly to all data regardless of time. This method rescales data within a predetermined range [0,1] using the following equation (eqn (2)):

$$x'_{i,n} = \frac{x_{i,n} - \min(x_i)}{\max(x_i) - \min(x_i)}(n\mathrm{Max} - n\mathrm{Min}) + n\mathrm{Min} \quad (2)$$

However, Min–Max normalization has a limitation in that it is more sensitive to the presence of outliers compared to Z-score normalization.[53] Thus, the choice of normalization method should be determined based on the specific characteristics of biosignals, as different approaches may be more suitable depending on the nature of the data. Singh *et al.* conducted a study in which they applied 14 different normalization techniques to preprocess data, followed by ML training, and evaluated the classification accuracy against that of unnormalized data. Even though the 14 normalization methods differed from one another, with Z-score normalization obtaining the best rank in the full feature set approach, all methods improved classification accuracy and reduced runtime compared to unnormalized data.

2.1.3 Feature extraction & selection. As measurement techniques advance from single-lead to multi-lead systems, the size and diversity of acquired biosignals are increasing.[54] Consequently, extracting or selecting meaningful features from the vast amount of data has become crucial for effective signal analysis. To achieve this, two key methods are employed: (1) feature extraction and (2) feature selection. Feature extraction transforms raw data to uncover new features that are not explicitly present, often through dimensionality reduction. In contrast, feature selection merely chooses relevant features while preserving the original data structure, without transforming the raw data. Feature extraction and selection are separate processes, but they are often used together, by extracting distinct features from the data through feature extraction and applying feature selection to choose the most relevant features.[55] By applying these processes, the researchers can prevent overfitting, which occurs when the model is overly sensitive to irrelevant features, leading to classification errors.[32] This, in turn, enhances the performance of subsequent ML models and improves their interpretability.

Feature extraction is the process of reducing data dimensionality to extract new features, and it is important to consider in advance which domain the required features should belong to. Biosignals can be broadly categorized into four domains: (1) time domain, (2) frequency domain, (3) joint time-frequency domain, (4) signal decomposition and sparse domain.[33] To transform raw data into these domains, a wide range of methods can be applied, from relatively simple statistical approaches to more advanced supervised and unsupervised ML algorithms, all of which help extract meaningful features for data analysis. For example, statistical features such as mean, maximum/minimum values, skewness, and inter-beat-interval derived from the statistical approaches, belong to the time domain.[56,57] Using an algorithm, such as t-SNE to identify overlapping features can help mitigate overfitting.[58] As an alternative, the PCA method captures features with the highest variance, *i.e.*, it extracts principal components that are distinct and do not contain redundant information. Electrophysiological signals, such as ECG and EEG, are non-stationary signals; thus, both time and frequency components play a crucial role in their analysis. As a result, the time and frequency domains may be insufficient, depending on the purpose of signal analysis, to fully represent the features of electrophysiological data. In such cases, transforming data into the time-frequency domain using algorithms such as short-time Fourier transform (STFT), wavelet transform (WT), or Wigner–Ville distribution (WVD) is a suitable approach (Fig. 2d).[59] Transforming data into the signal decomposition and sparse domain using algorithms such as empirical mode decomposition (EMD) or dictionary learning is appropriate for identifying the underlying structure and patterns within a dataset. In addition, electrophysiological signals exhibit common wave patterns, and morphological features can be visually extracted as distinctive characteristics.[60] Moreover, when raw data are mixed with noise, reducing data dimensionality to select features that are not affected by noise is one of the key objectives of feature extraction.

In contrast to feature extraction, feature selection is a simpler complex process, as it does not transform the data type but merely selects relevant features. However, with the development of measurement techniques and feature extraction algorithms, the number of features has increased exponentially, necessitating more complex feature selection methods to handle the large number of irrelevant features.[61] Selecting appropriate features enables subsequent ML models to avoid being dominated by irrelevant features. Feature selection is categorized into three methods based on their correlation with the classifier, namely, the filter method, wrapper method, and embedded method.[62] The filter method selects features based on statistical measures, such as variance and correlation with the target variable, without using a classifier, whereas the wrapper method evaluates different feature combinations by iteratively training a classifier to determine the most relevant feature subset.[63,64] On the other hand, the embedded method incorporates feature selection into the model training process, where the classifier itself assesses the contribution of each feature and assigns corresponding weights.[65] While the filter method has the advantage of being computationally efficient, it does not account for interactions between features, which may lead to suboptimal performance.[66] In contrast, the wrapper and embedded methods evaluate feature relevance using a classifier, which increases computational complexity but often results in improved feature selection.[67]

2.1.4 Missing value handling. Biosignals collected from wearable devices and biomedical sensors are often prone to missing values due to various factors such as sensor malfunctions, signal interference, motion artifacts, and transmission

errors. Also, unlike experimental data that are collected per research protocol, the primary function of clinical data is to help clinicians care for patients, so the procedures for its collection are not often systematic.[68] Failure to address missing values appropriately can lead to biased results, reduced model performance, and misinterpretation of physiological conditions. Therefore, implementing effective missing value handling techniques is essential for ensuring data integrity and improving the robustness of ML models.[34,35] There are several approaches to handling missing values in biosignal processing, which can be broadly categorized into deletion-based methods and imputation techniques.

Deletion-based methods are the simplest approach, where records or features containing missing values are removed from the dataset.[69] This method includes listwise deletion, where entire samples with missing values are excluded, and pairwise deletion, where missing values are ignored only in calculations that do not require them.[70] While deletion methods are straightforward and preserve the integrity of complete raw data points, they can lead to information loss, especially if a significant proportion of data is missing. This approach is generally suitable when missing values are minimal and randomly distributed.

Imputation techniques aim to fill in missing values with estimated values based on existing data.[71] One of the most common imputation methods is mean imputation, where missing values are replaced with the mean of the observed data for that feature. Similarly, median and mode imputation can be used for non-normally distributed or categorical data, respectively. However, simple imputation methods may introduce bias and fail to capture complex dependencies within the data.[72] More advanced techniques, such as regression imputation, utilize statistical models to predict missing values based on relationships between variables. For example, multiple imputations generate several plausible values for each missing data point by incorporating random variability, enhancing the robustness of the imputation process. Another powerful imputation method is *k*-nearest neighbors (KNN) imputation, where missing values are estimated based on the most similar observations in the dataset. This technique is particularly useful for biosignal data, as it preserves local patterns and correlations between variables. Similarly, ML-based imputation methods, such as random forests and deep learning models, can be employed to predict missing values using complex patterns in the data. These approaches often outperform traditional methods, especially for high-dimensional and nonlinear biosignals.

Handling missing values in biosignal data presents several challenges. One major concern is determining the underlying mechanism of missingness, which can be classified into three categories: (1) missing completely at random (MCAR), (2) missing at random (MAR), and (3) missing not at random (MNAR).[73,74] In MCAR, missing values occur randomly and do not depend on any observed or unobserved variables. In MAR, the probability of missing values depends on observed variables but not on unobserved ones. In MNAR, missingness is related to the unobserved data itself, making it the most challenging case to address. Understanding the missing data mechanism is crucial in selecting the most appropriate handling method. Another challenge is the impact of missing values on real-time biosignal processing. In wearable sensor applications, real-time data processing requires efficient handling of missing values to ensure continuous monitoring and decision-making. Traditional imputation methods may not be feasible in real-time scenarios, necessitating adaptive algorithms that dynamically estimate missing values based on streaming data. Furthermore, the presence of missing values can introduce biases in ML models, making it essential to evaluate the effectiveness of the chosen handling method through rigorous validation techniques.

2.1.5 Denoising filters. During recording biosignals using wearable devices, the signals are often subjected to various interferences and noises in practical applicatons. Therefore, implementing effective denoising filters is crucial. Accurate denoising plays a vital role in improving signal fidelity and enabling precise assessment of physiological activity and abnormalities. In clinical settings, this is particularly important, as even low-level noise can mask diagnostically relevant features.[36] An effective denoising strategy contributes significantly to downstream processing by (1) increasing the precision of feature extraction, (2) enhancing the reliability and performance of automated diagnostic algorithms, and (3) facilitating clinical interpretation of the data. While a wide range of machine learning techniques are available for this purpose, selecting the most suitable model and tuning its parameters remains a complex task. This is because the optimal architecture and hyperparameters often depend on the specific type of noise present, requiring careful customization of neural network structures to match the characteristics of each dataset.

One of the most simple approach is to remove typical frequency range of signals using high or low-pass filters, or typical amplitude peaks. For example, in ECG signals, the most representative noise includes baseline wander (BW), which is a sinusoidal component at the frequency of respiration caused by breathing.[37] To address BW, which generally occurs between 0.15 and 0.3 Hz, conventional high-pass filters and cubic spline interpolation have been used, but they risk distorting key ECG features such as the ST segment. More advanced methods, such as wavelet-based denoising and empirical mode decomposition (EMD), provide multi-scale decomposition of the signal for more selective noise suppression.[38] Recently, deep learning approaches have demonstrated promising results, particularly when combined with wavelet transforms or morphological priors.[39] These models are capable of adaptively distinguishing signal from noise, even in the presence of overlapping frequency content or nonstationary characteristics, achieving high signal-to-noise ratio (SNR) improvements while preserving clinical interpretability.

Designing an effective denoising filter is challenging because different noise types often overlap with the signal both in time and frequency domains, making simple filtering insufficient. Additionally, overly aggressive filtering can distort important physiological features, leading to reduced diagnostic accuracy. Therefore, advanced denoising strategies, particularly those leveraging data-driven approaches such as deep learnin,

offer a promising solution by learning to selectively suppress noise while preserving critical signal morphology. As wearable and real-time monitoring systems continue to evolve, the development of robust, adaptive, and interpretable denoising frameworks will be essential for improving the reliability and clinical utility of biosignal analysis.

2.2 Machine learning algorithms for biosignal analysis

Following the preprocessing steps described above, ML algorithms are applied to identify patterns and relationships within the data, and to perform tasks such as prediction and classification. When handling with unlabeled data (unsupervised learning), clustering algorithms can be used to identify patterns, structures, and relationships within the dataset. When working with labeled data, where input data is paired with output labels, supervised learning algorithms can model the relationship between input and output variables and generate predictions for new inputs. Supervised learning can be divided into regression and classification, depending on the type of predictor variables. Regression models are designed to predict continuous values by capturing the quantitative relationships among variables, while classification models aim to predict categorical outcomes by assigning new input data to predefined classes, either binary or multiclass. In this section, ML algorithms are reviewed based on their primary analytical functions—clustering, regression, and classification. Notably, several ML models can serve multiple purposes depending on the context and configuration. For example, neural networks have demonstrated broad applicability across all three functional categories. Table 2 presents a functional overview of representative ML algorithms, categorized according to their roles in biosignal analysis.

2.2.1 Clustering. Clustering is an unsupervised learning technique used to group data points. Since it only relies on unlabeled data, clustering is particularly useful for identifying hidden patterns in large, complex datasets. This capability is particularly valuable in healthcare because it may lead to novel insights into our health system.[97,98] Biosignals acquired from wearable devices often exhibit high variability and individual specificity, making clustering an effective approach for uncovering underlying patterns. The integration of clustering and wearable devices typically aims to enhance the accuracy of data analysis such as prediction and classification. Clustering has already demonstrated its capability well in these tasks, proving

Table 2 Summary of machine learning algorithms for biosignal analysis

Type	Machine learning algorithm	Advantages for biosignal properties	Ref.
Clustering	Hierarchical clustering	No need for predefined number of clusters.	75
	K-means	Applicable to large datasets due to its simplicity and computational efficiency.	76
	Fuzzy *C*-means	Suitable for ambiguous data owing to its flexibility in allocating data points to multiple clusters.	77
	DBSCAN	Detects nonlinear cluster structures; robust to noise and outliers.	78
	Graph clustering	Captures network-like relationships in biosignals, useful for functional connectivity or interaction analysis.	79
	Model-based clustering (*e.g.*, GMM)	Enables probabilistic modeling of overlapping physiological states and automatic selection of cluster numbers.	80
	Neural networks (*e.g.*, SOM, LVQ, ART)	Learns complex, nonlinear signal structures and adapts dynamically to real-time biosignal changes.	81,82
Regression	Linear regression	High computational efficiency, fast execution, and straightforward interpretability.	83–85
	Ridge linear regression	Reduction of overfitting and improvement of predictive performance in the presence of multicollinearity.	86
	Regression tree	Effective modeling of nonlinear relationships and intuitive interpretability.	87
	Random forest model	Enhancement of prediction accuracy, reduction of overfitting, and robustness to diverse biosignal types.	88
	Artificial neural network	Suitability for moderately complex nonlinear pattern modeling and scalability to large biosignal datasets.	89
	Deep neural network	Capability for hierarchical feature extraction and representation of complex biosignal structures.	90
Classification	Logistic regression	Performs probabilistic classification using a linear decision boundary; suitable for binary and multi-class classification of biosignals. Offers fast training and high interpretability, enabling effective disease diagnosis and condition monitoring.	91
	SVM (support vector machine)	Effectively handles high-dimensional biosignal data and solves non-linear classification problems through kernel functions. Robust against overfitting with strong generalization capability.	92
	k-NN (*k*-nearest neighbors)	Instance-based method that classifies data based on proximity to neighbors; ideal for small-scale or spatially distributed biosignal datasets. Requires no prior training and adapts well to diverse data structures.	93
	Decision tree	Uses a hierarchical tree structure to split data based on feature conditions. Enables intuitive interpretation and fast prediction, handling both continuous and categorical biosignal features.	94
	Random forest	Ensemble model that aggregates multiple decision trees to enhance classification accuracy and stability. Captures complex relationships in biosignal data with improved robustness.	95
	Gradient boosting	Sequentially minimizes residual errors to model complex patterns in biosignals. Demonstrates high accuracy in real-world applications and is resilient to noise and incomplete data.	96

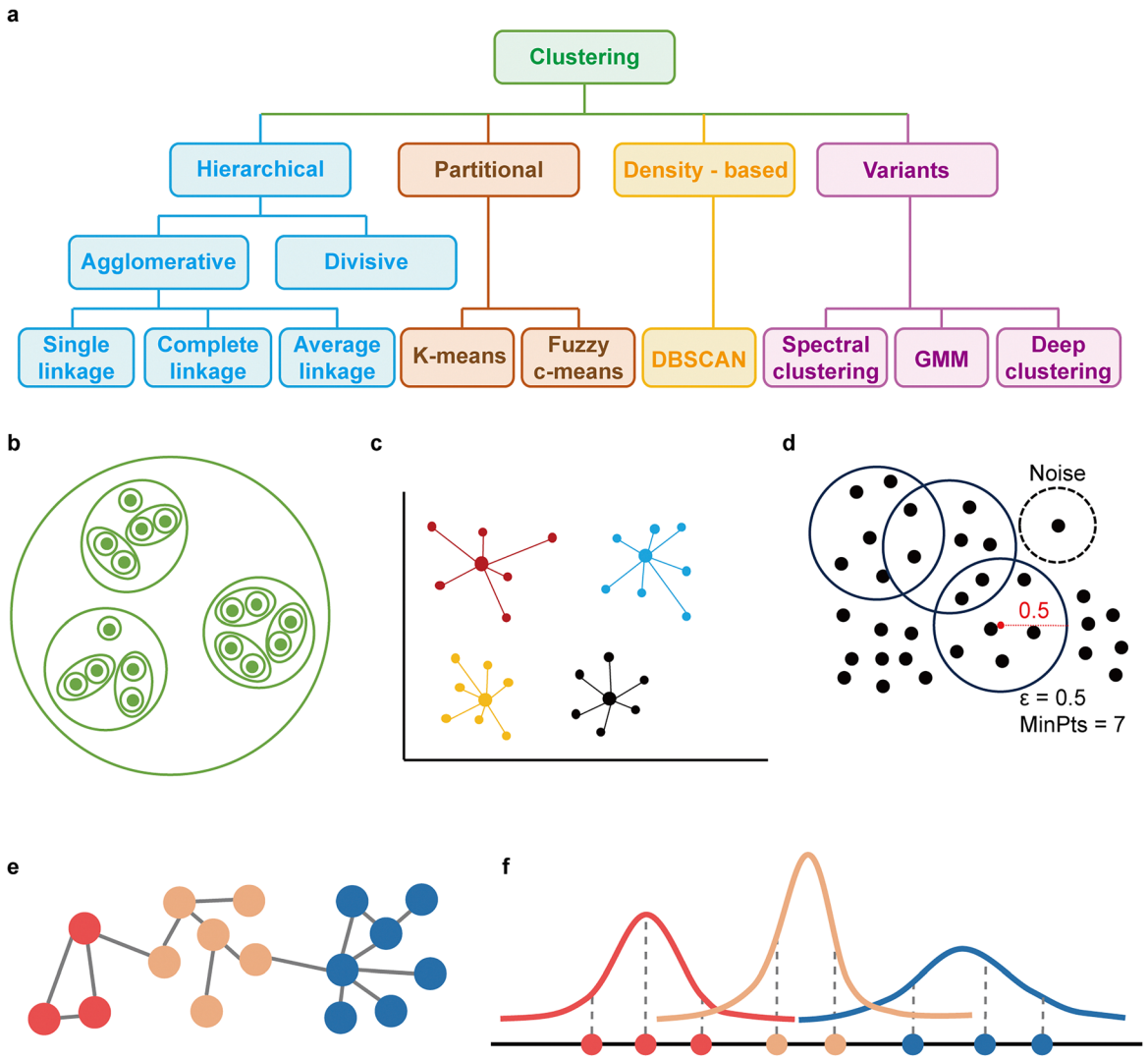


Fig. 3 Clustering. (a) Categorization of clustering. (b) Illustration of hierarchical clustering. (c) Illustration of *k*-means clustering. (d) Illustration of density-based spatial clustering of applications with noise (DBSCAN). E = 0.5, MinPts = 7. A noise point that does not satisfy to be a member of a cluster is shown with a dashed circle. (e) Illustration of graph clustering. (f) Illustration of model-based clustering using Gaussian mixture models (GMM).

to be a powerful tool for extracting meaningful insights into our health system.[99,100] Clustering techniques are broadly categorized according to their underlying mechanisms, with the primary distinction being between hierarchical, partitional, and density-based clustering (Fig. 3a). This section provides an overview of the basics of these types and their example models.

2.2.1.1 Hierarchical clustering. Hierarchical clustering is extensively used for analyzing large datasets. As its name suggests, it considers the data as a system of hierarchy and separates the data in progressive steps either in a bottom-up or a top-down manner to form a dendrogram.[75] Hierarchical clustering can be categorized into two primary approaches based on the direction of the clustering process: agglomerative (bottom-up) and divisive (top-down) methods. Both processes repetitively create clusters based on the distances of the components measured by distinct methods for each of them and yield clustered plots as shown in Fig. 3b.

A divisive process starts with a single larger cluster composed of all objects and progressively divides the objects into smaller clusters until every object has its own distinct cluster. Since it follows a top-down process that operates directly on data points, simple operations such as Euclidean distance or Manhattan distance are used for distance measurement. On the other hand, an agglomerative process treats each data point as a cluster and builds larger clusters that embrace the smaller clusters. This is repeated until a cluster that includes all objects is formed. Unlike a divisive process, the agglomerative process is a bottom-up process that starts with designating a cluster for every point. Therefore, some procedure for measuring the distance between clusters is required. Three methods may be used for this purpose: single-linkage clustering, complete-linkage clustering, and average-linkage clustering. In single-linkage clustering, the shortest distance between the clusters is determined. In complete-linkage clustering, the longest distance between any two points in different clusters is determined. In average-linkage clustering,

the average distance of all links between the clusters is determined. The determined distances guide the merging process of agglomerative clustering.

2.2.1.2 Partition-based clustering. Partition-based clustering divides data into a predefined number of clusters. This is achieved using criterion functions such as the Euclidean distance, which is equivalent to the shortest straight-line distance between two points.[76] A representative type of partition-based clustering algorithm is *k*-means clustering. In *k*-means clustering, the number of clusters k is initially selected, and a centroid, the center of the cluster, is determined for each cluster. Then, data points are rearranged to minimize their distances from the centroid, followed by resetting the centroids according to the new arrangements of the points (Fig. 3c). This iterative process continues until convergence, where the centroids no longer change significantly between iterations. *K*-means clustering has advantages in the simplicity of the algorithms and applicability to large datasets owing to its optimization capability to larger data, since both its number of repetitions and k range below 100 and parallel computation is possible. However, this technique lacks a standardized method for initial partition and determination of k, while it is also sensitive to outliers and noise.

