26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Machine Learning for Tactile Perception: Advancements, Challenges, and Opportunities

Zhixian Hu, Lan Lin, Waner Lin, Yingtian Xu, Xuan Xia, Zhengchun Peng, Zhenglong Sun, and Ziya Wang**

The past decades have seen the rapid development of tactile sensors in material, fabrication, and mechanical structure design. The advancement of tactile sensors has heightened the expectation of sensor functions, and thus put forward a higher demand for data processing. However, conventional analysis techniques have not kept pace with the tactile sensor development and still suffer from some severe drawbacks, like cumbersome models, poor efficiency, and expensive costs. Machine learning, with its prominent ability for big data analysis and fast processing speed, can offer many possibilities for tactile data analysis. Herein, the machine learning techniques employed for processing tactile signals are reviewed. Supervised learning and unsupervised learning for analog signals are covered, and processing spike signals with machine learning are summarized. Furthermore, the applications in robotic tactile perception and human activity monitoring are presented. Finally, the current challenges and future prospects in sensors, data, algorithms, and benchmarks are discussed.

1. Introduction

It always bothers researchers to find an appropriate method to bridge the gap between the data they collect and the goals they pursue, no exception for people who study tactile sensors. With the development of material science, electronic engineering, and fabrication technology, tactile sensors have evolved from cumbersome bodies to miniature forms, enriched the functions from single variable sensing to various stimulus perceptions, increased the sensing ability while reducing the sensor sizes, and become more scalable.[1–4] As a consequence, both information per unit area and the total amount of information provided by the tactile sensors have boomed. The data have become more complex with the coupling of multiple signals. On the one hand, this advancement has refined and densified tactile signals, enabling a better understanding of tactile information. On the other hand, the improvement has presented new challenges for tactile signal processing in the aspects of running speed, signal decoupling, and information extraction. Traditionally, after precise calibration on tactile sensors, researchers perform statistical analysis, employ transformation theories, count on physical models or simulations, or utilize other mathematical tools to obtain further information from raw data.[5–9] However, these data processing approaches are inadequate when dealing with complex tactile sensing issues, such as providing rapid responses, analyzing heterogeneous signals, offering accurate sensation judgment, and handling large-scale data analysis. While in other perception and understanding fields like computer vision, speech recognition, and natural language processing, machine learning, an emerging data processing technique, has grown by leaps and bounds and achieved outstanding performance in analyzing enormous data streams quickly.[10,11] As machine learning has the advantages of high precision, strong robustness, and competitive computation speed, researchers in tactile sensing have attempted to utilize this technique to understand tactile data further for better practical applications.

According to inputs of machine learning algorithms and sensor types, tactile data are mainly divided into two categories: analog and spike (**Figure 1**). Analog signals are continuous signals with arbitrary values, which are common data types obtained from general tactile sensors like piezoresistive, piezocapacitive, piezoelectric, triboelectric, optical, bio-photocapacitive, and

Z. Hu, L. Lin, Z. Peng, Z. Wang
Center for Stretchable Electronics and Nano Sensors
College of Physics and Optoelectronic Engineering
Shenzhen University
Shenzhen 518060, China
E-mail: wangziya@szu.edu.cn

Z. Hu, Y. Xu, Z. Sun
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
Shenzhen 518172, China
E-mail: sunzhenglong@cuhk.edu.cn

W. Lin
Department of Micro-Nano Electronics
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai 200240, China

X. Xia
Shenzhen Institute of Artificial Intelligence and Robotics for Society
Shenzhen 518129, China

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202200371.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202200371