Another method of partition-based clustering is fuzzy *c*-means clustering, which differs from *k*-means clustering in that each data point can belong to multiple clusters simultaneously.[77] This is achieved through the concept of fuzzy membership, where each data point is assigned a membership degree indicating its degree of belonging to each cluster, rather than being strictly assigned to a single cluster. The term 'fuzzy' reflects the inherent ambiguity in defining cluster boundaries. A membership degree is a set of numbers ranging from 0 to 1 that is characteristic of each point. Since membership degrees represent relative associations, their sum across all clusters for a given data point must equal 1. Notably, the membership degree quantifies the association of a data point with each cluster. The procedure starts with determining C, the number of clusters. Together with the number of data N, the membership degree matrix U is defined by a simple operation $U = C \times N$. Then, the center of the cluster is computed. Unlike *k*-means clustering where centroids are determined, the degree of membership is used as a weight to calculate the weighted average of the points in each cluster. This is followed by updating the membership degrees relative to the cluster centers. Mathematically, this operation can be expressed by the following function (eqn (3)).

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{c} u_{ij}^{m}\left\|x_i - v_j\right\|^2, 1 < m < \infty \quad (3)$$

Where m is the fuzzy partition matrix exponent that controls the ambiguity of the boundaries, u_{ij} is the degree of membership, x_i is the ith member of a cluster, v_j is the center of the cluster, and J is the clustering objective function. This process is iteratively repeated until the cluster centers converge to a stable state with minimal change.

2.2.1.3 Density-based clustering. Another widely used method of clustering is to group the points according to density, computed by density functions.[78] A density-based clustering algorithm used extensively in wearable biosignal analysis is density-based spatial clustering of applications with noise (DBSCAN, Fig. 3d). DBSCAN facilitates two parameters: ε and MinPts. ε is the minimum radius of the neighborhood around a data point, determining how close points need to be to be in a cluster. MinPts is the minimum number of data points required to form a cluster.

Also, to define a cluster, points are divided into 3 groups: (1) core points, which lie within the cluster, (2) border points, which are located at the cluster's boundary, and (3) noise points, which do not belong to either of the first two groups. To form clusters, a data point is first selected. Then, neighboring points within ε are found about the selected data point. If the number of neighboring points exceeds MinPts, a new cluster is formed, and those points are determined as core points. The neighboring points of core points are also added to the cluster, and this expansion process continues until no further points can be included. This set of procedures is repeated until all data points have been processed. The primary advantage of DBSCAN lies in the simplicity of operation. Unlike traditional clustering methods, DBSCAN does not require prior specification of the number of clusters and is capable of detecting nonlinear cluster structures within a dataset. However, its performance is highly dependent on the appropriate selection of ε and MinPts, which can be challenging, particularly in high-dimensional spaces where density estimation becomes less reliable.

2.2.1.4 Variants of clustering methods. Graph clustering models data as a network of interconnected nodes, where edges represent relationships or similarities between these nodes.[79] The core principle of graph clustering lies in representing data in the form of a graph, where nodes (or vertices) denote data points, and edges signify the connections or similarities among these nodes (Fig. 3e). The strength of these connections can be quantified using edge weights, which help identify tightly connected groups or communities within the network, higher weights indicating stronger connections. This modeling approach allows for the detection of intrinsic data structures that traditional clustering methods often overlook. Graph clustering techniques are particularly effective in identifying densely connected subgroups within large-scale graphs. The fundamental mathematical tools in graph clustering involve matrices, such as adjacency matrices, degree matrices, and Laplacian matrices, which capture the relationships between nodes.[75] Common algorithms used in graph clustering include spectral clustering and community detection algorithms, which leverage these matrix representations to identify clusters effectively.

Model-based clustering adopts a probabilistic approach to identify clusters within data by assuming that the data are generated from a mixture of underlying probability distributions, typically Gaussian.[80] Each cluster corresponds to one of these distributions, characterized by parameters such as mean and covariance. A key method in model-based clustering is the Gaussian mixture model (GMM), which models data as a combination of multiple Gaussian components (Fig. 3f). Each data

point is assigned to the Gaussian distribution with the highest probability. The expectation-maximization (EM) algorithm is commonly employed to iteratively estimate the parameters of these distributions, alternating between calculating the probability of data points belonging to each cluster (E-step) and updating the parameters to maximize the likelihood of the observed data (M-step). Model selection criteria, such as the Bayesian information criterion (BIC), are used to determine the optimal number of clusters by balancing model complexity and fit. However, it can be computationally intensive when dealing with large datasets.

Deep clustering techniques excel in handling non-linear relationships and unstructured data, making them suitable for applications such as image recognition, speech processing, and natural language understanding.[81] Despite their advantages, these methods require large amounts of data and computational resources, and their performance can be sensitive to network architecture and hyperparameter settings. Nonetheless, neural network-based clustering represents a powerful advancement in the field, offering enhanced capabilities for discovering intricate patterns in data. Neural networks are also used for clustering through models like self-organizing maps (SOM), learning vector quantization (LVQ), and adaptive resonance theory (ART).[82] These models help find natural patterns in data without labels by using competitive learning. For example, SOM keeps spatial relationships between data points, while ART adjusts to new data in real time. Such methods are especially useful in biosignal analysis, image processing, and real-time monitoring.

In summary, clustering techniques encompass a diverse set of methodologies, each tailored to specific data characteristics and analytical goals. From traditional approaches like hierarchical and partition-based clustering to advanced methods such as graph clustering, model-based clustering, and deep clustering, these techniques provide powerful tools for uncovering hidden structures within data. As data complexity and volume continue to grow, the integration of ML with clustering algorithms will play an increasingly crucial role in diverse fields, including healthcare, wearable devices, and artificial intelligence (AI).[101]

2.2.2 Regression. Regression models are widely utilized for predicting continuous variables, such as heart rate, blood pressure (BP), and blood glucose levels, by training with labeled data.[89,102,103] The application of regression analysis to biosignal data acquired from biosensors enables the quantification of various biometric parameters and health conditions.[104,105] Furthermore, regression models can mitigate data insufficiency by predicting missing continuous values, thereby enhancing the completeness of datasets.[106,107] These methods are generally categorized into linear and non-linear regression based on the relationship between variables, with a comprehensive discussion of each model provided below (Fig. 4a).

2.2.2.1 Linear regression. Linear regression is one of the most fundamental ML models, particularly suitable when a linear relationship exists between a dependent variable and one or more independent variables (Fig. 4b).[108,109] The linear regression model is mathematically represented by the equation (eqn (4)):

$$Y = \beta_0 + \beta_1 X \tag{4}$$

where β_0 represents the intercept, and β_1 denotes the coefficient for the independent variable X. The optimal values of β_0 and β_1 are determined by minimizing the mean squared error (MSE), given by (eqn (5)):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

where n, y_i, and $\hat{y}_i$ represent the number of data points, the actual value of data, and the predicted values, respectively.

Linear regression has been applied to analyze various linear relationships in biosignals, such as the relationship between cortisol and glucose, as well as between diaphragm depth and respiratory rate.[83,84] Furthermore, Baik *et al.* conducted a quantitative analysis of the relationship between skin color changes and skin pH, employing linear regression to develop a model that accurately measures skin pH levels.[85] The key advantages of linear regression include its computational efficiency, rapid execution, and straightforward interpretation of linear relationships between variables. However, this method is not well-suited for modeling non-linear data. Additionally, when there is high multicollinearity among the independent variables, the model may suffer from overfitting, leading to diminished predictive performance.

Ridge linear regression is a method that incorporates a regularization term into the standard linear regression model to mitigate overfitting and improve the model's predictive accuracy.[110,111] In conventional linear regression, the model is optimized by minimizing residual sum of square (RSS) based on the training data. However, this approach may result in increased RSS across all data points, including those in the test set. To address this limitation, ridge regression introduces a shrinkage penalty, which is the squared value of the regression coefficient (β_1), into the objective function (Fig. 4c). The resulting objective function is as follows (eqn (6)):

$$\text{Objective function} = \mathrm{RSS} + \lambda{\beta_1}^2 \tag{6}$$

where λ is a tuning parameter that regulates the strength of the regularization. This term governs the trade-off between model complexity and accuracy, with larger values of λ shrinking the regression coefficients, thereby simplifying the model. Conversely, smaller values of λ reduce the regularization effect, making the model approach the behavior of a standard linear regression model. For instance, Song *et al.* collected sweat alcohol concentration and heart rate data using 3D-printed epifluidic electronic skin and applied ridge regression to enhance the accuracy of predicting reaction time to respond to a given stimulus and commission errors.[86]

2.2.2.2 Non-linear regression. Regression tree model is a tree-based ML algorithm designed to estimate the value of a continuous dependent variable, particularly when a non-linear relationship exists between variables.[87] It partitions the data

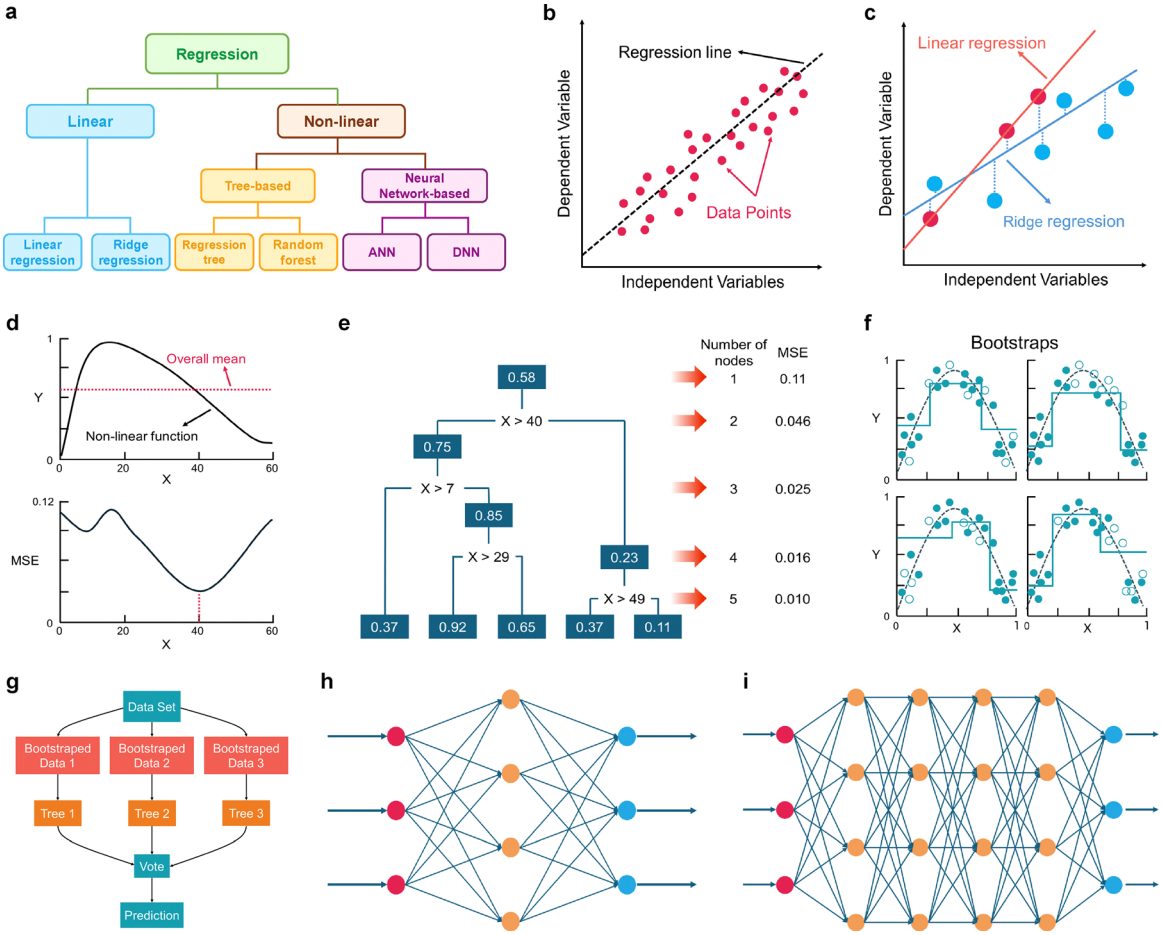


Fig. 4 Regression. (a) Categorizing regression models in ML. (b) Linear regression to represent a linear relationship between variables. Red points and blue points denote training data and test data, respectively. (c) Ridge regression with regularization to prevent overfitting. (d) A non-linear function for applying regression tree (top), minimization of MSE for determining optimal partition positions in a regression tree (bottom). (e) The full regression tree for the prediction of non-linear function. For each partition, nodes are added one by one and MSE decreases. (f) Four distinct bootstrap samples derived from a single dataset and their corresponding regression tree predictions. (g) Random forest, which is combination of multiple regression trees to create more accurate and robust predictive model. (h) ANN consisting of an input layer (red), hidden layer (orange), and an output layer (blue). (i) DNN as an advanced form of ANN, with additional hidden layers to handle more complex data analysis.

into subsets and assigns the average value of each subset as the predicted outcome, aiming to minimize MSE during the partitioning process. For example, a continuous non-linear function representing the relationship between X and Y can initially be approximated using the overall mean, resulting in an MSE of 0.11. However, partitioning the data into subsets and minimizing the weighted average of their respective MSE values enhances predictive accuracy (Fig. 4d). The optimal partition occurs at $X = 40$, which establishes the first branching point in the regression tree. Each node represents a subset characterized by the mean of the dependent variable, with values of 0.75 for $X \leq 40$ and 0.23 for $X > 40$. Further recursive partitioning refines the model, ultimately yielding a regression tree with five terminal nodes and a substantially reduced MSE of 0.010 (Fig. 4e). In this way, regression tree models effectively capture non-linear relationships and provide intuitive visual interpretation. However, they are prone to overfitting when the tree is excessively deep, as small modifications in the tree structure may lead to significant changes in predictions.

To address these limitations, a random forest model can be utilized. This ensemble learning algorithm, based on regression trees, enhances predictive performance by aggregating the outputs of multiple trees, each trained on a randomly selected subset of the data through bootstrap sampling (Fig. 4f).[112,113] This process reduces the correlation between individual trees, allowing each tree to make predictions based on distinct features of the data. Ultimately, the random forest model combines the predictions from multiple trees, resulting in a more robust and stable model that mitigates overfitting and enhances accuracy (Fig. 4g). Lee *et al.* leveraged random forest regression to conduct a quantitative analysis of color variations in colorimetric sensors, which enabled more precise predictions of pH and glucose concentrations.[88] However, the model has drawbacks, including long training times due to the large number of trees and reduced interpretability due to its complex structure.

Neural network-based machine learning techniques, which have also demonstrated effectiveness in clustering tasks, are

widely adopted in regression modeling to capture complex and non-linear relationships in biosignal data. Additionally, an artificial neural network (ANN) is a computational model inspired by biological neural networks. ANN is designed to process input data and identify patterns to facilitate regression tasks. It consists of an input layer that receives data, one or more hidden layers that process the data and extract complex features, and an output layer that produces the final prediction (Fig. 4h). ANNs typically contain one or two hidden layers and are suitable for relatively simple nonlinear data analysis.[114] For example, Wang *et al.* employed ANNs to train on photoplethysmography (PPG) signals and corresponding BP data, effectively modeling the non-linear relationship between PPG signals and BP.[89] As a result, they were able to achieve a high level of accuracy in predicting BP, with a mean absolute error of 4.02 ± 2.79 mmHg for systolic BP. However, ANNs have limitations in capturing relationships between highly complex variables, which may affect their performance in more intricate modeling tasks.

A deep neural network (DNN) is an advanced extension of an ANN that incorporates multiple hidden layers (Fig. 4i).[115] This deeper architecture enables the model to learn more complex and abstract patterns, enhancing its ability to make accurate predictions. Haleem *et al.* employed a DNN-based multi-layer perceptron model for real-time blood glucose prediction.[90] The model utilized morphological features automatically extracted from ECG signals and achieved an accuracy of approximately 89% in blood glucose estimation.

As described above, applying regression models enables more effective continuous health monitoring and disease diagnosis by facilitating real-time assessment of physiological states. However, as the suitability of a regression model depends on the underlying data characteristics and relationships, selecting an appropriate model tailored to the specific biosignal data is critical for ensuring optimal predictive performance.

2.2.3 Classification. Classification is a process of assigning data samples with labels to one of several predefined classes. This technique is utilized for tasks such as disease diagnosis and condition monitoring, which involve predicting discrete outcomes.[116,117] Various classification models are applied based on the structure and purpose of the data. In this section, we explore key classification models such as logistic regression, support vector machine (SVM), *k*-NN, decision tree, random forest, and gradient boosting, discussing their characteristics and applications, particularly in analyzing biosignal datasets (Fig. 5a).

2.2.3.1 Linear and Kernel-based models. Logistic regression is a commonly utilized linear model for binary classification,

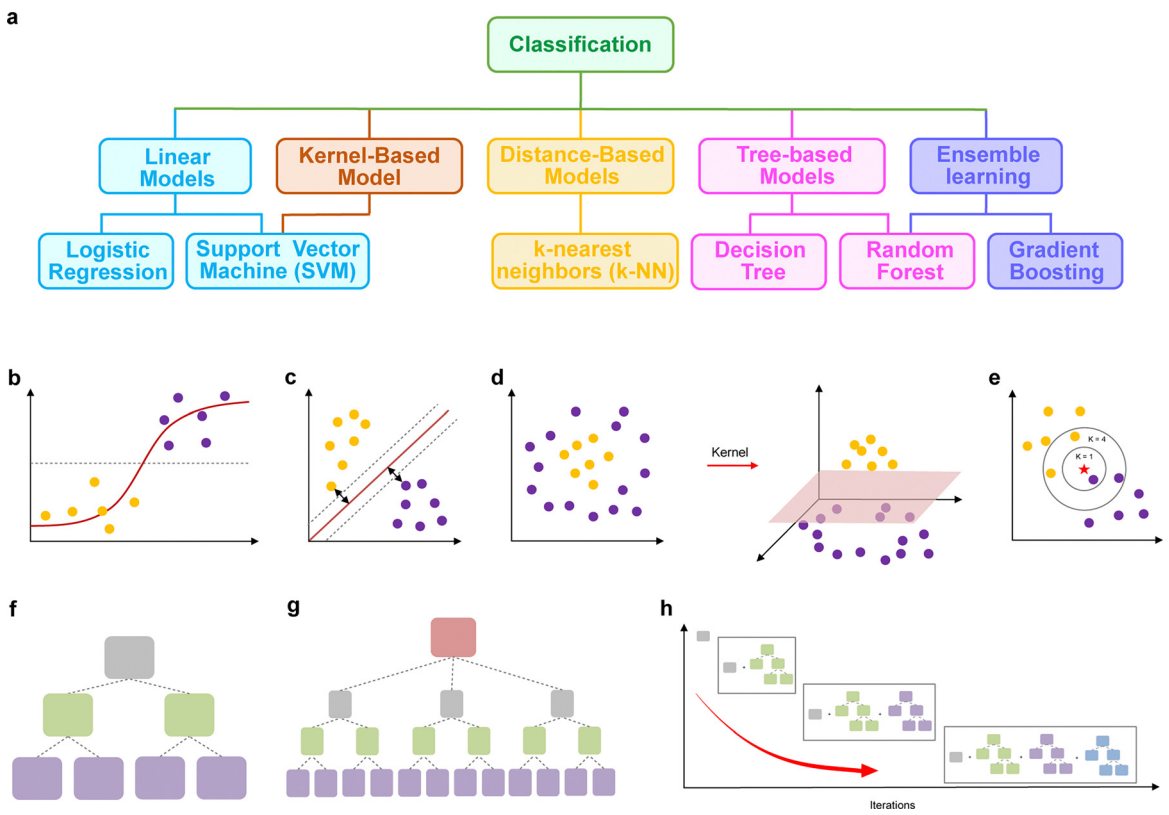


Fig. 5 Classification. (a) Categorizing classification models in ML. (b) Logistic regression with a sigmoid curve indicating the probability of binary outcomes. (c) Linear support vector machine (SVM) illustrating decision boundaries for binary classification. (d) Kernel SVM projecting data into a higher-dimensional space to achieve separability. (e) *k*-Nearest neighbors (*k*-NN) classification based on proximity to *k* nearest points. (f) Decision tree visualizing a hierarchical data-splitting structure. (g) Random forest aggregating predictions from multiple decision trees. (h) Gradient boosting iteratively refining predictions to minimize residual errors.

establishing the correlation between input and output variables by utilizing the sigmoid function to predict probability values (Fig. 5b). The sigmoid curve (red line) illustrates how the model transforms raw input data into a probability distribution, with the threshold (dotted line) determining the class boundary. A class is assigned based on a predefined threshold (typically 0.5). Additionally, softmax regression allows logistic regression to be adapted for multi-class classification. It employs maximum likelihood estimation (MLE) to optimize parameters and binary cross-entropy as the loss function.[118] Its advantages include simplicity, fast training speed, and probability-based predictions. However, its linear decision boundary makes it unsuitable for non-linear data classification, and its sensitivity to outliers and multicollinearity can adversely affect classification accuracy. Several studies have employed logistic regression to predict disease diagnosis and progression.[119,120] For example, Kim *et al.* applied logistic regression to predict the progression of Alzheimer's disease (AD) and this model reduced the complexity of conventional diagnostic methods while effectively handling both continuous and categorical data, enabling fast classification.[91] Using multiple reaction monitoring-mass spectrometry (MRM-MS), the researchers analyzed blood samples for 10 protein biomarkers and APOE genotypes. Logistic regression successfully classified Aβ-positive and Aβ-negative groups through binary classification and predicted AD progression (three stages) using multinomial classification combined with Korean mini-mental status examination (K-MMSE) scores. This demonstrated logistic regression's effectiveness in combining biomarker and genetic data for disease progression prediction.

SVM is a supervised learning model that separates data using a hyperplane, especially effective for binary classification.[121] The primary goal of SVM is to determine the most suitable hyperplane that maximizes the margin between classes, ensuring high generalized performance. As illustrated in Fig. 5c, linear SVM determines an optimal decision boundary (solid red line) that maximizes the margin (dotted lines) between two classes. However, when the data is not linearly separable, a more advanced approach is required. For instance, Kernel SVM data is mapped into a higher-dimensional space through a kernel function, allowing for a linear decision boundary in that transformed space (Fig. 5d). These SVM models excel in high-dimensional data with robust overfitting prevention. However, it requires optimization of kernel functions and hyperparameters (C, γ), and training may be slow for large datasets. One study utilized SVM to classify cognitive and emotional engagement levels of students using EEG data.[92] EEG from 21 students was recorded with standard stimuli including continuous performance tests (CPT), music background, and social feedback. This study applied an SVM with a radial basis function (RBF) kernel to address non-linear characteristics, effectively analyzing complex EEG patterns and assessing the cognitive and emotional engagement levels.

2.2.3.2 Distance-based models. *k*-NN is a non-linear model that assigns a data point by analyzing the k closest data points in the dataset.[122] This method applies to both binary and multi-class classification, particularly in environments where the distribution of data points is critical in interpreting the data. *k*-NN stores data without pre-training and performs classification by calculating distances during prediction. This simplicity enables fast calculation and flexible application to diverse data structures. Fig. 5e illustrates *k*-NN classification, where a new data point (red star) is assigned a class based on its *k*-nearest neighbors. With $k = 1$, the classification depends only on the closest neighbor, making the model highly sensitive to noise and resulting in high variance. In contrast, $k = 4$ considers more neighbors, producing a more generalized decision. For example, Görür *et al.* developed a tongue movement-based interface (TMI) to assist people with disabilities in controlling assistive devices.[93] TMI used EEG-based glossokinetic potential (GKP) signal to classify the tongue movements as left or right *via* binary classification of *k*-NN model. Owing to the capability of *k*-NN model to quickly classify with small datasets, the prediction results effectively reflected the complex characteristics of biosignal data. However, *k*-NN is valuable to high-dimensional datasets, reducing distance calculation reliability. Larger datasets also lead to delaying the predictions due to increased computational requirements. As the prediction by *k*-NN relies on the distances between the data points, preprocessing the data with normalization or standardization helps the model to show better performance. This process ensures equal contribution from all dataset features, ensuring no single feature dominates the distance computation. Additionally, the choice of k significantly impacts the output results. For example, small k values risk overfitting, while large k values may cause underfitting. Thus, choosing the optimal number of k by comparing the output with different k is essential to draw the best prediction result.

2.2.3.3 Tree-based and ensemble models. Decision tree is a non-linear model that classifies data by using a hierarchical tree structure for decision-making.[94] Each split point represents a condition on a specific feature, and leaf nodes represent final classes or values. As shown in Fig. 5f, a decision tree starts from a root node (gray), splits into internal nodes (green) based on feature conditions, and finally reaches the leaf nodes (purple) that determine the classification output. Decision tree is intuitive, easy to interpret, and capable of processing both continuous and categorical data without prior assumptions about data distribution. It supports missing value handling and multi-class classification while offering fast training and prediction. However, decision trees are prone to overfitting and can exhibit instability, as small changes in data may significantly alter the tree structure. Performance may also degrade in high-dimensional data due to bias-variance tradeoffs. These limitations can be addressed by ensemble methods like random forest and gradient boosting.

Random forest is an ensemble learning model that trains multiple decision trees in parallel and aggregates their predictions using majority voting for classification tasks. Each tree is trained on randomly sampled data through bootstrap aggregating (bagging) and is split based on randomly selected features, reducing correlations between trees.[113] This enhances generalization performance,

prevents overfitting, and improves prediction stability. As shown in Fig. 5g, random forest comprises several decision trees, each trained on distinct data subsets. The model starts from a root node (red) and branches into several trees (gray), which classify data independently. The final prediction is obtained by aggregating individual tree outputs, enhancing robustness and reducing variance compared to a single decision tree. Random forest performs well in binary, multi-class, and multi-label classification, in addition to regression problems. It also calculates feature importance during training, aiding data interpretation. However, random forest requires more computational resources and memory as dataset size increases, and interpretability is lower compared to individual trees. Yaari *et al.* utilized random forest for early detection of ovarian cancer using DNA-singlewall carbon nanotubes (SWCNTs)-based optical nanosensor arrays.[95] Protein biomarkers in uterine lavage samples were analyzed *via* random forest model to perform binary classification for single biomarkers, multi-class classification for biomarker combinations, and multi-label classification for coexisting biomarkers. This comprehensive analysis of biomarkers with random forest enables the detection of multiple biomarkers in biofluids through DNA-SWCNT nanosensor array, facilitating the prediction of the presence of each biomarker.

Gradient boosting is an ensemble learning model that sequentially trains multiple decision trees, improving performance by learning from errors of previous trees. It operates by calculating the gradient of the loss function and progressively minimizing residuals, capturing complex data patterns and achieving high prediction accuracy.[123] As shown in Fig. 5h, gradient boosting iteratively refines predictions by adding trees that correct the errors of prior models. The red arrow represents the progressive minimization of residual errors, leading to enhanced model performance over iterations. The model starts with a weak learner (left), and as more trees are added, it gradually improves classification accuracy. Gradient boosting provides flexibility in defining loss functions, making it suitable for regression and classification tasks. Its iterative refinement also enhances robustness to noisy or incomplete data, making it well-suited for real-world applications. Implementations such as eXtreme Gradient Boosting (XGBoost) further enhance performance through parallel processing and regularization. For example, one study demonstrated detecting epileptic seizures from EEG data in real-time through gradient boosting.[96] Gradient boosting effectively separated EEG signals from noise and reduced false alarms, enabling reliable seizure detection and artifact detection (EEG normal/artifact) with an average sensitivity of 65.27%. However, Gradient boosting requires careful hyperparameter tuning, which is computationally intensive and sensitive to data noise, necessitating overfitting prevention measures.

2.3 Model evaluation metrics

Ensuring reliable performance is critical in ML-based healthcare applications, where incorrect predictions can lead to misdiagnoses and directly impact patient health. This section explores key evaluation metrics and techniques used to assess the reliability of ML models. The choice of evaluation method depends on the nature of the problem—classification, regression, and clustering (Fig. 6a).[124] Additionally, we discuss approaches for assessing a model's generalization capability, ensuring robustness across diverse physiological conditions and biosignal variations.[125]

Fig. 6 Model evaluation metrics. (a) Categorization of model evaluation metrics. (b) Confusion matrix. (c) Receiver operating characteristic curve. (d) Precision–recall curve. (e) Schematic illustration of the *K*-fold cross-validation process.

2.3.1 Model evaluation in classification. Evaluating classification performance in ML relies on assessing the proportion of correct predictions. Several key metrics, derived from fundamental concepts such as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), provide insights into model effectiveness.[126,127] As illustrated in Fig. 6b, the confusion matrix offers a visual representation of the relationship between predicted and actual classes, facilitating a comprehensive assessment of classification performance.

One of the most fundamental metrics derived from the confusion matrix is accuracy, which calculates the proportion of correctly classified instances, including both TP and TN, out of the total predictions, as defined in eqn (7).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{7}$$

While accuracy provides an intuitive measure of model performance, it can be misleading in cases of severe class imbalance. For instance, in a cancer patient detection model, where the ratio of healthy individuals (negative) to cancer patients (positive) is 9 : 1, a model that predicts all samples as healthy would achieve a 90% accuracy. However, this model fails to identify any actual cancer patients, rendering it ineffective for medical applications.

To address the limitations, additional metrics such as recall, precision, and the F1-score are employed. Recall, defined in eqn (8), quantifies the proportion of actual positive cases that the model correctly identifies.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

In the previous cancer detection example, the model exhibits a recall of 0%, meaning it fails to identify any cancer patients despite achieving high accuracy. This highlights the importance of recall in medical diagnostics, as a model with low recall may miss critical cases, leading to undetected patients who do not receive necessary treatment. High-recall models are therefore essential in healthcare applications to ensure that patients are properly diagnosed and treated.

Precision measures the proportion of correctly predicted positive cases out of all positive predictions made by the model (eqn (9)).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

Precision represents how many of the model's positive predictions are actually correct. A high-precision model minimizes FP, which is crucial for preventing unnecessary medical interventions or misdiagnoses.

On the other hand, precision and recall often exhibit a trade-off, requiring a balanced evaluation depending on the application. In critical medical diagnostics, recall is typically prioritized to ensure that no patient goes undetected, whereas precision becomes more important in cases where false positives could lead to harmful consequences, such as unnecessary treatments or psychological distress. The F1-score offers a comprehensive metric by harmonizing recall and precision, especially valuable in datasets with class imbalances. It is computed as the harmonic mean of precision and recall (eqn (10)), providing a single score that reflects both false positives and false negatives.

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{10}$$

Beyond these metrics, evaluating the overall performance of classification models requires a more comprehensive assessment, particularly in imbalanced datasets. One widely used method is the receiver operating characteristic (ROC) curve, which plots the TP rate—equivalent to recall—against the FP rate (FPR) (eqn (11)).[128]

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{11}$$

As illustrated in Fig. 6c, the area under the curve (AUC) in the ROC curve serves as a global performance measure, where values closer to 1 indicate a more effective model. A model whose ROC curve approaches the top-left corner of the plot demonstrates strong classification ability.

However, in datasets with significant class imbalances, where negative instances vastly outnumber positive ones, the ROC curve may overestimate model performance. This is because the FPR is directly influenced by the total number of negative samples (TN + FP). In such cases, the precision–recall (PR) curve provides a more informative alternative, particularly in scenarios where detecting rare conditions is critical. Since precision is independent of the total number of negative instances, the AUC in PR curve offers a more reliable assessment of a model's effectiveness in imbalanced datasets (Fig. 6d). Consequently, unlike the ROC curve, which considers both positive and negative classifications, the PR curve focuses solely on the model's ability to identify the minority class.

When applying classification models to biosignal analysis—especially in wearable health monitoring systems—it is essential to select evaluation metrics that align with the data characteristics and model objectives.[129] Given the potential impact of class imbalance and the critical need for accurate health monitoring, metrics such as recall, F1-score, and PR-AUC often provide more meaningful insights into model performance than accuracy alone.

2.3.2 Model evaluation in regression. In regression problems, model performance is evaluated by quantitatively measuring the difference between predicted values and actual values.[130] Unlike classification, accuracy is not suitable for regression tasks that involve continuous predictions. This section outlines five proper evaluation metrics for regression models: mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), mean percentage error (MPE), and the R^2 score.[131]

MAE is one of the most intuitive metrics for evaluating regression models. It calculates the average of the absolute differences between predicted and actual values (eqn (12)).

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{12}$$

where y_i, $\hat{y}_i$, and n represent the actual value, the predicted value, and the number of samples, respectively.

Since it only measures the magnitude of errors, MAE is less sensitive to outliers, making it a useful metric when dealing with datasets containing some extreme values.

MSE addresses errors by squaring the differences between predicted and actual values, and then averaging them as defined in eqn (4).

Squaring the errors amplifies larger discrepancies while reducing the impact of smaller ones, making MSE highly sensitive to large errors. This sensitivity is particularly necessary in scenarios where substantial prediction errors can lead to severe consequences. For example, in diabetic patients, a sudden spike in blood glucose levels serves as a critical indicator requiring immediate insulin administration. Inaccurate predictions in such cases could result in delay treatment, increasing the risk of severe complications. In such high-stakes applications, MSE is a valuable metric for ensuring that large errors are minimized. To express errors in the same units as the actual values, RMSE is used, which is the square root of MSE (eqn (13)).

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{13}$$

RMSE provides a more interpretable metric for comparing model performance, particularly when a direct understanding of error magnitude is necessary.

While MAE, MSE, and RMSE are scale-dependent metrics, they do not provide information about whether the model tends to overestimate or underestimate values. This limitation complicates model comparisons across datasets with different units and makes it difficult to determine the directional bias of predictions. To address these issues, MPE and the R^2 score provide insights into the directionality of errors and facilitate comparisons across different datasets.

MPE calculates the average of the percentage differences between predicted and actual values, providing a sense of the model's bias (eqn (14)).

$$\mathrm{MPE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i} \times 100\right) \tag{14}$$

However, MPE has notable limitations. Since it averages both overestimations and underestimations, these errors can offset each other, distorting the evaluation of model performance. Additionally, when actual values are close to zero, MPE can produce excessively large values, rendering it unreliable for performance assessment.

R^2 score is a widely used metric for evaluating the overall performance of regression models. It measures how well the model explains the variance in the actual values (eqn (15)).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}\left(y_i - \underline{y}\right)^2} \tag{15}$$

where $\underline{y}$ represents the mean of actual values.

For instance, an R^2 value of 0.9 in a blood glucose prediction model indicates that the model explains 90% of the variance in the patient's glucose levels. An R^2 score close to 1 suggests strong predictive performance, whereas an R^2 of 0 indicates that the model performs no better than simply predicting the mean of the target values. Additionally, negative R^2 values imply that the model performs worse than a baseline prediction using the mean, suggesting poor model fit.

However, R^2 has certain limitations, particularly in cases of overfitting, where the model performs exceptionally well on training data but poorly on unseen data. To mitigate this issue, adjusted R^2 is often used (eqn (16)).[132]

$$R_{\mathrm{adjusted}}{}^2 = 1 - \left(1 - R^2\right)\frac{n-1}{n-k-1} \tag{16}$$

where k represents the number of independent variables.

The adjusted R^2 score penalizes excessive model complexity, reducing its value as more predictors are added unless they provide substantial explanatory power. Notably, adjusted R^2 is always lower than or equal to R^2, ensuring that adding irrelevant variables does not artificially inflate model performance. A higher adjusted R^2 indicates that the model effectively explains data variability while including only the necessary predictors, making it a more reliable metric for model evaluation.

2.3.3 Model evaluation in clustering. Evaluating the performance of a clustering model differs significantly from that of supervised learning models. Since clustering is an unsupervised learning approach without predefined ground truth labels, its effectiveness cannot be directly measured using accuracy or other label-dependent metrics. Instead, various evaluation metrics assess how well the model groups the data, focusing on factors such as cluster cohesion, separation, and overall clustering quality.[133,134]

One commonly used metric is the Silhouette coefficient, which simultaneously considers the cohesion within clusters and the separation between different clusters. Cohesion refers to how closely related the data points in a cluster are, while separation measures how distinct a cluster is from others. The Silhouette coefficient for a single data point is computed using eqn (17).

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{17}$$

where i represents an individual data point, a is the average intra-cluster distance, and b is the average nearest-cluster distance.

The Silhouette coefficient ranges from -1 to 1, where higher values suggest better clustering quality.

In certain cases where ground truth labels are available, label-dependent metrics can be applied to evaluate clustering performance. For example, in a hospital setting with large-scale patient data, some patients may have received a confirmed diagnosis, while others remain undiagnosed. In such scenarios, clustering can be utilized to automatically group patients based on their clinical data, and clustering performance can be assessed using metrics commonly used in supervised learning, such as TP, TN, FP, and FN.

One such metric is the rand index (RI), which measures the similarity between the clustering results and actual labels by

evaluating the proportion of data point pairs that are correctly grouped together or correctly separated. The evaluation formula for RI follows the same structure as accuracy in classification. While RI provides a straightforward measure of clustering performance, it has a significant limitation: random clustering can still yield relatively high RI values. This limitation makes RI less reliable in some contexts.

To address this issue, the adjusted rand index (ARI) is introduced. ARI adjusts the RI by accounting for the expected RI (E[RI]) value under random clustering, thereby eliminating the influence of chance. It achieves this by subtracting the E[RI] from the observed RI and normalizing the result based on the best possible clustering outcome, as shown in eqn (18).

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{(\mathrm{RI}) - E[\mathrm{RI}]} \tag{18}$$

Unlike RI, which ranges from 0 to 1, ARI takes values between −1 and 1. An ARI value close to 1 indicates that the clustering results closely match the original labels, signifying near-perfect clustering performance. Conversely, an ARI value near −1 suggests that the results exhibit a structure that is opposite to the ground truth labels. A value of 0 implies that the clustering assignment are equivalent to random labeling.

In applications such as patient subgroup identification or gene expression clustering, the ARI provides a more reliable assessment of clustering quality, especially when some ground truth labels are available. Selecting appropriate clustering evaluation metrics is crucial for biosignal analysis and health monitoring applications, ensuring that the models not only group data effectively but also provide meaningful and actionable insights.

2.3.4 Evaluation of generalization performance. Beyond evaluating performance in classification, regression, and clustering, it is essential to ensure that a ML model generalizes well to new, unseen data rather than just performing well on the training set.[135,136] Assessing generalization performance prevents overfitting and provide effectiveness in real-world applications. In this section, we introduce several widely used methods, including train-test split, cross-validation (CV), and bootstrap sampling.

The train-test split is the simplest evaluation method, where the dataset is divided into two subsets: a training set and a test set.[137] Typically, 70–80% of the data is used for training, while the remaining 20–30% is reserved for testing. This method is quick and easy to implement, but it has limitations due to its reliance on a single data split, which may lead to variability in evaluation results depending on how the data is partitioned. To mitigate this issue, techniques such as stratified sampling can be employed to maintain the distribution of classes in both training and test sets or the process can be repeated multiple times with random splits to average the results.

CV is a technique that partitions the dataset into multiple subsets and repeatedly evaluates the model's performance across these subsets. This approach maximizes data utilization and enhances the reliability of the model evaluation. One of the most common CV techniques is K-fold CV, where the dataset is divided into K equally sized folds. Each fold is used once as a validation set while the remaining $K - 1$ folds serve as the training set. This process is repeated K times, and the final performance is calculated by averaging the results from all iterations. For instance, in 5-Fold CV, the data is split into five parts, and the model is trained and validated five times, ensuring that every data point is used for both training and validation (Fig. 6e). This method provides a more generalized performance evaluation by utilizing the entire dataset.[138]

Another metric is leave-one-out cross-validation (LOOCV), where each data point is used as a validation set while the remaining data serves as the training set. This process is repeated for every data point, and the results are averaged. LOOCV is highly effective when dealing with small datasets, but it can be computationally expensive for larger datasets due to the high number of iterations required.