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

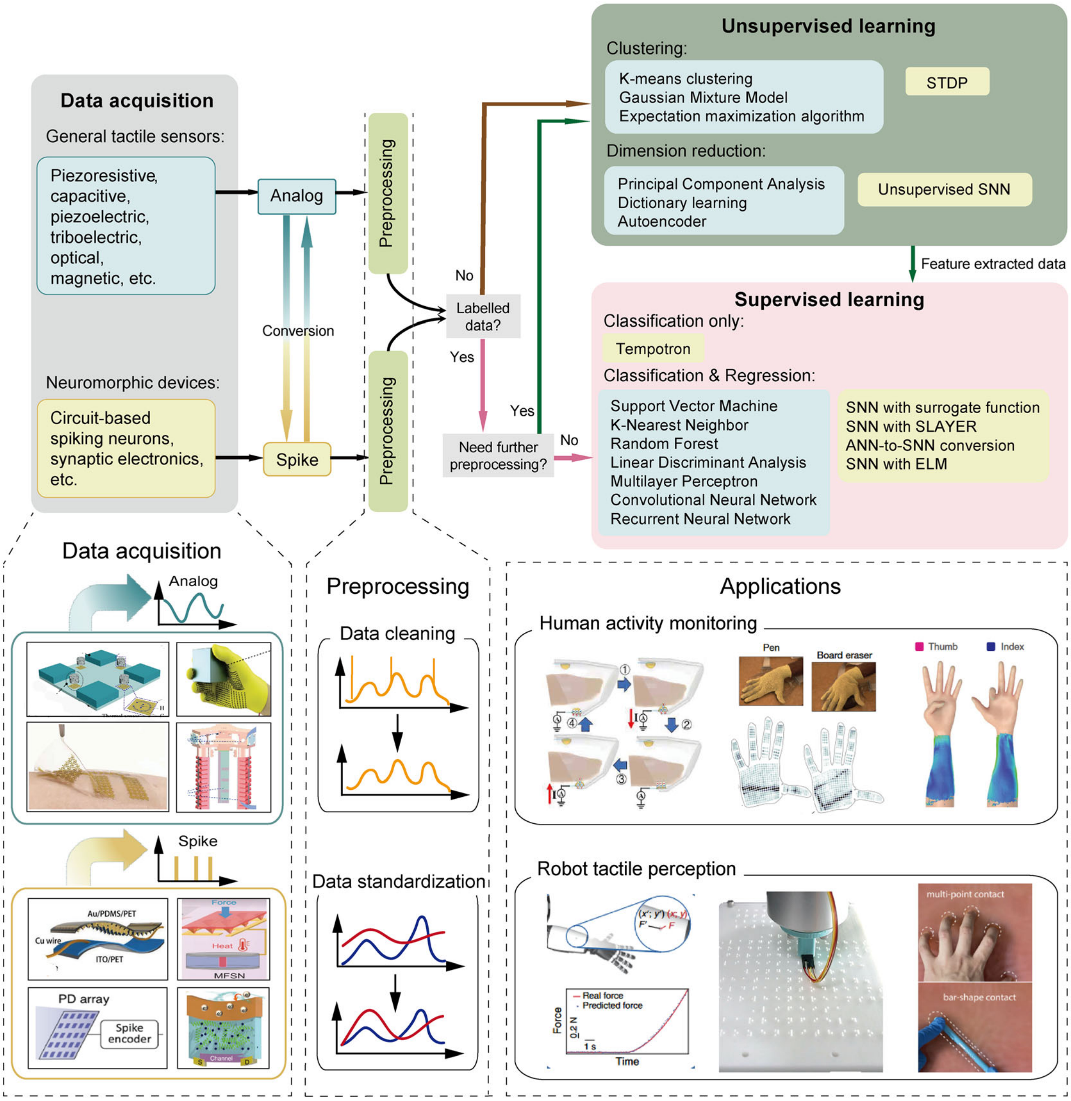

Figure 1. Overview of machine learning workflow for tactile sensing. The process contains tactile data acquisition, preprocessing, feature extraction, and machine learning algorithms. With obtained signals and related task goals, readers could follow the flow chart and choose appropriate processing methods to obtain desired results for application. In the panel of data acquisition, the blue block denotes sensors for analog signals, and the yellow block denotes sensors for spike signals. In the unsupervised and supervised learning panels, algorithms in the blue block represent analog signal methods, whereas algorithms in the yellow block represent spike signal methods. Figures for analog signal generation: Top-left, Reproduced with permission.[51] Copyright 2022, Wiley-VCH. top-right, Reproduced with permission.[55] Copyright 2019, Springer Nature; bottom-left, Reproduced with permission.[12] Copyright 2022, AAAS; bottom-right, Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[44] Copyright 2020, The Authors, published by Springer Nature. Figures for spike signal generation: top-left, Reproduced with permission.[32] Copyright 2019, Wiley-VCH; top-right, Reproduced with permission.[33] Copyright 2022, Wiley-VCH; bottom-left, Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[177] Copyright 2021, The Authors, published by Springer Nature; bottom-right, Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[178] Copyright 2020, The Authors, published by Springer Nature. Figures for application in human activity monitoring: left, Reproduced with permission under the terms of CC BY-NC Creative Commons Attribution NonCommercial License 4.0 (https://creativecommons.org/licenses/by-nc/4.0/).[45] Copyright 2020, The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. Published by AAAS; middle, Reproduced with permission.[55] Copyright 2019, Springer Nature; right, reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[110]. Copyright 2020, The Authors, published by Springer Nature. Figures for application in robotic tactile perception: left, Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[54] Copyright 2022, The Authors, published by Springer Nature; middle, Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[179] Copyright 2022, The Authors, published by Wiley-VCH; right, Reproduced with permission.[111] Copyright 2022, AAAS.