Bootstrap sampling is another important method for evaluating generalization performance. This technique involves sampling with replacement to create multiple random subsets from the original dataset. The model is then trained and evaluated on these subsets, and the performance is assessed using data points that were not included in the training set, known as out-of-bag data. Bootstrap sampling is particularly useful in situations with limited data, as it increases the reliability of the model's evaluation and helps estimate model variance. However, the repetitive nature of sampling and training can lead to higher computational costs.

Each of these evaluation methods has its unique characteristics, advantages, and limitations. CV is effective in maximizing data usage and preventing overfitting but comes with higher computational costs. Train-test split offers a fast and straightforward evaluation, though it may suffer from variability due to data partitioning. Bootstrap sampling enhances model reliability and uncertainty estimation but can be computationally intensive due to repeated sampling.

Selecting the appropriate evaluation method depends on the specific problem and the amount of available data. For instance, CV is ideal for ensuring robust performance in large datasets, while bootstrap sampling is advantageous when dealing with small datasets. Understanding the strengths and weaknesses of each method enables more accurate assessments of a model's generalization capability.

In conclusion, evaluating the generalization performance of ML models is a crucial process for verifying predictive accuracy of new data. By selecting and applying suitable evaluation methods, models can be optimized to deliver reliable and consistent performance in real-world scenarios, ultimately ensuring their effectiveness in practical applications.

3. Machine learning applications in biosignal processing for health monitoring

3.1 Neural signals

The continuous monitoring of neural signals in patients with neurological disorders is essential for their well-being.[139–143]

Wireless, wearable sensors enable real-time neural monitoring, capturing signals ranging from neuron-level single unit-potentials and local field potentials (LFPs) to larger-scale electrocorticograms (ECoGs) and EEGs.[144–146] However, interpreting raw neural signals is challenging due to biological artifacts such as eye blinking or respiratory patterns, necessitating post-processing for clarity and accurate diagnosis. Additionally, the vast amount of data collected from these sensors makes manual analysis impractical.[147–149] To address these challenges, ML algorithms are increasingly being utilized to efficiently process large-scale data,[150] detect critical signals,[151] and enable closed-loop drug delivery,[152] improving both diagnostics and therapeutic interventions.[153–156]

Conventional ML algorithms are widely used to classify input variables, particularly in biomedical applications. In supervised learning, input data, such as biosignals and biomarkers, is used to predict response variables, including sleep stages and disease phenotypes.[157,158] The integration of these algorithms into wearable sensors enhances real-time processing, enabling the development of high-accuracy health monitoring systems for continuous and personalized healthcare.[159–161] Xu *et al.* have developed an in-ear wearable sensor capable of continuously monitoring electrophysiological signals and classifying brain states (Fig. 7a).[162] To evaluate its performance, alpha modulation, which is spontaneous EEG activity in the 8–12 Hz frequency range, was analyzed at four-time points: one pre-exercise and three post-exercise measurements. Classification, as described in Section 2.2.3, was used to categorize the brain states. A filter-bank-based common-spatial-pattern (FBCSP) analysis was

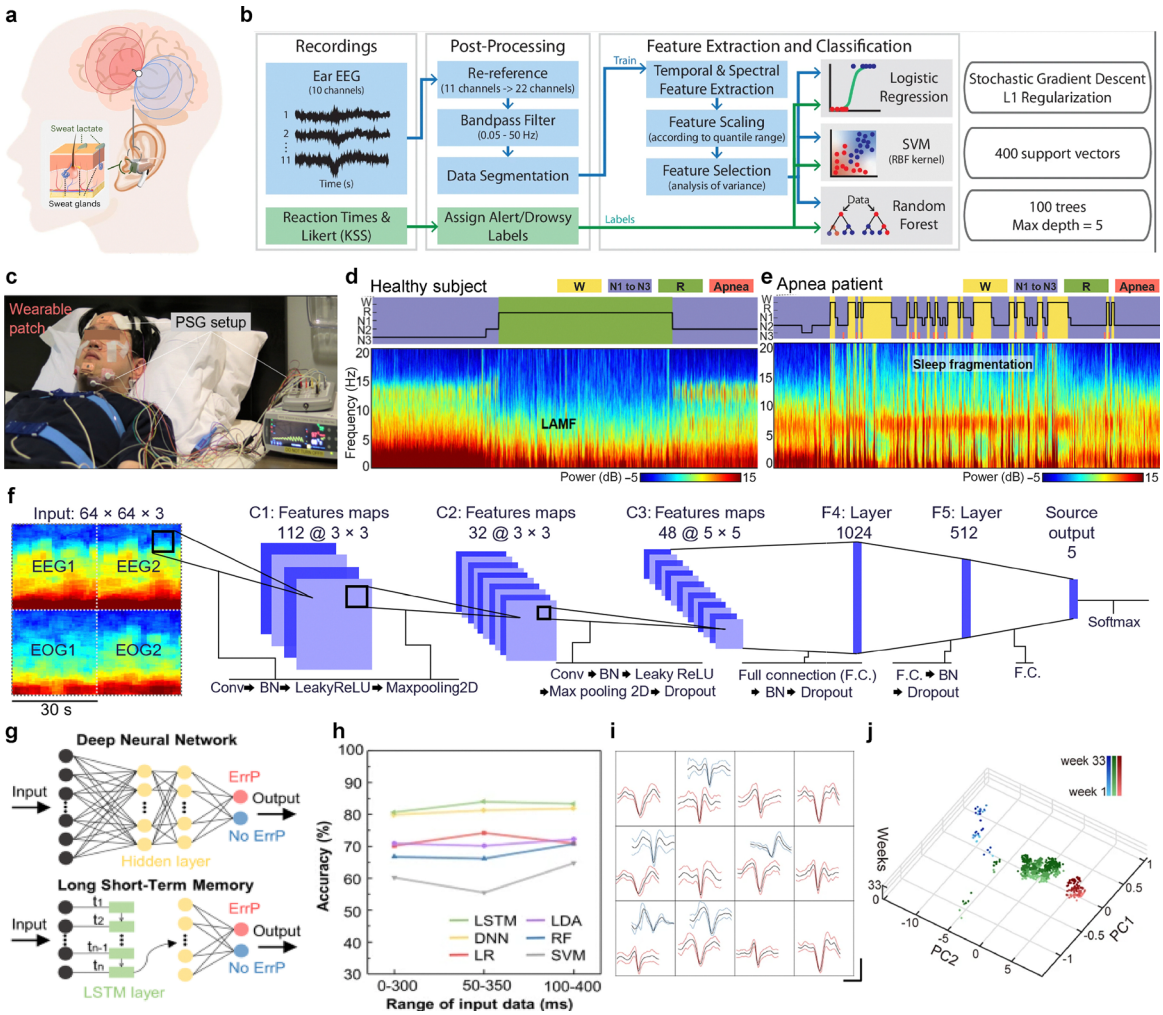


Fig. 7 Applications of wearable sensors for neural signal monitoring. (a) A schematic of the in-ear wearable sensor for EEG monitoring. Reproduced with permission from ref. 162. Copyright 2023, Springer Nature. (b) Processing architecture of the wireless in-ear EEG device for monitoring drowsiness. Reproduced with permission from ref. 163. Copyright 2024, Springer Nature. (c) Image of the wearable sleep patch in comparison to PSG setup for sleep stage classification and apnea detection. (d) and (e) Hypnogram and spectrogram of recorded EEG signals from the wearable sleep patch from a healthy subject and apnea patient, respectively. (f) Illustration of the CNN architecture employed for sleep stage classification. Reproduced with permission from ref. 157. Copyright 2023, AAAS. (g) Illustration of the CNN and LSTM architecture employed for classification of ErrP signals. (h) Classification accuracies of various ML algorithms tested for ErrP signal detection. Reproduced with permission from ref. 164. Copyright 2022, Springer Nature. (i) PCA-clustered single-unit spikes from different regions in a mouse brain. Scale bars, 500 μV (vertical) and 200 ms (horizontal). (j) Time evolution plots of single-unit spikes over 33 weeks. Reproduced with permission from ref. 165. Copyright 2024, Springer Nature.

employed for two-class feature extraction and classification. In the first stage, multiple bandpass filters optimized the signals, followed by common-spatial-filter transformation, which extracted low-dimensional spatial features. A mutual information-based feature extraction technique was used to select the most discriminative features. In the second stage, these features were fed into a SVM for brain state classification. This resulted in a higher classification accuracy (89.14%) for the post-exercise-immediate brain state compared to the post-exercise-relaxed state (64.22%), demonstrating clear distinctions in brain activity before and immediately after exercise.

Beyond signal monitoring, ML can also be used to assess cognitive performance.[166] Kaveh *et al.* designed a wireless in-ear EEG device capable of real-time drowsiness classification using ML algorithms. The device features gold-plated electrodes, with four in-ear electrodes for EEG signal acquisition and two out-ear electrodes serving as reference and ground. The raw EEG signals undergo refinement, including bandpass filtering to remove noise, followed by segmentation into 10 s and 50 s windows for feature extraction.[163] ML algorithms – logistic regression, SVM, and random forest – were employed for alertness and drowsiness identification. Fig. 7b illustrates the full processing pipeline, from EEG signal acquisition to drowsiness classification. To ensure model generalization, three cross-validation techniques were used: user-specific, leave-one-trial-out, and leave-one-user-out. While all three models achieved high accuracy, the SVM classifier outperformed the others with 93.3% accuracy for a never-before-seen user, demonstrating the system's ability to adapt across individuals. By integrating ML, the device enables automated, high accuracy drowsiness detection, making real-time cognitive state monitoring possible.

When processing large amounts of complex data, the decoding process becomes more complex and inconsistent.[167,168] To address this, as mentioned in Section 2.2.1, neural networks are preferred for handling data under such conditions. Kwon *et al.* introduced a wearable sleep patch capable of monitoring electrical signals, including EEGs, electrooculograms (EOGs), and electromyograms (EMGs) by using convolutional neural networks (CNN).[157] Unlike polysomnography (PSG), which requires multiple wired connections, the device enables continuous EEG monitoring with a compact patch on the forehead (Fig. 7c). Real-time EEG recordings allow for automatic sleep stage classification and sleep apnea detection. Fig. 7d presents a hypnogram and spectrogram of a healthy subject, showing uninterrupted sleep, while Fig. 7e visually illustrates repetitive signals corresponding to wake stages and apnea events in an apnea patient. A CNN algorithm was trained on sleep data from 32 healthy participants and 40 apnea patients to analyze EEG signals and assess sleep quality (Fig. 7f). Multiple layers of the CNN, incorporating batch normalization and max pooling layers, enables automated sleep stage classification by extracting relevant patterns from EEG signals. This demonstrates the patch's capability for real-time, automated sleep monitoring and disorder detection. The wearable patch achieved a high prediction accuracy of 88.52%, closely matching the results of PSG conducted by a sleep technician, highlighting its potential as a reliable alternative for sleep assessment.

ML reinforcement for a personalized AI system is also possible through the continuous monitoring of neural signals. Shin *et al.* reported on a wireless, earbud-like EEG measurement device integrated with a brain–AI closed-loop system (BACLoS) for enhancing AI decision making through real-time brain wave analysis.[164] The BACLoS system employs deep learning algorithms to classify and interpret EEG signals, specifically detecting error-related potential (ErrP) signals. ErrP signals occur when the user perceives an unpredicted or incorrect machine response. By detecting these signals, ML enables the system to refine itself autonomously, improving decision-making accuracy over time. Several ML algorithms were tested for ErrP signal classification, with Fig. 7g illustrating the architectures of DNNs and long short-term memory (LSTM) networks. An LSTM is a type of recurrent neural network (RNN) – a sequence-processing model that continually updates an internal memory cell based on past and current inputs. However, it can store data for a longer period of time with an improved remembering capacity compared to a standard recurrent cell. This extended temporal awareness makes LSTMs especially effective when dealing with time-series data such as EEG.[169] Among these, LSTM achieved the highest classification accuracy (83.81%). Fig. 7h presents the classification accuracies of different algorithms including LSTM, DNN, linear regression, linear discriminant analysis, random forest, and SVM, highlighting the system's adaptability in real-time neural signal processing.

As explained in Section 2.1, monitored neural signals undergo preprocessing with the aid of ML to extract meaningful patterns.[170] Park *et al.* reported on a soft neural system designed to monitor single-unit activities in the brain of rodents.[165] The system utilizes liquid metal neural probes, which conform to the brain's structure and enable high-resolution recording of single-unit spikes. These signals are later processed using PCA clustering, which isolates distinct neural units by grouping similar signal features (Fig. 7i). Additionally, time evolution plots of PCA-clustered single-unit spikes indicate that the same neurons were consistently recorded for up to 33 weeks, confirming the long-term stability of the implanted neural probe (Fig. 7j). Research extends beyond preprocessing and monitoring, aiming for the development of closed-loop feedback systems. Ouyang *et al.* introduced a wireless, implantable device capable of autonomous biosignal recording and closed-loop neuromodulation.[152] A CNN based seizure detection model was trained and embedded into the device, assigning seizure scores to 3-second EEG segments. To validate the closed-loop system, epileptic seizures were induced in a rat model through pilocarpine injection. EEG was continuously monitored, and when the seizure score exceeded a predefined threshold over a 30-second window, neuromodulation was triggered, releasing an anti-seizure drug from an onboard drug reservoir.

The advancement of wearable sensors with ML has revolutionized neural signal acquisition, enabling continuous, unobtrusive data collection for applications in cognitive assessment, neurological disorder detection, and closed-loop feedback systems.[18] ML algorithms enhance efficiency and scalability in neural signal processing by optimizing feature extraction

and classification, leading to more accurate and real-time analysis. Ongoing research explores the development of theranostic devices that incorporate ML for human-in-the-loop systems, allowing adaptive, closed-loop interventions for personalized treatment. With continued advancements, neural sensors hold the potential to drive innovations in personalized healthcare, improve neurotherapeutic strategies, and enhance brain–machine interfaces for next-generation medical and consumer applications.

3.2 Cardiovascular signals

Cardiovascular monitoring devices play a crucial role in assessing cardiac health by analyzing key physiological parameters such as ECG, BP, and heart rate (HR). These devices are utilized not only in the medical field for the diagnosis and management of cardiovascular diseases (CVDs) but also in wellness and fitness applications for continuous health monitoring.[171–173] The integration of ML into cardiovascular monitoring enhances the interpretation of these physiological signals, providing several benefits. Firstly, ML algorithms facilitate the detection, classification, and prediction of cardiovascular conditions, enabling early diagnosis and risk assessment. Secondly, ML-based models efficiently process complex ECG data, minimizing noise generation and extracting meaningful features for accurate analysis. Leveraging these advantages, ML-driven cardiovascular devices can deliver real-time feedback in a timely manner to improve patient outcomes. Integrating these functionalities into cardiovascular monitoring systems strengthens personalized healthcare by enabling automated disease detection and real-time diagnostics, even in the absence of healthcare professionals.

One of the most prominent applications of ML in cardiovascular monitoring involves assessing an individual's current physiological state and detecting potential cardiovascular conditions based on real-time sensor data.[174] By leveraging ML techniques, cardiovascular devices can efficiently process physiological signals, enabling continuous monitoring and early disease detection. Yang *et al.* developed a wearable ECG sensor utilizing highly conformal on-skin electrodes fabricated through the interlocking of silk fibroin and conductive polypyrrole (PPy), designed to record real-time ECG signals during two hours of running.[175] The extracted ECG features were analyzed using RNNs to classify the emotional status of subjects based on their cardiac activity. In their study, ECG signals were collected from five different subjects while they watched video clips, and these signals were classified into emotional states of ''happy'' or ''sad.'' A total of 169 features were extracted from each ECG dataset and used as inputs to the neural network (Fig. 8a). To evaluate the model's effectiveness, an independent group of five subjects was presented with four different video clips and asked to classify their emotions based on their physiological responses. The model achieved an F1 score of 0.73, demonstrating the effectiveness of the ML system in classifying emotional states based on ECG signals.

For real-time cardiovascular monitoring, Kim *et al.* developed an ''all-in-one'' stretchable-hybrid electronics (SHE) system that integrates ML to analyze ECG signals and classify both user motion activity and cardiac status.[176,181] The SHE system was constructed using a flexible and stretchable ecoflex (1 : 2) elastomer and a nanomembrane gold electrode, ensuring conformal skin adhesion to the skin (Fig. 8b). The elimination of gaps between the electrode and the skin is crucial for ECG monitoring, as such gaps can lead to motion artifacts caused by changes in resistance and capacitance at the interface. The acquired ECG signals, along with acceleration and angular velocity data, were wirelessly transmitted *via* bluetooth to commercial electronic devices such as smartphones for further processing. The analysis was performed using a CNN-based algorithm named ECGSeq2Seq, which applies a sequence-to-sequence annotation approach to ECG data classification. The architecture consists of convolutional layers followed by batch normalization layers, with residual connections between corresponding convolutional and deconvolutional layers to maintain gradient flow, enhance training convergence, and mitigate overfitting issues. Using this method, the system successfully performed semantic segmentation of ECG data and classified cardiac conditions into categories such as normal, myocardial infarction (MI), heart failure (HF), miscellaneous arrhythmias (AR), supraventricular ectopic beats (SVEB), and ventricular ectopic beats (VEB). Additionally, a similar CNN-based model, ActivityResNet, was employed for motion classification, distinguishing between idle sitting or standing, walking, walking downstairs, walking upstairs, and running (Fig. 8c). Beyond experimental systems, ML has also demonstrated clinical applicability in predicting and optimizing treatment strategies for cardiac patients. For instance, in the intensive care unit (ICU), a light gradient boosting machine (LGBM) algorithm has been successfully employed to predict cardiac arrest within 0.5 to 24 hours with an area under the receiver operating characteristic (AUROC) of 0.889, demonstrating high predictive accuracy.[182] Additionally, in patients with chronic heart failure, 30–50% do not respond to cardiac resynchronization therapy (CRT), posing a significant clinical challenge. ML models have been explored to predict CRT responsiveness, allowing for more efficient patient stratification and timely intervention.[183] These advancements underscore ML's critical role in optimizing treatment decisions, minimizing delays, and ensuring that life-saving therapies are administered within the crucial golden hour.[184]

Recent advancements in daily-use electronic skin (e-skin) technology have further enhanced ECG monitoring capabilities. Cui *et al.* developed a 12-lead ECG device based on an MXene-polyurethane mesh (MPM) design, incorporating both CNN and LSTM models to achieve 99% accuracy in the classification of four types of arrhythmias.[177] The MPM e-skin features ultra-low contact impedance (4.68 kΩ at 1 kHz), a high signal-to-noise ratio (16.5 dB), and excellent breathability (2.1838 kg m^{-2} day^{-1}), attributed to the high electrical conductivity of Ti_3C_2 MXene and the mesh structure of electrospun polyurethane fibers. The hybrid CNN and LSTM algorithm enables real-time, *in situ* arrhythmia monitoring and diagnosis. The 12-lead ECG signals acquired using the MPM e-skin demonstrated signal quality comparable to commercial Ag/AgCl gel electrodes, confirming its suitability for medical-grade ECG signal acquisition

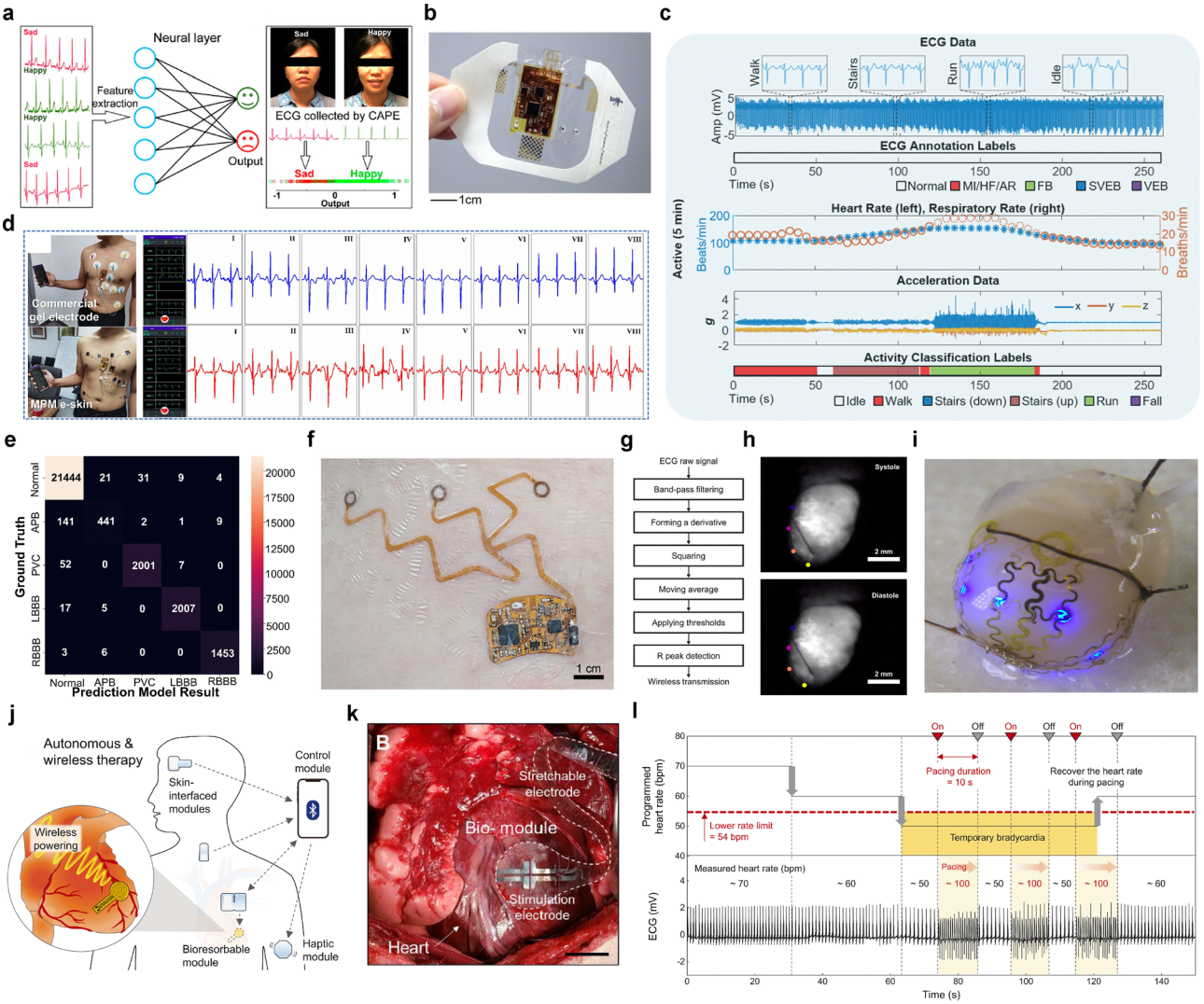


Fig. 8 ML application in cardiovascular signal monitoring. (a) ML methodology of ECG feature extraction to classify emotions. Reproduced with permission from ref. 175. Copyright 2020, American Chemical Society. (b) Photograph of tegaderm-integrated SHE. (c) *In vivo*, real-time, ambulatory monitoring of ECG and motion activity with a SHE on a human subject. Reproduced with permission from ref. 176. Copyright 2019, Wiley-VCH. (d) 12-Lead (eight-channel) ECG signals acquired by the system when using the commercial gel electrodes (blue) and the MPM e-skin electrodes (red). (e) Confusion matrix of the intelligent algorithm model for the four types of arrhythmia diagnosis. Reproduced with permission from ref. 177. Copyright 2023, Elsevier. (f) Photograph of the Kagome metastructure-tethered wireless ECG sensor patch attached to skin. (g) Algorithm developed to detect the *R* peaks of ECG curves for wireless transmission of measured signals. Reproduced with permission from ref. 178. Copyright 2022, Wiley-VCH. (h) Snapshot of heart with motion-tracking markers during systole and diastole phase. (i) Photographic image of array applied to mouse heart. Reproduced with permission from ref. 179. Copyright 2022, American Association for the Advancement of Science. (j) Schematic illustration of a system for autonomous and wireless pacing therapy. (k) Photograph of an canine heart with the stimulation electrode of the bioresorbable module sutured to the ventricular epicardium. (l) Programmed HR (top) and measured ECG (bottom) of a human whole heart. Reproduced with permission from ref. 180. Copyright 2022, American Association for the Advancement of Science.

(Fig. 8d). To enhance clinical applicability, the authors implemented a hybrid CNN-LSTM model that integrates three CNN layers for morphological feature extraction and an LSTM layer for temporal feature analysis. The inclusion of batch normalization and dropout layers prevented overfitting, allowing the downsized model to achieve a prediction accuracy exceeding 99.3% (Fig. 8e). Furthermore, the downsized algorithm was successfully deployed on portable devices such as Raspberry Pi and iPhone 13, demonstrating its potential as a practical, daily-wearable ECG monitoring system.

Other than cardiovascular status monitoring, ML is applied to enhance the accuracy and efficiency of data acquisition in cardiovascular monitoring devices. By improving data collection and processing, ML algorithms contribute to minimizing signal artifacts, optimizing power consumption, and extending device operational longevity. For instance, in armband-based ECG monitoring setups, EMG artifact contamination is a significant challenge due to the proximity of electrodes to the biceps and triceps, leading to higher EMG signal interference. To address this issue, Lázaro *et al.* developed a SVM-based ECG

artifact detection system that classifies 10-second ECG segments as normal or artifact-affected based on nine non-fiducial features, including Shannon entropy and skewness.[185] Once the ECG segments were labeled as artifact, normal, or indeterminate, PCA filtering was applied to attenuate EMG noise in the ECG signals. The first principal component extracted through PCA, which contains the least EMG noise, was selected to generate a synthesized ECG signal, while the last principal component, representing the most significant EMG noise, was discarded. This approach effectively removed EMG contamination, resulting in cleaner ECG signals with improved signal integrity. In addition to artifact removal, ML-driven data compression techniques have been introduced to facilitate efficient data transmission and reduce power consumption in wearable ECG monitoring systems. Hwang *et al.* developed a versatile hybrid e-skin patch incorporating a flexible printed circuit board tethered to the skin through a Kagome metastructure.[178] The e-skin system consists of a polyurethane-based elastomeric film (Tegaderm) forming the top and bottom layers of the Kagome metastructure, with a polyimide-based wireless sensor embedded between them. This design effectively distributes skin strain while maintaining high breathability (14.47 ± 0.30 g m^{-2} h^{-1}), and no significant skin irritation was observed after five days of continuous attachment (Fig. 8f). To optimize data efficiency, an ML-based algorithm was implemented to selectively detect and transmit only the *R* peaks of ECG signals, significantly reducing the amount of data required for processing while still providing accurate heart rate and respiration rate measurements (Fig. 8g). Compared to continuous wireless ECG readouts, the selective transmission of *R* peaks accurately reflected heart rate values, ranging from 68 to 77 beats per minute, demonstrating the effectiveness of this efficient approach.

The integration of ML into cardiovascular monitoring and improved data processing has laid the foundation for its application in autonomous treatment. In the future, ML is expected to play a crucial role not only in diagnosing cardiac conditions but also in actively administering therapeutic interventions. By leveraging real-time physiological data analysis and predictive modeling, ML-driven systems can autonomously deliver targeted stimulation to prevent or mitigate cardiac diseases. To illustrate the potential of ML-driven autonomous treatment, the following studies highlight closed-loop systems that incorporate real-time cardiac monitoring and stimulation algorithms. Ausra *et al.* introduced a wireless, battery-free cardiac device equipped with on-board computational capabilities for autonomous cardiac monitoring and intervention.[147,179] To accommodate the continuous mechanical deformation induced by heart motion, deep learning algorithms were employed to track systolic and diastolic cardiac cycles, thereby informing the mechanical design of the device (Fig. 8h). The strain values derived from these cardiac contractions were utilized to simulate and fabricate a serpentine electronic structure capable of sustaining the required elastic deformations. The resulting customized array successfully conformed to an *ex vivo* mouse heart without structural compromise (Fig. 8i). For real-time pacing under arrhythmic conditions, an embedded algorithm was implemented to autonomously calculate the interval between *R* waves, enabling precise heart rate measurement. The system demonstrated an accuracy of ± 0.35 Hz for normal heartbeats and ± 0.24 Hz for abnormal rhythms, comparable to commercial wireless heart rate monitoring devices, which typically exhibit an accuracy range of ± 0.2 to ± 0.6 Hz. *In vivo* studies further validated the system's ability to wirelessly monitor heart rate and deliver cardiac pacing over a 12-day period, demonstrating its long-term, autonomous cardiac regulation potential. Similarly, Choi *et al.* developed a transient closed-loop system integrating a wireless network of skin-mounted biosensors with a bioresorbable pacemaker for continuous cardiac monitoring and stimulation.[180] The system performed adaptive cardiac pacing based on physiological data, which was wirelessly transmitted to a mobile application *via* bluetooth low energy (BLE) protocol (Fig. 8j). *In vivo* validation using a canine whole-heart model confirmed the system's efficacy in cardiac stimulation (Fig. 8k). A feedback algorithm implemented within the mobile application autonomously determined whether electrical stimulation was required by comparing the measured heart rate to preprogrammed lower and upper rate limits. As a proof of concept, the system was tested on an *ex vivo* human heart model for bradycardia detection and intervention. Upon detecting bradycardia, the system automatically initiated pacing at 100 beats per minute for a predetermined duration. Following the pacing event, the system continuously evaluated the ECG signal to determine whether additional stimulation was necessary. If the heart rate exceeded 60 beats per minute, which is above the bradycardia threshold set at 54 beats per minute, the system ceased further stimulation (Fig. 8l).

3.3 Biochemical signals

Monitoring chemical biomarkers, including proteins and hormones, is essential for health monitoring as it provides real-time insights into physiological and pathological processes. Thus, bioelectronics capable of reliably detecting various chemical biomarkers has been extensively developed, with the ultimate goal of integrating them in our daily lives.[186,187] For better comprehensive health management, innovative platforms for the continuous detection of multiple chemical biomarkers have been recently reported. Especially, metabolic syndromes, such as hypertension and diabetes, arise from complex pathological factors, emphasizing the necessity of monitoring a wide range of chemical biomarkers. In this regard, analyzing multiple biomarkers simultaneously has become an important task to reliably interpret the relationships between biomarkers and physiological states. To address this challenge, researchers have attempted to improve their capability to process and analyze diverse chemical biomarker data with the aid of ML models. By applying appropriate ML algorithms, chemical biomarker sensing platforms can provide more accurate physiological information, enabling personalized health management and preventive care for users.

An example of applying ML algorithms for chemical biomarker detection is predicting the concentration of chemical biomarkers detected by colorimetric sensors. For example, Wang *et al.* presented an AI-wearable microfluidic colorimetric

sensor system (WMC) to monitor key biomarkers in human tears.[188] This sensor employed a polydimethylsiloxane (PDMS)-based flexible microfluidic epidermal patch to collect tears and used a colorimetric reaction to sense Ca^{2+}, vitamin C, H^{+} (pH), and proteins. The extracted RGB values of individual biomarkers were converted into feature signals for deep learning models to predict their concentrations in tears (Fig. 9a). Among six deep learning models tested, the convolutional recurrent

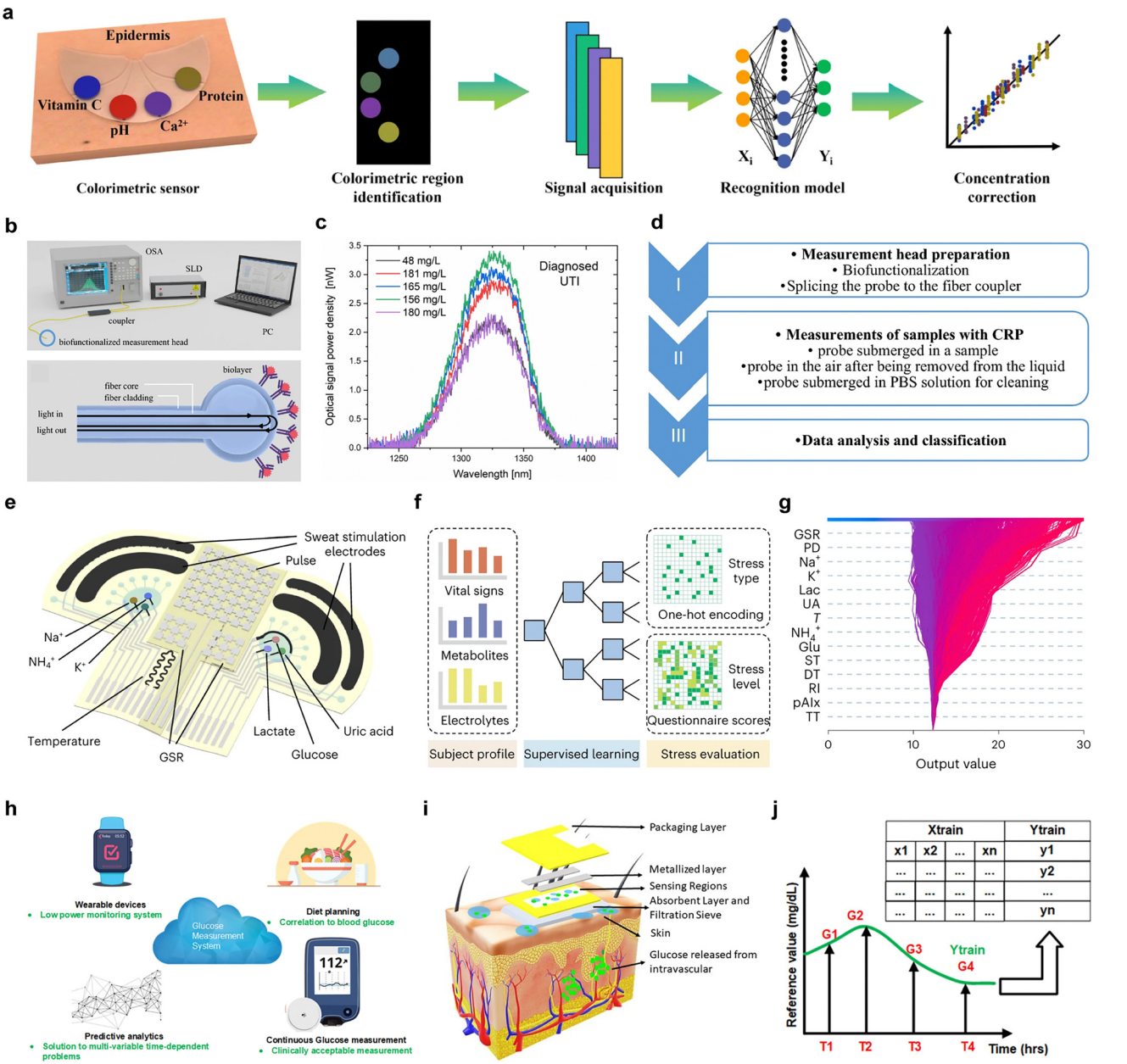


Fig. 9 Biochemical signals. (a) Structure of PDMS-based flexible microfluidic epidermal patch and basic process of the AI-WMCS that proposes the deep-learning artificial intelligence-assisted colorimetric sensing of four biomarkers. Reprinted with permission from ref. 188. Copyright 2024, Springer Nature. (b) Measurement setup of SLD (super luminescent diode), OSA (optical spectrum analyzer) and structure of sensor's measurement head. (c) Representative optical signal from sensors that shows CRP level determined with the use of ELISA for patients diagnosed with UTI. (d) Schematic illustration of experimental workflow. Reprinted with permission from ref. 190. Copyright 2024, Springer Nature. (e) Schematic and optical images of the flexible CARES sensor patch attached to the wrist of human subject with main functionalities (vital sign monitoring, key metabolites, electrolyte detection). (f) ML pipeline for CARES-enabled stressor classification and stress/anxiety level assessment. (g) SHAP decision plot representing how the ML model determines the state anxiety level. Reprinted with permission from ref. 191. Copyright 2024, Springer Nature. (h) Overview of the glucose monitoring system using non-invasive sweat sensor and ML-based model to predict real-time sweat glucose values. (i) Sensing architecture of the sensor that measures glucose concentrations in human sweat using an affinity capture probe-functionalized sensor surface. (j) Representative continuous signal and the conversion of the measured input parameters to glucose concentrations using discrete data points. The glucose concentrations from the sweat were measured by ELISA and used to interpolate with the impedance signal matching with those time points to obtain a smooth and continuous sweat glucose concentration output. Reprinted with permission from ref. 192. Copyright 2022, Springer Nature.

neural network (CNN–GRU) model demonstrated the highest accuracy (lowest loss value) in predicting the concentrations of four biomarkers, making it the most suitable neural network model. Combining CNN and GRU allows for more comprehensive modeling of the input sequence's spatial-temporal relationships while requiring fewer parameters, which makes them more efficient for training and deployment. Incorporating color temperature and pH feature data, training the 1D–CNN–GRU model (for pH) and the 3D–CNN–GRU model (for Ca^{2+}, vitamin C, and proteins) resulted in improved concentration prediction performance for both models, achieving R^2 values exceeding 0.99. These features significantly enhance the practicality of wearable colorimetric biosensors for effective healthcare management. Along with neural network-based deep learning algorithms, various ML models, such as multiple linear regression, decision tree, random forest, and XGBoost, can be applied to predict the concentration of chemical biomarkers from the colorimetric sensors.[189]

Another application of ML is health state monitoring by using classification models. Małgorzata Szczerska's research group utilized an interferometric sensor to detect C-reactive protein (CRP) levels in urine and applied ML algorithms to distinguish between inflammation and non-inflammation states of human subjects.[190] CRP serves as an indicator of inflammation caused by oncological, cardiovascular, bacterial, or viral events, which is conventionally analyzed by enzyme-linked immunosorbent assay (ELISA). The authors demonstrated optical fiber-based CRP sensor with a biofunctionalized tip for CRP bonding, which overcomes the limitations of ELISA analysis including high cost and necessity of expertise (Fig. 9b). The optical signals measured by the sensor indicated a significant increase in urinary CRP levels due to the presence of blood in the urine or urine concentrations effects caused by urosepsis urinary tract infections (Fig. 9c). Using 27 different ML classifiers, 42 human urine samples were classified into normal CRP levels (moderate elevation, $\leq$10 mg L^{-1}) and high CRP levels (marked severe elevation) (Fig. 9d). The XGB Classifier achieved the highest accuracy, with an AUC and F1-score of 100%, demonstrating its potential for pre-screening and early diagnosis. In the third stage of the table in Fig. 9d, XGBoost Classifier model was used in classifying human urine samples based on CRP levels.

Furthermore, ML algorithms offer significant advantages in processing multiple types of chemical biomarker data simultaneously, which is essential to enhance the accuracy of detecting health states. These algorithms can quantitatively or qualitatively classify data signals according to specific criteria or normalize various data types with different scales for accurate prediction. Xu *et al.* presented a non-invasive electronic skin, consolidated artificial-intelligence-reinforced electronic skin (CARES), that is capable of sensing multiple stress-related biomarkers for long-term stress response monitoring (Fig. 9e).[191] CARES monitors three key biophysical signals—pulse waveform, galvanic skin response (GSR), and skin temperature—along with six molecular biomarkers in human sweat: glucose, lactate, uric acid, Na^+, K^+, and NH_4^+. The collected data was then processed using XGBoost algorithms to distinguish between different stressors and quantify stress levels under three experimental conditions: a cold pressor test (CPT), a virtual reality (VR) environment, and exercise (Fig. 9f). A series of calibration and normalization steps were applied to physiological CARES signals from 10 subjects to ensure robust feature extraction. Furthermore, Shapley additive explanation (SHAP) analysis was conducted to evaluate the feature importance of each biomarker for different stressors, identifying that GSR, pulse, Na^+, K^+, NH_4^+, and lactate as the most influential biomarkers in predicting state anxiety levels (Fig. 9g). To assess state anxiety levels, XGBoost regression model was employed, successfully predicting state-trait anxiety inventory form Y (STAI-Y) scores with a confidence level exceeding 98% and a high coefficient of determination. In Fig. 9f, XGBoost model is used in quantifying stress levels among different stressors and predicting the anxiety state score based on the collected data from CARES signals. Likewise, Mondal *et al.* utilized random forest model for DNA detection and classification, showing the best performance among other ML algorithms including SVM, decision tree, multi-layer perceptron (MLP), and logistic regression.[193]

Lastly, ML algorithms have also been employed to convert discrete biomarker data into continuous data, enabling the reliable tracking of real-time dynamics of chemical concentrations. For example, Shalini Prasad's research group predicted real-time continuous sweat glucose values based on intermittently detected sweat glucose data obtained from a non-invasive sweat sensor (Fig. 9h).[192] The sensor measured glucose concentrations in passively secreted human sweat using electrochemical impedance spectroscopy (EIS) and an affinity capture probe-functionalized sensor surface (Fig. 9i). The data obtained from the sensor served as input for decision tree regression. A correlation matrix was employed for data interpolation, and the refined signal data was used as a dependent variable in the ML algorithm. The decision tree model demonstrated optimal performance with an R^2 value of 0.93 and a RMSE value of 0.11, and it was tested using samples from three human subjects (Fig. 9j). Compared with reference values to evaluate sweat glucose progression, the decision tree model, used as a regression algorithm, successfully captured trend changes, though some amplitude errors were observed due to the small dataset size.