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

magnetic sensors.[12–23] These sensors have been widely applied in wearable electronic sensing systems,[24] bio-integrated sensing systems,[25] robot perception,[2,26,27] and other areas. Though traditional tactile signals are analog signals, a trend for generating spikes has been growing steeply in designing tactile sensors during recent years due to the favor of the spikes' power efficiency, short delay, and, more importantly, physiological basis.[27–29] Spikes are emitted by some tactile sensors connected with neuromorphic structures like circuit-based spiking neurons[30,31] or from synaptic electronics.[32–36] Ideal spike signals are all-or-none valued and are usually represented by discrete single-bit events.[37] A more broad definition of spike signals derived from neurology is like a specific cell location's membrane potential, which has two states, excitement and rest.[38] The excitement state is where the signal has a sharp rising edge and then a sharp falling edge, and the rest state is where the signal keeps a static value. Although spike signals have demonstrated considerable promise for tactile sensing, they are still mostly disregarded in reviews on tactile data processing. This review tries to make up for this deficiency. One typical difference between analog signals and spikes is that analog signals are collected on a common clock cycle with arbitrary values, whereas spikes signal the occurrence of some characteristic events with temporal precision.[39] The processing of analog signals focuses more on the quantified values, while the processing of spike signals pays more attention to the temporal information of events. The distinctions between clock-based analog signals and event-based spike signals lead to differences in processing methods. Thus, separated machine learning algorithms have been designed and utilized to deal with the two types of inputs. Raw data from tactile sensors, whether analog or spike signals, are usually irregular, heterogeneous, and complex.[3] They require cautious and powerful treatment to reveal the underlying high-level information.

Processing tactile data with machine learning contains several steps: data acquisition, preprocessing, feature extraction, and machine learning algorithm (Figure 1). Data are acquired from tactile sensors that quantify factors of contact interaction between the sensors and some stimuli.[40] Biological tactile receptors perceive mechanical, thermal, and other stimuli, which are transmitted to the central nervous system for tactile perception.[41] Researchers try to mimic the biological sense of tactile and have designed tactile sensors measuring force, temperature, humidity, and other signals during physical interactions. After obtaining tactile signals, preprocessing is conducted first. Preprocessing usually includes data cleaning and data standardization. Raw data obtained from tactile sensors inevitably contain noises or other interference signals. Data cleaning tries to eliminate these disturbances by methods like filters. Also, corrupt and inaccurate data are detected and corrected, either manually or by relying on some algorithms (e.g., clustering methods[42]). Data standardization is required to transform data from different tactile sources, with varying scales of time, or from inconsistent sensors into a consistent standard for the ease of further analysis. Preprocessing is crucial for tactile information processing, not limited to the scope of machine learning but general in all data processing procedures. After data preprocessing, feature extraction is implemented by computing defined features or using machine learning techniques. For tactile data of low dimension, feature extraction can be omitted. While with the trend of developing tactile sensors with high density and fast sampling frequency,[2,43] many high-dimensional tactile signals have come up and raised the need for feature extraction. Some processing methods seem not to include extra steps of feature extraction. Still, the effects of feature extraction can be achieved through the intermediate stages of machine learning algorithms. Interlayers of neural networks are a typical example of this. The selections of features and corresponding feature extraction methods are important for the accuracy of the final results. Decent choices not only maintain the most information of the tactile signals but also reduce the data redundancy maximally. Then the processed data are fed into a machine learning algorithm to fulfill the desired application goals for tactile sensing. The selection of features and the choice of machine learning algorithms both play an important role in the final results. On the one hand, with appropriate feature extraction strategies, even a simple machine learning algorithm can provide a good result. On the other hand, a robust enough machine learning algorithm can perform satisfyingly even without feature extraction steps. In practice, a combination of careful selections of both aspects provides researchers with powerful data processing tools.