As demonstrated in the studies mentioned above, ML algorithms serve numerous functions when integrated with biosensors capable of sensing various chemical biomarkers. In this regard, selecting appropriate models tailored to specific data types and distinct purposes is essential to provide valuable information for health management. Consequently, leveraging ML for chemical signal collection and processing facilitates efficient biomarker detection and enables the prevention and management of diseases, making it an innovative tool in future healthcare.

3.4 Other biosignals and multimodal machine learning

While traditional biosignals such as electrophysiological signals and metabolic markers have been extensively studied, recent advances in wearable technologies and ML have enabled the exploration of other dynamic physiological signals.[194–196] These include biomechanical deformations, cutaneous resistance

changes, laryngeal vibrations, and multisensory environmental interactions, offering new frontiers in health monitoring, human–machine interfaces, and rehabilitation technologies.[197–200]

Che *et al.* introduced a wearable sensing-actuation system designed to assist speech without relying on vocal folds (Fig. 10a).[201] This system leverages soft magnetoelastic materials to capture extrinsic laryngeal muscle movements, converting them into high-fidelity electrical signals. The sensing module captures biomechanical signals from the throat, which are classified using an SVM-based algorithm, as discussed in Section 2. This classification enables the generation of synthetic voice signals through the actuation module, circumventing the need for vocal fold vibrations. The confusion matrices included in the figure showcase the high accuracy of voice recognition, with validation and testing accuracies of 98% and 96.5%, respectively. More recently, Kim *et al.* proposed a silent speech interface (SSI) utilizing biosignals derived from facial strain data captured through ultrathin crystalline-silicon-based strain gauges.[202] The captured biosignals are processed using a 3D convolutional deep learning algorithm, which effectively encodes both spatial and temporal features of the strain data. This approach, which employs the same CNN model introduced in Section 2, enables the classification of an extensive word set, achieving an average recognition accuracy of 87.53%.

Transitioning to different types of biosignals, Moin *et al.* reported a wearable biosensing system that utilizes surface electromyography (sEMG) signals for real-time hand gesture recognition (Fig. 10b).[203] The system employs an adaptive ML

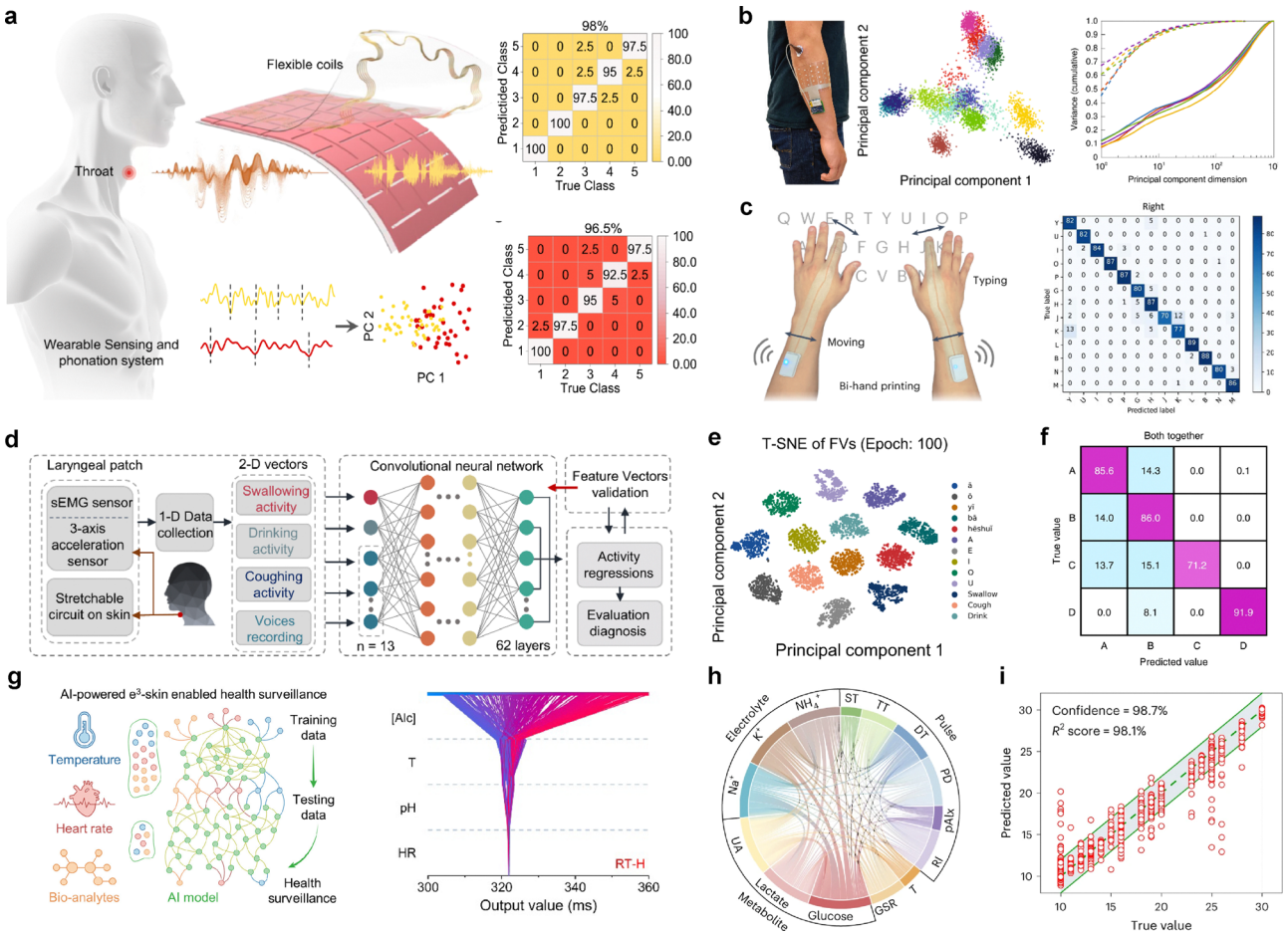


Fig. 10 Processing of other biosignals and multimodal biosignals through ML applications in wearable health monitoring systems. (a) Wearable sensing-actuation system capturing laryngeal muscle movements for speech assistance, classified using an SVM-based algorithm. Reprinted with permission from ref. 201. Copyright 2024, Springer Nature. (b) sEMG-based hand gesture recognition system utilizing hyperdimensional computing for adaptive classification. Reprinted with permission from ref. 203. Copyright 2021, Springer Nature. (c) Nanomesh receptor for proprioceptive biosignal monitoring with time-dependent contrastive learning. Reprinted with permission from ref. 204. Copyright 2023, Springer Nature. (d) Stretchable throat-monitoring device integrating sEMG and triaxial acceleration, processed *via* CNN for activity recognition. (e) t-SNE visualization of feature vectors for phonation-related activity classification. Reprinted with permission from ref. 205. Copyright 2023, Springer Nature. (f) Confusion matrix demonstrating classification performance of a multimodal biosignal-based phonation system. Reprinted with permission from ref. 206. Copyright 2022, Springer Nature. (g) AI-powered e^3-skin integrating biochemical and physiological sensors for predictive health surveillance. Reprinted with permission from ref. 86. Copyright 2023, The American Association for the Advancement of Science. (h) Chord diagram illustrating correlations between physiological and biochemical biosignals in stress monitoring. (i) Regression analysis of predicted *vs.* true stress-related physiological responses, achieving a confidence level of 98.7% and an R^2 score of 98.1%. Reprinted with permission from ref. 191. Copyright 2024, Springer Nature.

framework based on hyperdimensional (HD) computing, enabling in-sensor model training and updates. Although HD computing is not among the machine learning approaches introduced in Section 2—since it falls outside the scope of conventional models such as kernel-based, distance-based, or tree-based classifiers—it represents a distinct neuro-inspired learning paradigm that mimics the high-dimensional representations of information observed in the human brain. In HD computing, data are encoded as ultra-high-dimensional binary or bipolar vectors—typically with 10 000 or more dimensions—and learning is performed through lightweight vector operations such as binding, bundling, and similarity comparison. This approach enables fast, memory-efficient, and noise-resilient learning, making it well suited for real-time, adaptive applications in wearable devices. The system achieved an accuracy of 97.12% for 13 hand gestures and maintained high performance even when expanded to 21 gestures. Similarly, Kim *et al.* introduced a substrate-less nanomesh receptor designed to capture proprioceptive signals through skin stretch-induced resistance changes (Fig. 10c).[204] The biosignals are processed using an unsupervised meta-learning framework with time-dependent contrastive learning, facilitating user-independent, data-efficient recognition of diverse hand tasks. Although TD-C learning is not covered in Section 2—since it falls outside the scope of conventional machine learning classifiers such as kernel-based or tree-based models—it is a form of self-supervised learning that leverages temporal continuity to extract informative feature representations from unlabelled data. This system demonstrates rapid adaptation to new users and tasks, achieving over 80% accuracy within just 20 training epochs.

There is growing interest in multimodal biosignal analysis aimed at capturing diverse physiological information, enabling more robust and accurate insights through ML-driven integration. Xu *et al.* reported a fully integrated, standalone stretchable device platform for wireless monitoring and ML-based processing of diverse vibrations and muscle activities from the throat (Fig. 10d).[205] The system captures both sEMG signals and triaxial acceleration data, reflecting muscle electrical activity and inertial vibrations, respectively. These biosignals are processed using a 2D-like sequential feature extractor based on a CNN, consistent with the neural network-based machine learning approaches introduced in Section 2, enabling classification of various actions such as swallowing, talking, drinking, and coughing. The model achieves a high classification accuracy of 98.2% for 13 distinct states and maintains robust performance with a 92% accuracy even when applied to new subjects. Fig. 10e illustrates the clustering of feature vectors using t-SNE, highlighting the model's capacity to differentiate complex physiological signals for real-time health monitoring and rehabilitation evaluation. Similarly, Kim *et al.* introduced a conformable sensory face mask (cMaSK) designed to decode both biological and environmental signals (Fig. 10f).[206] This system integrates multimodal sensors to capture data such as skin temperature, humidity, breathing patterns, and verbal activities. The collected biosignals are processed using a *k*-means clustering algorithm, an unsupervised ML method as described in Section 2, to classify mask positions and assess fit quality. The confusion matrix in Fig. 10f highlights the model's ability to accurately differentiate mask positions. This approach allows for real-time monitoring of mask usage, providing feedback on fit effectiveness with a classification accuracy of up to 92.8%.

Broadening the boundaries of multimodal biosignal integration by encompassing a broader spectrum of physiological and biochemical indicators, Song *et al.* introduced a 3D-printed epifluidic elastic electronic skin (e^3-skin) designed for ML-powered multimodal health surveillance (Fig. 10g).[86] The e^3-skin integrates diverse physiochemical sensors capable of monitoring temperature, heart rate, and sweat bio-analytes such as alcohol, pH, and glucose. The collected biosignals are processed using an AI-powered model that leverages supervised ML algorithms to analyze and interpret complex physiological data. The model employs feature extraction techniques to convert raw sensor data into meaningful health indicators, followed by classification and regression algorithms to predict health-related outcomes. Specifically, the system utilizes ridge regression, a regularized linear model also introduced in Section 2, to predict behavioral impairments such as reaction time and inhibitory control. More recently, Xu *et al.* introduced an AI-powered electronic skin (CARES) for continuous stress response monitoring, incorporating multimodal sensing of physiological and biochemical signals (Fig. 10h and i).[191] The CARES system simultaneously tracks three vital signs—pulse waveform, galvanic skin response (GSR), and skin temperature—alongside six molecular biomarkers in sweat, including glucose, lactate, uric acid, sodium, potassium, and ammonium. By leveraging a supervised ML pipeline, the system classifies stressors with an accuracy of 98.0% and predicts psychological stress responses with a confidence level of 98.7%. The model integrates time-series physiological and biochemical data, employing gradient-boosted decision trees (XGBoost) for robust classification and regression. Feature importance analysis using Shapley additive explanations (SHAP) reveals that GSR, pulse, and electrolyte levels contribute significantly to stress differentiation, demonstrating the model's ability to capture complex stress-related physiological interactions. The chord diagram in Fig. 10h highlights the interconnections among different biomarkers, while the regression performance in Fig. 10i confirms the strong predictive capacity of the AI model in estimating stress-related metrics.

4. Conclusions

Wearable sensors have evolved rapidly in recent years with the development of flexible and stretchable materials and the increased utilization of biocompatible materials, enabling non-invasive, continuous, real-time monitoring of biosignals.[207–211] ML is supporting these hardware advances to improve the precision and efficiency of analyzing a wide spectrum of biosignals, dramatically expanding the capabilities of wearable and implantable devices.

ML models play a pivotal role in biosignal analysis by performing several key functions. In this review, we introduced ML algorithms that can be applied depending on the biosignal

characteristics and the purpose of processing the data. Recent studies have demonstrated the implementation of ML methodologies on biosignals acquired by bioelectronics for data preprocessing, clustering, regression, and classification. For example, data preprocessing is used to remove noise and refine the data so that it can be effectively applied to ML models. Clustering techniques can be utilized to identify patterns in the signals and categorize them into meaningful groups, and regression and classification models can be applied to predict the presence of certain diseases or detect anomalies. Furthermore, recent advancements have enabled the analysis of the correlation between various vital signs, including heart rate, BP, and oxygen saturation. This development stems from the integration of sophisticated signal processing techniques with optimized ML models, resulting in enhanced precision and efficiency in the interpretation of vital signs. Building on this trend, recent advances in machine learning have led to the emergence of architectures such as generative adversarial networks (GANs), transformers, and spiking neural networks (SNNs), which are increasingly being explored in biosignal analysis for tasks such as data augmentation, long-range temporal modeling, and energy-efficient computation.

ML is particularly effective in processing large-scale multimodal signals, enabling the integrated analysis of neural signals, cardiovascular signals, and biochemical markers. This capability allows for a comprehensive assessment of overall health status and facilitates more precise monitoring of disease progression. Furthermore, ML can be utilized for disease prevention by identifying the pre-disease state and detecting transitional phases leading to disease onset. These advancements contribute to the development of automated healthcare systems that enhance the detection of chronic diseases and enable early diagnosis.

Beyond signal acquiring insightful information about health states, the ability to measure diverse biosignals serves as a foundation for therapeutic applications, constructing effective treatment strategies informed by continuous physiological data. Numerous studies have focused on developing advanced technologies assisted by ML including adaptive therapy systems that adjust treatment strategies based on real-time physiological signals, closed-loop biofeedback mechanisms that autonomously regulate therapy, and AI-assisted diagnostic technologies for early disease detection and clinical decision support.[212–214] By maximizing the analytical capabilities of wearable and implantable devices, ML will not only facilitate personalized healthcare but also contribute to large-scale public health monitoring and advancements in precision medicine. In the future, the role of ML-driven bioelectronics will continue to expand, accelerating the development of next-generation digital healthcare solutions.

Author contributions

I. J., W. G. C., E. K., W. P., H. S., J. L., M. O., E. K., J. P., T. L., D. K., S. H. A., S. K., contributed equally to this work, and H. C. contributed to this study. J.-U. Park supervised, reviewed, and edited the manuscript. CRediT: Inhea Jeong conceptualization, investigation, visualization, writing – original draft, writing – review & editing; Won Gi Chung conceptualization, investigation, visualization, writing – original draft, writing – review & editing; Enji Kim conceptualization, writing – review & editing; Wonjung Park conceptualization, investigation, writing – original draft, writing – review; Hayoung Song conceptualization, investigation, visualization, writing – original draft, writing – review; Jakyoung Lee conceptualization, investigation, visualization, writing – original draft; Myoungjae Oh conceptualization, investigation, writing – original draft; Eunmin Kim conceptualization, investigation, visualization, writing – original draft, writing – review; Joonho Paek conceptualization, investigation, visualization, writing – original draft, writing – review; Taekyeong Lee conceptualization, investigation, visualization, writing – original draft; Dayeon Kim conceptualization, investigation, visualization, writing – original draft; Seung Hyun An conceptualization, investigation, writing – original draft; Sumin Kim conceptualization, investigation, writing – original draft; Hyunjoo Cho investigation, writing – original draft; Jang-Ung Park supervision, review & editing.

Data availability

No new data were generated or analyzed, and no primary research results, software, or code are included in this review.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was supported by the Ministry of Science & ICT (MSIT) through the ERC Program (RS-2024-00406240), and by the National Research Foundation (2023R1A2C2006257), the STEAM Research Programs (RS-2024-00460364), and the Sejong Science Fellowship (RS-2025-00514998). The authors also thank the Institute for Basic Science (IBS-R026-D1) and the Korea Institute of Science and Technology (KIST) Institutional Program (2E33191 and 2E33190) for financial support.

References

1 J. Kim, A. S. Campbell, B. E.-F. de Ávila and J. Wang, *Nat. Biotechnol.*, 2019, **37**, 389–406.
2 M. Lin, H. Hu, S. Zhou and S. Xu, *Nat. Rev. Mater.*, 2022, **7**, 850–869.
3 Y. W. Kwon, E. Kim, C. S. Koh, Y.-G. Park, Y.-M. Hong, S. Lee, J. Lee, T. J. Kim, W. Mun, S. H. Min, S. Kim, J. A. Lim, H. H. Jung and J.-U. Park, *ACS Nano*, 2025, **19**, 7337–7349.
4 S. Xu, J. Kim, J. R. Walter, R. Ghaffari and J. A. Rogers, *Sci. Transl. Med.*, 2022, **14**, eabn6036.
5 E. Song, J. Li, S. M. Won, W. Bai and J. A. Rogers, *Nat. Mater.*, 2020, **19**, 590–603.
6 W. G. Chung, E. Kim, Y. W. Kwon, J. Lee, S. Lee, I. Jeong and J.-U. Park, *Adv. Funct. Mater.*, 2024, **34**, 307990.

7 Y. H. Cho, Y.-G. Park, S. Kim and J.-U. Park, *Adv. Mater.*, 2021, **33**, 2005805.
8 W. Park, H. Seo, J. Kim, Y.-M. Hong, H. Song, B. J. Joo, S. Kim, E. Kim, C.-G. Yae, J. Kim, J. Jin, J. Kim, Y. Lee, J. Kim, H. K. Kim and J.-U. Park, *Nat. Commun.*, 2024, **15**, 2828.
9 J. Jang, B. G. Hyun, S. Ji, E. Cho, B. W. An, W. H. Cheong and J.-U. Park, *NPG Asia Mater.*, 2017, **9**, e432.
10 H. S. An, Y.-G. Park, K. Kim, Y. S. Nam, M. H. Song and J.-U. Park, *Adv. Sci.*, 2019, **6**, 1901603.
11 B. Oh, Y.-G. Park, H. Jung, S. Ji, W. H. Cheong, J. Cheon, W. Lee and J.-U. Park, *Soft Rob.*, 2020, **7**, 564–573.
12 B. W. An, K. Kim, H. Lee, S.-Y. Kim, Y. Shim, D.-Y. Lee, J. Y. Song and J.-U. Park, *Adv. Mater.*, 2015, **27**, 4322–4328.
13 S. Lee, W. G. Chung, H. Jeong, G. Cui, E. Kim, J. A. Lim, H. Seo, Y. W. Kwon, S. H. Byeon, J. Lee and J.-U. Park, *Adv. Mater.*, 2024, **36**, 2404428.
14 Y. W. Kwon, Y. S. Jun, Y.-G. Park, J. Jang and J.-U. Park, *Nano Res.*, 2021, **14**, 3070–3095.
15 J. Park, J. Kim, S.-Y. Kim, W. H. Cheong, J. Jang, Y.-G. Park, K. Na, Y.-T. Kim, J. H. Heo, C. Y. Lee, J. H. Lee, F. Bien and J.-U. Park, *Sci. Adv.*, 2018, **4**, eaap9841.
16 Y. Wang, H. Haick, S. Guo, C. Wang, S. Lee, T. Yokota and T. Someya, *Chem. Soc. Rev.*, 2022, **51**, 3759–3793.
17 B. Park, J. H. Shin, J. Ok, S. Park, W. Jung, C. Jeong, S. Choy, Y. J. Jo and T. Kim, *Science*, 2022, **376**, 624–629.
18 S. Kim, Y. W. Kwon, H. Seo, W. G. Chung, E. Kim, W. Park, H. Song, D. H. Lee, J. Lee, S. Lee, K. Lim, I. Jeong, D.-Y. Jo and J.-U. Park, *ACS Appl. Electron. Mater.*, 2023, **5**, 1926–1946.
19 A. A. Salah, *Machine Learning: Concepts, Methodologies, Tools and Applications*, IGI Global Scientific Publishing, 2012, pp. 704–723.
20 A. Maćkiewicz and W. Ratajczak, *Comput. Geosci.*, 1993, **19**, 303–342.
21 H. Abdi and L. J. Williams, *Wiley Interdiscip. Rev.:Comput. Stat.*, 2010, **2**, 433–459.
22 S. Zhao, B. Zhang, J. Yang, J. Zhou and Y. Xu, *Nat. Rev. Methods Primers*, 2024, **4**, 70.
23 Y.-C. Yeh, W.-J. Wang and C. W. Chiou, *Measurement*, 2009, **42**, 778–789.
24 S. Makeig, A. J. Bell, T. P. Jung and T. J. Sejnowski, *Adv. Neural Inf. Process. Syst.*, 1996, **8**, 145–151.
25 . A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*, Springer Science & Business Media, 1st edn, 2007.
26 S. Zhao, X. Tang, W. Tian, S. Partarrieu, R. Liu, H. Shen, J. Lee, S. Guo, Z. Lin and J. Liu, *Nat. Neurosci.*, 2023, **26**, 696–710.
27 L. L. Guo, S. R. Pfohl, J. Fries, A. E. W. Johnson, J. Posada, C. Aftandilian, N. Shah and L. Sung, *Sci. Rep.*, 2022, **12**, 2726.
28 A. Apicella, F. Isgrò, A. Pollastro and R. Prevete, *Eng. Appl. Artif. Intell.*, 2023, **123**, 106205.
29 O. Kilim, A. Olar, T. Joó, T. Palicz, P. Pollner and I. Csabai, *Sci. Rep.*, 2022, **12**, 21302.
30 R. Cuevas-Diaz Duran, H. Wei and J. Wu, *BMC Genomics*, 2024, **25**, 444.
31 D. Singh and B. Singh, *Appl. Soft Comput.*, 2020, **97**, 105524.
32 Z. M. Hira and D. F. Gillies, *Adv. Bioinf.*, 2015, **2015**, 198363.
33 S. Krishnan and Y. Athavale, *Biomed. Signal Process. Control*, 2018, **43**, 41–63.
34 I. Pratama, A. E. Permanasari, I. Ardiyanto and R. Indrayani, in 2016 International Conference on Information Technology Systems and Innovation (ICITSI), 2016, pp. 1–6.
35 A. Sánchez-Morales, J.-L. Sancho-Gómez, J.-A. Martínez-García and A. R. Figueiras-Vidal, *Neural Comput. Appl.*, 2020, **32**, 13233–13244.
36 Y. Jia, H. Pei, J. Liang, Y. Zhou, Y. Yang, Y. Cui and M. Xiang, *Bioengineering*, 2024, **11**, 1109.
37 J. A. van Alsté, W. van Eck and O. E. Herrmann, *Comput. Biomed. Res.*, 1986, **19**, 417–427.
38 X. Wan, H. Wu, F. Qiao, F. Li, Y. Li, Y. Yan and J. Wei, *Comput. Math. Methods Med.*, 2019, **2019**, 7196156.
39 C. Pravin and V. Ojha, 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020, pp. 1094–1100.
40 L. van der Maaten, E. Postma and H. Herik, *J. Mach. Learn. Res.*, 2007, **10**, 2.
41 F. Kamiran and T. Calders, *Knowl. Inf. Syst.*, 2012, **33**, 1–33.
42 Y. Tang, D. Chen and X. Li, *ACM Comput. Surv.*, 2021, **54**, 87:1–87:36.
43 H. S. Obaid, S. A. Dheyab and S. S. Sabry, 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON), Jaipur, India, 2019, pp. 279–283.
44 X. Huang, L. Wu and Y. Ye, *Int. J. Pattern Recognit. Artif. Intell.*, 2019, **33**, 1950017.
45 W. Klonowski, *Nonlinear Biomed. Phys.*, 2009, **3**, 2.
46 M. Arvaneh, C. Guan, K. K. Ang and C. Quek, *IEEE Trans. Neural Networks Learn. Syst.*, 2013, **24**, 610–619.
47 C. Wang, Y. Li, L. Wang, S. Liu and S. Yang, *Neurosci. Lett.*, 2023, **809**, 137306.
48 S. Saha, K. I. Ahmed, R. Mostafa, A. H. Khandoker and L. Hadjileontiadis, *Healthcare Technol. Lett.*, 2017, **4**, 39–43.
49 J. M. Alyea, B. E. Dixon, J. Bowie and A. S. Kanter, in *Health Information Exchange*, ed. B. E. Dixon, Academic Press, 2016, pp. 137–148.
50 H. Benhar, A. Idri and J. L. Fernández-Alemán, *Comput. Methods Programs Biomed.*, 2020, **195**, 105635.
51 F. I. Mowbray, S. M. Fox-Wasylyshyn and M. M. El-Masri, *Can. J. Nurs. Res.*, 2019, **51**, 31–37.
52 J. Pan, Y. Zhuang and S. Fong, in *Soft Computing in Data Science*, ed. M. W. Berry, A. H. Mohamed and B. W. Yap, Springer, Singapore, 2016, pp. 72–88.
53 G. Dougherty, *Pattern Recognition and Classification: An Introduction*, Springer Science & Business Media, 2012.
54 A. K. Singh and S. Krishnan, *Biomed. Eng. Online*, 2023, **22**, 22.
55 R. Nussinov, B. R. Yavuz, H. C. Demirel, M. K. Arici, H. Jang and N. Tuncbag, *Front. Cell Dev. Biol.*, 2024, **12**, 1376639.
56 S. Uddin and H. Lu, *Sci. Rep.*, 2024, **14**, 1670.
57 K. Naeini, F. Sarhaddi, I. Azimi, P. Liljeberg, N. Dutt and A. M. Rahmani, *ACM Trans. Comput. Healthcare*, 2023, **4**, 24:1–24:22.

58 S. Ullah, N. Ullah, M. F. Siddique, Z. Ahmad and J.-M. Kim, *Appl. Sci.*, 2024, **14**, 10339.
59 V. V. Moca, H. Bârzan, A. Nagy-Dăbâcan and R. C. Mureş an, *Nat. Commun.*, 2021, **12**, 337.
60 H. Lee and M. Shin, *Sensors*, 2021, **21**, 4331.
61 Y. Saeys, I. Inza and P. Larrañaga, *Bioinformatics*, 2007, **23**, 2507–2517.
62 Z. Noroozi, A. Orooji and L. Erfannia, *Sci. Rep.*, 2023, **13**, 22588.
63 A. Bommert, X. Sun, B. Bischl, J. Rahnenführer and M. Lang, *Comput. Stat. Data Anal.*, 2020, **143**, 106839.
64 C.-W. Chen, Y.-H. Tsai, F.-R. Chang and W.-C. Lin, *Expert Syst.*, 2020, **37**, e12553.
65 K. Tadist, S. Najah, N. S. Nikolov, F. Mrabti and A. Zahi, *J. Big Data*, 2019, **6**, 79.
66 N. Sánchez-Maroño, A. Alonso-Betanzos and M. Tombilla-Sanromán, in *Intelligent Data Engineering and Automated Learning – IDEAL 2007*, ed. H. Yin, P. Tino, E. Corchado, W. Byrne and X. Yao, Springer, Berlin, Heidelberg, 2007, **vol. 4881**, pp. 178–187.
67 N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr and J. M. O'Sullivan, *Front. Bioinf.*, 2022, **2**, 927312.
68 Y. Luo, *Briefings Bioinf.*, 2022, **23**, bbab489.
69 A. C. Acock, *J. Marriage Fam.*, 2005, **67**, 1012–1028.
70 O. Toka and M. Çetin, *Gazi Univ. J. Sci.*, 2016, **29**, 799–809.
71 S. Sinharay, H. S. Stern and D. Russell, *Psychol. Methods*, 2001, **6**, 317–329.
72 M. Pampaka, G. Hutcheson and J. Williams, *Int. J. Res. Method Educ.*, 2016, **39**, 19–37.
73 R. J. A. Little, *J. Am. Stat. Assoc.*, 1988, **83**, 1198–1202.
74 E.-L. Silva-Ramírez, R. Pino-Mejías, M. López-Coello and M.-D. Cubiles-de-la-Vega, *Neural Networks*, 2011, **24**, 121–129.
75 A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding and C.-T. Lin, *Neurocomputing*, 2017, **267**, 664–681.
76 A. K. Jain, M. N. Murty and P. J. Flynn, *ACM Comput. Surv.*, 1999, **31**, 264–323.
77 S. Ghosh and S. K. Dubey, *Int. J. Adv. Comput. Sci. Appl.*, 2013, **4**, 35–39.
78 M. Ester, H.-P. Kriegel, J. Sander and X. Xu, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, Portland, Oregon, 1996, pp. 226–231.
79 S. E. Schaeffer, *Comput. Sci. Rev.*, 2007, **1**, 27–64.
80 C. Fraley and A. E. Raftery, *J. Am. Stat. Assoc.*, 2002, **97**, 611–631.
81 M. Caron, P. Bojanowski, A. Joulin and M. Douze, in *Deep Clustering for Unsupervised Learning of Visual Features*, ed. V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss, Springer International Publishing, Cham, 2018, vol. 11218, pp. 139–156.
82 K.-L. Du, *Neural Networks*, 2010, **23**, 89–107.
83 R. Ortiz, B. Kluwe, J. B. Odei, J. B. Echouffo Tcheugui, M. Sims, R. R. Kalyani, A. G. Bertoni, S. H. Golden and J. J. Joseph, *Psychoneuroendocrinology*, 2019, **103**, 25–32.
84 M. Lin, Z. Zhang, X. Gao, Y. Bian, R. S. Wu, G. Park, Z. Lou, Z. Zhang, X. Xu, X. Chen, A. Kang, X. Yang, W. Yue, L. Yin, C. Wang, B. Qi, S. Zhou, H. Hu, H. Huang, M. Li, Y. Gu, J. Mu, A. Yang, A. Yaghi, Y. Chen, Y. Lei, C. Lu, R. Wang, J. Wang, S. Xiang, E. B. Kistler, N. Vasconcelos and S. Xu, *Nat. Biotechnol.*, 2024, **42**, 448–457.
85 S. Baik, J. Lee, E. J. Jeon, B. Park, D. W. Kim, J. H. Song, H. J. Lee, S. Y. Han, S.-W. Cho and C. Pang, *Sci. Adv.*, 2021, **7**, eabf5695.
86 Y. Song, R. Y. Tay, J. Li, C. Xu, J. Min, E. Shirzaei Sani, G. Kim, W. Heng, I. Kim and W. Gao, *Sci. Adv.*, 2023, **9**, eadi6492.
87 M. Krzywinski and N. Altman, *Nat. Methods*, 2017, **14**, 757–758.
88 T. Lee, H.-T. Lee, J. Hong, S. Roh, D. Y. Cheong, K. Lee, Y. Choi, Y. Hong, H.-J. Hwang and G. Lee, *Anal. Methods*, 2022, **14**, 4749–4755.
89 L. Wang, W. Zhou, Y. Xing and X. Zhou, *J. Healthcare Eng.*, 2018, **2018**, 7804243.
90 M. S. Haleem, O. Cisuelo, M. Andellini, R. Castaldo, M. Angelini, M. Ritrovato, R. Schiaffini, M. Franzese and L. Pecchia, *Biomed. Signal Process. Control*, 2024, **92**, 106065.
91 Y. Kim, J. Kim, M. Son, J. Lee, I. Yeo, K. Y. Choi, H. Kim, B. C. Kim, K. H. Lee and Y. Kim, *Sci. Rep.*, 2022, **12**, 1282.
92 A. Apicella, P. Arpaia, M. Frosolone, G. Improta, N. Moccaldi and A. Pollastro, *Sci. Rep.*, 2022, **12**, 5857.
93 K. Görür, M. R. Bozkurt, M. S. Bascıl and F. Temurtas, *Acad. Platform – J. Eng. Sci.*, 2021, **9**, 112–125.
94 J. R. Quinlan, *Mach. Learn.*, 1986, **1**, 81–106.
95 Z. Yaari, Y. Yang, E. Apfelbaum, C. Cupo, A. H. Settle, Q. Cullen, W. Cai, K. L. Roche, D. A. Levine, M. Fleisher, L. Ramanathan, M. Zheng, A. Jagota and D. A. Heller, *Sci. Adv.*, 2021, **7**, eabj0852.
96 T. M. Ingolfsson, S. Benatti, X. Wang, A. Bernini, P. Ducouret, P. Ryvlin, S. Beniczky, L. Benini and A. Cossettini, *Sci. Rep.*, 2024, **14**, 2980.
97 X. Xiao, J. Yin, J. Xu, T. Tat and J. Chen, *ACS Nano*, 2024, **18**, 22734–22751.
98 A. Site, J. Nurmi and E. S. Lohan, *IEEE Access*, 2021, **9**, 112221–112235.
99 Y.-H. Nho, J. G. Lim and D.-S. Kwon, *IEEE Access*, 2020, **8**, 40389–40401.
100 Z. Wang, M. Jiang, Y. Hu and H. Li, *IEEE Trans. Inf. Technol. Biomed.*, 2012, **16**, 691–699.
101 R. A. Haraty, M. Dimishkieh and M. Masud, *Int. J. Distrib. Sens. Networks*, 2015, **11**, 615740.
102 K. Tao, J. Li, J. Li, W. Shan, H. Yan and Y. Lu, *Front. Physiol.*, 2021, **12**, 742754.
103 M. Syafrudin, G. Alfian, N. L. Fitriyani, I. Fahrurrozi, M. Anshari and J. Rhee, in 2022 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), 2022, pp. 295–299.
104 J. Kim, M. Kim, M.-S. Lee, K. Kim, S. Ji, Y.-T. Kim, J. Park, K. Na, K.-H. Bae, H. Kyun Kim, F. Bien, C. Young Lee and J.-U. Park, *Nat. Commun.*, 2017, **8**, 14997.
105 K. Lim, J. Lee, S. Kim, M. Oh, C. S. Koh, H. Seo, Y.-M. Hong, W. G. Chung, J. Jang, J. A. Lim, H. H. Jung and J.-U. Park, *Nat. Commun.*, 2024, **15**, 7147.
106 M. A. Myszczynska, P. N. Ojamies, A. M. B. Lacoste, D. Neil, A. Saffari, R. Mead, G. M. Hautbergue, J. D. Holbrook and L. Ferraiuolo, *Nat. Rev. Neurol.*, 2020, **16**, 440–456.

107 J. G. Greener, S. M. Kandathil, L. Moffat and D. T. Jones, *Nat. Rev. Mol. Cell Biol.*, 2022, **23**, 40–55.
108 N. Altman and M. Krzywinski, *Nat. Methods*, 2015, **12**, 999–1000.
109 D. Maulud and A. M. Abdulazeez, *J. Appl. Sci. Technol. Trends*, 2020, **1**, 140–147.
110 G. James, D. Witten, T. Hastie and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, Springer US, New York, NY, 2021.
111 A. E. Hoerl and R. W. Kennard, *Technometrics*, 1970, **12**, 55–67.
112 N. Altman and M. Krzywinski, *Nat. Methods*, 2017, **14**, 933–934.
113 L. Breiman, *Mach. Learn.*, 2001, **45**, 5–32.
114 O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed and H. Arshad, *Heliyon*, 2018, **4**, e00938.
115 Y. LeCun, Y. Bengio and G. Hinton, *Nature*, 2015, **521**, 436–444.
116 H. Seo, Y.-M. Hong, W. G. Chung, W. Park, J. Lee, H. K. Kim, S. H. Byeon, D. W. Kim and J.-U. Park, *Sci. Adv.*, 2024, **10**, eadk7805.
117 B. W. An, E.-J. Gwak, K. Kim, Y.-C. Kim, J. Jang, J.-Y. Kim and J.-U. Park, *Nano Lett.*, 2016, **16**, 471–478.
118 S. Dreiseitl and L. Ohno-Machado, *J. Biomed. Inform.*, 2002, **35**, 352–359.
119 E. W. Steyerberg, F. E. Harrell, G. J. J. M. Borsboom, M. J. C. Eijkemans, Y. Vergouwe and J. D. F. Habbema, *J. Clin. Epidemiol.*, 2001, **54**, 774–781.
120 P. Johnson, L. Vandewater, W. Wilson, P. Maruff, G. Savage, P. Graham, L. S. Macaulay, K. A. Ellis, C. Szoeke, R. N. Martins, C. C. Rowe, C. L. Masters, D. Ames and P. Zhang, *BMC Bioinf.*, 2014, **15**, S11.
121 C. Cortes and V. Vapnik, *Mach. Learn.*, 1995, **20**, 273–297.
122 T. Cover and P. Hart, *IEEE Trans. Inf. Theory*, 1967, **13**, 21–27.
123 J. H. Friedman, *Ann. Stat.*, 2001, **29**, 1189–1232.
124 J. Tohka and M. van Gils, *Comput. Biol. Med.*, 2021, **132**, 104324.
125 R. Angeline, S. N. Kanna, N. G. Menon and B. Ashwath, in *Artificial Intelligence in Healthcare*, ed. L. Garg, S. Basterrech, C. Banerjee and T. K. Sharma, Springer, Singapore, 2022, pp. 35–46.
126 J. Demšar, *J. Mach. Learn. Res.*, 2006, **7**, 1–30.
127 W. Zhu, N. Zeng and N. Wang, NESUG proceedings: Health Care and Life Sciences, 2010, vol. 19, p. 67.
128 R. Kumar and A. Indrayan, *Indian Pediatr.*, 2011, **48**, 277–287.
129 O. A. Debats, G. J. S. Litjens and H. J. Huisman, *PeerJ*, 2019, **7**, e8052.
130 G. Santafe, I. Inza and J. A. Lozano, *Artif. Intell. Rev.*, 2015, **44**, 467–508.
131 A. V. Tatachar, *Int. Res. J. Eng. Technol.*, 2021, **8**, 853–860.
132 M. Mittlböck, *Comput. Methods Programs Biomed.*, 2002, **68**, 205–214.
133 E. Amigó, J. Gonzalo, J. Artiles and F. Verdejo, *Inf. Retr.*, 2009, **12**, 461–486.
134 S. Kapil and M. Chawla, in 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), 2016, pp. 1–4.
135 P. Barbiero, G. Squillero and A. Tonda, *arXiv*, 2020, preprint, arXiv:2006.15680, DOI: **10.48550/arXiv.2006.15680**.
136 F. Doshi-Velez and B. Kim, in *Explainable and Interpretable Models in Computer Vision and Machine Learning*, ed. H. J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü and M. van Gerven, Springer International Publishing, Cham, 2018, pp. 3–17.
137 R. Medar, V. S. Rajpurohit and B. Rashmi, in 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017, pp. 1–6.
138 I. Tougui, A. Jilbab and J. E. Mhamdi, *Healthcare Inf. Res.*, 2021, **27**, 189–199.
139 J.-U. Park, M. Hardy, S. J. Kang, K. Barton, K. Adair, D. K. Mukhopadhyay, C. Y. Lee, M. S. Strano, A. G. Alleyne, J. G. Georgiadis, P. M. Ferreira and J. A. Rogers, *Nat. Mater.*, 2007, **6**, 782–789.
140 M.-S. Lee, K. Lee, S.-Y. Kim, H. Lee, J. Park, K.-H. Choi, H.-K. Kim, D.-G. Kim, D.-Y. Lee, S. Nam and J.-U. Park, *Nano Lett.*, 2013, **13**, 2814–2821.
141 B. W. An, S. Heo, S. Ji, F. Bien and J.-U. Park, *Nat. Commun.*, 2018, **9**, 2458.
142 B. W. An, J. H. Shin, S.-Y. Kim, J. Kim, S. Ji, J. Park, Y. Lee, J. Jang, Y.-G. Park, E. Cho, S. Jo and J.-U. Park, *Polymers*, 2017, **9**, 303.
143 Y.-G. Park, G.-Y. Lee, J. Jang, S. M. Yun, E. Kim and J.-U. Park, *Adv. Healthcare Mater.*, 2021, **10**, 2002280.
144 H. Song, M. Kim, E. Kim, J. Lee, I. Jeong, K. Lim, S. Y. Ryu, M. Oh, Y. Kim and J.-U. Park, *BMEMat*, 2024, **2**, e12048.
145 G. Buzsáki, C. A. Anastassiou and C. Koch, *Nat. Rev. Neurosci.*, 2012, **13**, 407–420.
146 J. Lee, S. Kim, W. G. Chung, E. Kim, H. Song, M. Oh, E. Kim, J. Liu, K.-I. Jang, T. Lee and J.-U. Park, *Adv. Eng. Mater.*, 2024, **26**, 2400499.
147 W. Park, J. Lee, W. G. Chung, I. Jeong, E. Kim, Y. W. Kwon, H. Seo, K. Lim, E. Kim and J.-U. Park, *Nano Energy*, 2024, **124**, 109496.
148 H. Seo, W. G. Chung, Y. W. Kwon, S. Kim, Y.-M. Hong, W. Park, E. Kim, J. Lee, S. Lee, M. Kim, K. Lim, I. Jeong, H. Song and J.-U. Park, *Chem. Rev.*, 2023, **123**, 11488–11558.
149 J. Jang, H. Kim, Y. M. Song and J.-U. Park, *Opt. Mater. Express*, 2019, **9**, 3878–3894.
150 W. G. Chung, J. Jang, G. Cui, S. Lee, H. Jeong, H. Kang, H. Seo, S. Kim, E. Kim, J. Lee, S. G. Lee, S. H. Byeon and J.-U. Park, *Nat. Nanotechnol.*, 2024, **19**, 688–697.
151 N. H. Kim, U. Park, D. W. Yang, S. H. Choi, Y. C. Youn and S. W. Kang, *Sci. Rep.*, 2023, **13**, 10299.
152 W. Ouyang, W. Lu, Y. Zhang, Y. Liu, J. U. Kim, H. Shen, Y. Wu, H. Luan, K. Kilner, S. P. Lee, Y. Lu, Y. Yang, J. Wang, Y. Yu, A. J. Wegener, J. A. Moreno, Z. Xie, Y. Wu, S. M. Won, K. Kwon, C. Wu, W. Bai, H. Guo, T. Liu, H. Bai, G. Monti, J. Zhu, S. R. Madhvapathy, J. Trueb, M. Stanslaski, E. M. Higbee-Dempsey, I. Stepien, N. Ghoreishi-Haack, C. R. Haney, T. Kim, Y. Huang, R. Ghaffari, A. R. Banks,