Machine learning tasks can be roughly categorized into four classes: classification, regression, clustering, and dimension reduction. Classification and regression are tasks of supervised learning in which data are labeled, and algorithms target predicting corresponding data labels. Classification deals with discrete labels, for example, predicting which object that tactile sensor is touching,[44–48] detecting whether a slip happens.[49,50] Regression is defined as solving tasks with continuous labels, such as estimating contact force[20,51,52] and position.[20] When continuous values are thresholded or discretized, regression problems can be reformulated into classification tasks.[53] For example, the force estimation problem mentioned above can be transferred into several well-defined force-level classification problems.[54] Though conventional data analysis methods can solve simple classification and regression tasks in tactile sensing, when dealing with classification and regression problems on difficult-to-model or hard-to-optimize values, models of conventional methods become cumbersome and may not achieve satisfying results. Whereas machine learning, with its data-driven traits, serves as a powerful tool to accomplish these kinds of tasks. Clustering and dimension reduction are tasks of unsupervised learning that aim at discovering hidden patterns of data. Clustering is to gather data in groups based on similarities, which is mostly used to analyze tactile data.[55] While sometimes, clustering can also be used for data cleaning in tactile sensing.[42] Dimension reduction, sometimes referred to as feature extraction, is to find a low-dimensional representation of high-dimensional data. Due to the tendency of a growing volume of tactile input, dimension reduction is becoming more and more common in tactile sensing. Machine learning is capable of handling other tasks like anomaly detection and associative learning in addition to the four previously listed ones. Nevertheless, these tasks are uncommon among the uses of tactile sensors. Meanwhile, besides supervised learning and unsupervised learning, reinforcement learning is another typical kind of machine learning algorithm. Reinforcement learning emphasizes solving decision-making problems and is commonly used in robotics manipulation for computing next-step actions. Except few cases directly taking tactile signals as inputs,[56] processing tactile signals for reinforcement learning is usually accomplished by supervised learning,[57–59] unsupervised learning,[60–64]

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

or other data processing techniques to perceive current states. Readers interested in reinforcement learning for robot manipulation can refer to these papers.[26,65,66] In a nutshell, the main objective of this review is to the algorithms and applications for the four tasks of classification, regression, clustering, and dimension reduction and the two machine learning categories of supervised learning and unsupervised learning.

In this article, current advances in the use of machine learning for tactile sensing are reviewed and discussed. This review focuses on connecting machine learning algorithms to analog and spike signal types for various tactile application goals, which could serve as a guide to choosing appropriate algorithms for tactile cognition. There exist some reviews on machine learning involving tactile data processing, such as dealing with stretchable sensing signals for human–machine interfaces,[3] processing data from wearable sensing electronic systems,[24] and fusing multimodal data for robots.[67] However, few reviews offered a comprehensive discussion on machine learning for tactile information processing, especially a lack of discussions on choosing appropriate algorithms to assist tactile cognition. This review aims at filling this gap. Besides, as spikes demonstrate huge potential as tactile signals in sensing, cognition, and feedback, this review covers the content on spike signal processing, which is seldom discussed in previous reviews. Section 2 demonstrates the processing of analog signals using unsupervised and supervised learning. The processing of spike signals follows. Section 4 covers applications of tactile sensing with machine learning. The final section includes the conclusion as well as the discussions on challenges and perspectives.

2. Machine Learning for Analog Signals

In tactile sensing, signals in analog form are the mainstream. Due to the technological improvement of sensor fabrication and emerging multilayer structures, analog signals have diversified in tactile sensing. Machine learning-assisted signal processing procedure is introduced to fit data that contain unknown information and are hard to construct physical models.

2.1. Unsupervised Learning

Unsupervised learning is a type of machine learning algorithm that examines unlabeled data in order to find internal relationships or hidden patterns. Unlabeled data means that the algorithms are unaware of the practical implications or manual labeling of the data. Without human intervention, unsupervised learning picks up information from the data. Unsupervised learning occurs in tactile sensing for the two tasks of data clustering and dimension reduction. Despite neither of the tasks typically being the primary objective of tactile sensing, there is an inclination toward including these tasks in processing tactile data due to the growing demand for the analysis of complicated signals and the potent capabilities of matching machine learning algorithms.

2.1.1. Data Clustering

Data clustering is to group data based on their similarities. It is one of the most popular studying directions in unsupervised learning and has been widely studied by researchers. Because tactile sensor researchers usually have a specific goal on what to obtain from tactile data, clustering always functions as an intermediate step of data processing.