T. C. Jhou, C. H. Good and J. A. Rogers, *Nat. Biomed. Eng.*, 2023, **7**, 1252–1269.
153 E. Kim, S. Kim, Y. W. Kwon, H. Seo, M. Kim, W. G. Chung, W. Park, H. Song, D. H. Lee, J. Lee, S. Lee, I. Jeong, K. Lim and J.-U. Park, *Interdiscip. Med.*, 2023, **1**, e20230003.
154 J. Jang, Y. S. Jun, H. Seo, M. Kim and J.-U. Park, *Sensors*, 2020, **20**, 3624.
155 W. H. Cheong, B. Oh, S.-H. Kim, J. Jang, S. Ji, S. Lee, J. Cheon, S. Yoo, S.-Y. Lee and J.-U. Park, *Nano Energy*, 2019, **62**, 230–238.
156 J. Jang, B. Oh, S. Jo, S. Park, H. S. An, S. Lee, W. H. Cheong, S. Yoo and J.-U. Park, *Adv. Mater. Technol.*, 2019, **4**, 1900082.
157 S. Kwon, H. S. Kim, K. Kwon, H. Kim, Y. S. Kim, S. H. Lee, Y.-T. Kwon, J.-W. Jeong, L. M. Trotti, A. Duarte and W.-H. Yeo, *Sci. Adv.*, 2023, **9**, eadg9671.
158 B. P. Smith, L. S. Auvil, M. Welge, C. B. Bushell, R. Bhargava, N. Elango, K. Johnson and Z. Madak-Erdogan, *Sci. Rep.*, 2020, **10**, 19128.
159 J. Park, J. Kim, K. Kim, S.-Y. Kim, W. H. Cheong, K. Park, J. H. Song, G. Namgoong, J. J. Kim, J. Heo, F. Bien and J.-U. Park, *Nanoscale*, 2016, **8**, 10591–10597.
160 W. G. Chung, E. Kim, H. Song, J. Lee, S. Lee, K. Lim, I. Jeong and J.-U. Park, *Adv. NanoBiomed Res.*, 2022, **2**, 2200081.
161 J. Park, D. B. Ahn, J. Kim, E. Cha, B.-S. Bae, S.-Y. Lee and J.-U. Park, *Sci. Adv.*, 2019, **5**, eaay0764.
162 Y. Xu, E. De la Paz, A. Paul, K. Mahato, J. R. Sempionatto, N. Tostado, M. Lee, G. Hota, M. Lin, A. Uppal, W. Chen, S. Dua, L. Yin, B. L. Wuerstle, S. Deiss, P. Mercier, S. Xu, J. Wang and G. Cauwenberghs, *Nat. Biomed. Eng.*, 2023, **7**, 1307–1320.
163 R. Kaveh, C. Schwendeman, L. Pu, A. C. Arias and R. Muller, *Nat. Commun.*, 2024, **15**, 6520.
164 J. H. Shin, J. Kwon, J. U. Kim, H. Ryu, J. Ok, S. Joon Kwon, H. Park and T. Kim, *npj Flexible Electron.*, 2022, **6**, 1–12.
165 Y.-G. Park, Y. W. Kwon, C. S. Koh, E. Kim, D. H. Lee, S. Kim, J. Mun, Y.-M. Hong, S. Lee, J.-Y. Kim, J.-H. Lee, H. H. Jung, J. Cheon, J. W. Chang and J.-U. Park, *Nat. Commun.*, 2024, **15**, 1772.
166 U. Topalovic, S. Barclay, C. Ling, A. Alzuhair, W. Yu, V. Hokhikyan, H. Chandrakumar, D. Rozgic, W. Jiang, S. Basir-Kazeruni, S. L. Maoz, C. S. Inman, M. Stangl, J. Gill, A. Bari, A. Fallah, D. Eliashiv, N. Pouratian, I. Fried, N. Suthana and D. Markovic, *Nat. Neurosci.*, 2023, **26**, 517–527.
167 E. Kim, E. Jeong, Y.-M. Hong, I. Jeong, J. Kim, Y. W. Kwon, Y.-G. Park, J. Lee, S. Choi, J.-Y. Kim, J.-H. Lee, S.-W. Cho and J.-U. Park, *Nat. Commun.*, 2025, **16**, 2011.
168 M. Mahmood, D. Mzurikwao, Y.-S. Kim, Y. Lee, S. Mishra, R. Herbert, A. Duarte, C. S. Ang and W.-H. Yeo, *Nat. Mach. Intell.*, 2019, **1**, 412–422.
169 Y. Yu, X. Si, C. Hu and J. Zhang, *Neural Comput.*, 2019, **31**, 1235–1270.
170 Y. W. Kwon, D. B. Ahn, Y.-G. Park, E. Kim, D. H. Lee, S.-W. Kim, K.-H. Lee, W.-Y. Kim, Y.-M. Hong, C. S. Koh, H. H. Jung, J. W. Chang, S.-Y. Lee and J.-U. Park, *Sci. Adv.*, 2024, **10**, eadn3784.
171 M. Kim, J. C. Hwang, S. Min, Y.-G. Park, S. Kim, E. Kim, H. Seo, W. G. Chung, J. Lee, S.-W. Cho and J.-U. Park, *Nano Lett.*, 2022, **22**, 7892–7901.
172 S. Kim, J. Lee, W. G. Chung, Y.-M. Hong, W. Park, J. A. Lim and J.-U. Park, *ACS Nano*, 2024, **18**, 24364–24378.
173 S. Kim, Y.-G. Park, J.-Y. Kim, E. Kim, D. H. Lee, J.-H. Lee, J. Cheon and J.-U. Park, *ACS Appl. Mater. Interfaces*, 2023, **15**, 28954–28963.
174 J. C. Hwang, M. Kim, S. Kim, H. Seo, S. An, E. H. Jang, S. Y. Han, M. J. Kim, N. K. Kim, S.-W. Cho, S. Lee and J.-U. Park, *Sci. Adv.*, 2022, **8**, eabq0897.
175 H. Yang, S. Ji, I. Chaturvedi, H. Xia, T. Wang, G. Chen, L. Pan, C. Wan, D. Qi, Y.-S. Ong and X. Chen, *ACS Mater. Lett.*, 2020, **2**, 478–484.
176 Y.-S. Kim, M. Mahmood, Y. Lee, N. K. Kim, S. Kwon, R. Herbert, D. Kim, H. C. Cho and W.-H. Yeo, *Adv. Sci.*, 2019, **6**, 1900939.
177 T. Cui, Y. Qiao, D. Li, X. Huang, L. Yang, A. Yan, Z. Chen, J. Xu, X. Tan, J. Jian, Z. Li, S. Ji, H. Liu, Y. Yang, X. Zhang and T.-L. Ren, *Chem. Eng. J.*, 2023, **455**, 140690.
178 W. Hwang, J. Kim, S. Park, T.-H. Kang, S. Kim, K. Lee, M.-G. Lee, R. Kwak, I.-S. Choi and H. Yi, *Adv. Mater. Technol.*, 2023, **8**, 2200477.
179 J. Ausra, M. Madrid, R. T. Yin, J. Hanna, S. Arnott, J. A. Brennan, R. Peralta, D. Clausen, J. A. Bakall, I. R. Efimov and P. Gutruf, *Sci. Adv.*, 2022, **8**, eabq7469.
180 Y. S. Choi, H. Jeong, R. T. Yin, R. Avila, A. Pfenniger, J. Yoo, J. Y. Lee, A. Tzavelis, Y. J. Lee, S. W. Chen, H. S. Knight, S. Kim, H.-Y. Ahn, G. Wickerson, A. Vázquez-Guardado, E. Higbee-Dempsey, B. A. Russo, M. A. Napolitano, T. J. Holleran, L. A. Razzak, A. N. Miniovich, G. Lee, B. Geist, B. Kim, S. Han, J. A. Brennan, K. Aras, S. S. Kwak, J. Kim, E. A. Waters, X. Yang, A. Burrell, K. San Chun, C. Liu, C. Wu, A. Y. Rwei, A. N. Spann, A. Banks, D. Johnson, Z. J. Zhang, C. R. Haney, S. H. Jin, A. V. Sahakian, Y. Huang, G. D. Trachiotis, B. P. Knight, R. K. Arora, I. R. Efimov and J. A. Rogers, *Science*, 2022, **376**, 1006–1012.
181 S. Ji, J. Jang, E. Cho, S.-H. Kim, E.-S. Kang, J. Kim, H.-K. Kim, H. Kong, S.-K. Kim, J.-Y. Kim and J.-U. Park, *Adv. Mater.*, 2017, **29**, 1700538.
182 H. Lee, H.-L. Yang, H. G. Ryu, C.-W. Jung, Y. J. Cho, S. B. Yoon, H.-K. Yoon and H.-C. Lee, *npj Digital Med.*, 2023, **6**, 1–10.
183 S. Khamzin, A. Dokuchaev, A. Bazhutina, T. Chumarnaya, S. Zubarev, T. Lyubimtseva, V. Lebedeva, D. Lebedev, V. Gurev and O. Solovyova, *Front. Physiol.*, 2021, **12**, 753282.
184 M. Kim, Y. Kang, S. C. You, H.-D. Park, S.-S. Lee, T.-H. Kim, H. T. Yu, E.-K. Choi, H.-S. Park, J. Park, Y. S. Lee, K.-W. Kang, J. Shim, J.-H. Sung, I.-Y. Oh, J. S. Park and B. Joung, *Sci. Rep.*, 2022, **12**, 37.
185 J. Lázaro, N. Reljin, M.-B. Hossain, Y. Noh, P. Laguna and K. H. Chon, *IEEE Trans. Biomed. Eng.*, 2020, **67**, 3464–3473.
186 Y.-G. Park, J. Jang, H. Kim, J. C. Hwang, Y. W. Kwon and J.-U. Park, *Adv. Electron. Mater.*, 2022, **8**, 2101034.
187 Y.-G. Park, H. Kim, S.-Y. Park, J.-Y. Kim and J.-U. Park, *ACS Appl. Mater. Interfaces*, 2019, **11**, 41497–41505.

188 Z. Wang, Y. Dong, X. Sui, X. Shao, K. Li, H. Zhang, Z. Xu and D. Zhang, *npj Flexible Electron.*, 2024, **8**, 1–11.
189 A. Poddar, N. Rangwani, S. Palekar and J. Kalambe, *Mater. Today: Proc.*, 2023, **73**, 100–107.
190 K. Cierpiak, P. Wityk, M. Kosowska, P. Sokołowski, T. Talaśka, J. Gierowski, M. J. Markuszewski and M. Szczerska, *Sci. Rep.*, 2024, **14**, 18854.
191 C. Xu, Y. Song, J. R. Sempionatto, S. A. Solomon, Y. Yu, H. Y. Y. Nyein, R. Y. Tay, J. Li, W. Heng, J. Min, A. Lao, T. K. Hsiai, J. A. Sumner and W. Gao, *Nat. Electron.*, 2024, **7**, 168–179.
192 D. Sankhala, A. U. Sardesai, M. Pali, K.-C. Lin, B. Jagannath, S. Muthukumar and S. Prasad, *Sci. Rep.*, 2022, **12**, 2442.
193 H. S. Mondal, K. A. Ahmed, N. Birbilis and M. Z. Hossain, *Sci. Rep.*, 2023, **13**, 3742.
194 H. Song, H. Shin, H. Seo, W. Park, B. J. Joo, J. Kim, J. Kim, H. K. Kim, J. Kim and J.-U. Park, *Adv. Sci.*, 2022, **9**, 2203597.
195 J. Jang, J. Kim, H. Shin, Y.-G. Park, B. J. Joo, H. Seo, J. Won, D. W. Kim, C. Y. Lee, H. K. Kim and J.-U. Park, *Sci. Adv.*, 2021, **7**, eabf7194.
196 J. Kim, J. Park, Y.-G. Park, E. Cha, M. Ku, H. S. An, K.-P. Lee, M.-I. Huh, J. Kim, T.-S. Kim, D. W. Kim, H. K. Kim and J.-U. Park, *Nat. Biomed. Eng.*, 2021, **5**, 772–782.
197 J. Kim, J. Kim, M. Ku, E. Cha, S. Ju, W. Y. Park, K. H. Kim, D. W. Kim, P.-O. Berggren and J.-U. Park, *Nano Lett.*, 2020, **20**, 1517–1525.
198 H. Shin, H. Seo, W. G. Chung, B. J. Joo, J. Jang and J.-U. Park, *Lab Chip*, 2021, **21**, 1269–1286.
199 Y.-G. Park, E. Cha, H. S. An, K.-P. Lee, M. H. Song, H. K. Kim and J.-U. Park, *Nano Res.*, 2020, **13**, 1347–1353.
200 S. Ji, J. Jang, J. C. Hwang, Y. Lee, J.-H. Lee and J.-U. Park, *Adv. Mater. Technol.*, 2020, **5**, 1900928.
201 Z. Che, X. Wan, J. Xu, C. Duan, T. Zheng and J. Chen, *Nat. Commun.*, 2024, **15**, 1873.
202 T. Kim, Y. Shin, K. Kang, K. Kim, G. Kim, Y. Byeon, H. Kim, Y. Gao, J. R. Lee, G. Son, T. Kim, Y. Jun, J. Kim, J. Lee, S. Um, Y. Kwon, B. G. Son, M. Cho, M. Sang, J. Shin, K. Kim, J. Suh, H. Choi, S. Hong, H. Cheng, H.-G. Kang, D. Hwang and K. J. Yu, *Nat. Commun.*, 2022, **13**, 5815.
203 A. Moin, A. Zhou, A. Rahimi, A. Menon, S. Benatti, G. Alexandrov, S. Tamakloe, J. Ting, N. Yamamoto, Y. Khan, F. Burghardt, L. Benini, A. C. Arias and J. M. Rabaey, *Nat. Electron.*, 2021, **4**, 54–63.
204 K. K. Kim, M. Kim, K. Pyun, J. Kim, J. Min, S. Koh, S. E. Root, J. Kim, B.-N. T. Nguyen, Y. Nishio, S. Han, J. Choi, C.-Y. Kim, J. B.-H. Tok, S. Jo, S. H. Ko and Z. Bao, *Nat. Electron.*, 2023, **6**, 64–75.
205 H. Xu, W. Zheng, Y. Zhang, D. Zhao, L. Wang, Y. Zhao, W. Wang, Y. Yuan, J. Zhang, Z. Huo, Y. Wang, N. Zhao, Y. Qin, K. Liu, R. Xi, G. Chen, H. Zhang, C. Tang, J. Yan, Q. Ge, H. Cheng, Y. Lu and L. Gao, *Nat. Commun.*, 2023, **14**, 7769.
206 J.-H. Kim, C. Marcus, R. Ono, D. Sadat, A. Mirzazadeh, M. Jens, S. Fernandez, S. Zheng, T. Durak and C. Dagdeviren, *Nat. Electron.*, 2022, **5**, 794–807.
207 M. Kim, S. Kim, Y. W. Kwon, H. Seo, W. G. Chung, E. Kim, W. Park, H. Song, D. H. Lee, J. Lee, S. Lee, I. Jeong, K. Lim, D.-Y. Jo and J.-U. Park, *Adv. Sens. Res.*, 2023, **2**, 2200049.
208 Y.-G. Park, I. Yun, W. G. Chung, W. Park, D. H. Lee and J.-U. Park, *Adv. Sci.*, 2022, **9**, 2104623.
209 Y.-G. Park, H. S. An, J.-Y. Kim and J.-U. Park, *Sci. Adv.*, 2019, **5**, eaaw2844.
210 B. P. Mathew, H. J. Yang, J. Kim, J. B. Lee, Y.-T. Kim, S. Lee, C. Y. Lee, W. Choe, K. Myung, J.-U. Park and S. Y. Hong, *Angew. Chem.*, 2017, **129**, 5089–5093.
211 Y. Jo, J. Y. Kim, S.-Y. Kim, Y.-H. Seo, K.-S. Jang, S. Y. Lee, S. Jung, B.-H. Ryu, H.-S. Kim, J.-U. Park, Y. Choi and S. Jeong, *Nanoscale*, 2017, **9**, 5072–5084.
212 D. Göndöcs and V. Dörfler, *Artif. Intell. Med.*, 2024, **149**, 102769.
213 A. S. Chandrabhatla, I. J. Pomeraniec, T. M. Horgan, E. K. Wat and A. Ksendzovsky, *npj Digital Med.*, 2023, **6**, 1–13.
214 A. M. Oliveira, L. Coelho, E. Carvalho, M. J. Ferreira-Pinto, R. Vaz and P. Aguiar, *J. Neurol.*, 2023, **270**, 5313–5326.