Among various clustering techniques, *k*-means clustering is commonly used in processing tactile data. *K*-means clustering is to cluster data into *k* groups, and each piece of data belongs to the group whose mean is the nearest to the data point.[68] In tactile sensing, *k*-means clustering has been used to minimize the redundancy of continuously collected data (**Figure 2**a).[55] A scalable tactile glove was designed to collect data, which consisted of a piezoresistive film and 548 conductive thread electrodes, aiming to learn the signatures of human grasp. Data were collected when participants grasped different objects, and a certain number of frames (*N* frames) were fed into a deep neural network for classification. A strategy of maximizing data variance was utilized to reduce data redundancy during evaluation. Tactile signals' dimension was first reduced, then k-means clustering was applied to find *N* clusters. For any randomly chosen input, the frame was complemented with $N-1$ frames from the other clusters, and these frames were input to a deep neural network. Besides reducing data redundancy, *k*-means clustering is also used to clean data. An attempt was made by Lee et al.[42] A highly stretchable cross-reactive sensor matrix was designed, and this work managed to discriminate strain, pressure, flexion, and temperature signals from intermixed stimuli using a machine learning algorithm. This work used finite element analysis to simulate a large amount of data for training. Key points were extracted using speed-up robust features (SURF) algorithms with simulated intermixed stimuli images. Then *k*-means clustering was applied, removing the duplicated and weak key points. Support vector machine (SVM) was utilized for further classification with the cleaned key points. It is worth mentioning that k-means clustering is an NP-hard problem with local optima, which indicates that no efficient algorithm guarantees a good cluster assignment.[69]

Besides k-means clustering, Gaussian mixture model[70] and expectation maximization algorithm[71,72] are also common and useful unsupervised learning algorithms for data clustering and can be used to process tactile data. Data clustering is effective for low-dimensional data, but it becomes challenging to scale to high-dimensional data. For this reason, clustering algorithms sometimes are paired with dimension reduction techniques.

2.1.2. Dimension Reduction

Dimension reduction is to transform data from a high-dimensional space to a low-dimensional subspace while preserving as much information as possible. When dealing with high-dimensional tactile signals, researchers sometimes encounter problems of redundant sparse data or heavy computation load of large-scale data, which is called the "curse of dimensionality".[73,74] Dimension reduction is an effective technique to handle this problem. Besides solving the curse of dimensionality, dimension reduction is also used for visualization and regularization. To find a low-dimensional representation of tactile data, principal component analysis (PCA), dictionary learning, and autoencoder are three common approaches researchers use.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

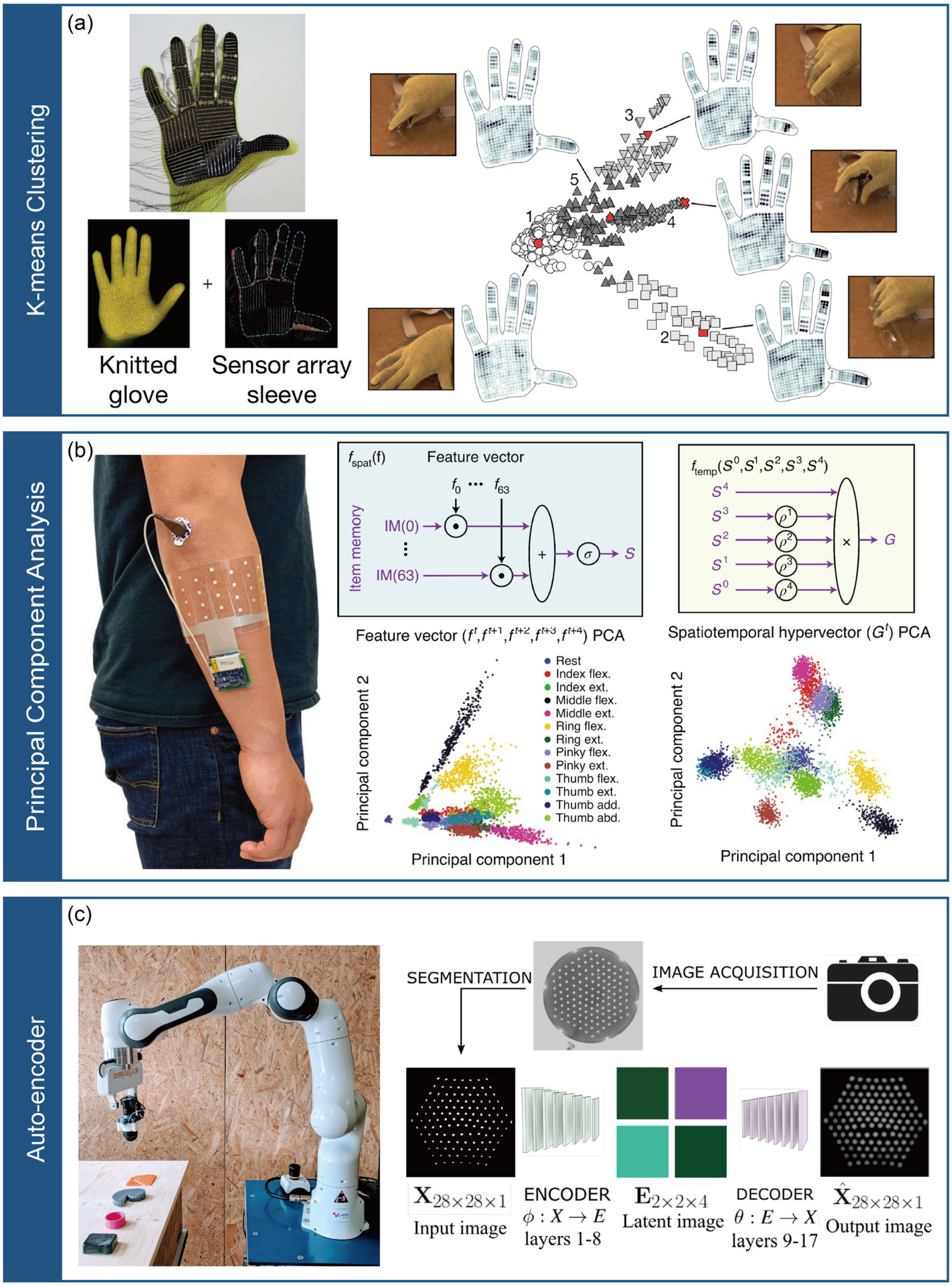

Figure 2. Unsupervised learning for analog signals. a) Data clustering with *k*-means clustering. Reproduced with permission.[55] Copyright 2019, Springer Nature. b) Dimension reduction with PCA. Reproduced with permission.[78] Copyright 2020, Springer Nature. c) Dimension reduction with auto-encoder. Reproduced with permission.[85] Copyright 2019, IEEE.

With the belief that directions with more variance contain more information, PCA was proposed to find principal components in which directions that maximize the variation of data projections.[75,76] Thus, PCA can find meaningful directions from unlabeled data and thereby process high-dimensional data to a manageable size for further data analysis. PCA was used to reduce the dimensionality of the tactile glove data from 548 to 8 for further data clustering.[55] Besides, PCA was applied

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

to reduce the size of data features from 600 to 200 for further machine learning classification to identify various objects, using data from triboelectric nanogenerator (TENG) sensors on a soft-robotic gripper.[44] PCA was also used by Zhou et al.[77] to process tactile data from yarn-based stretchable sensor arrays on hands, which extracted features for translating gestures to speech. Besides reducing computation load, PCA is used for data visualization to analyze data (Figure 2b).[17,48,78] Wen et al. designed a triboelectric smart glove to recognize sign language using a convolutional neural network (CNN).[17] To analyze the clustering performance of the CNN, they used PCA to reduce the data of the input layer and the last fully connected layer to 30 dimensions and plotted the feature clustering results by the t-distributed Stochastic Neighbor Embedding. The clustering results validated the effectiveness of the CNN for feature classification.

Dictionary learning,[79–81] in other words, sparse dictionary learning, is to find a sparse representation of data over a dictionary in the form of a linear combination of the dictionary's atoms. With a data set, dictionary learning finds a fixed-size dictionary in which the data representations have the maximum sparsity using atoms in the dictionary. The data dimensions are reduced to the maximum number of atoms used to represent input data. Roberge et al.[82] used a capacitive sensor to acquire pressure data and preprocessed the data by a log Mel-filter bank. Then they found sparse representations of tactile data spectrograms by dictionary learning and sent the sparse vectors to SVM for further dynamic tactile event classification. For feature extraction, Liu et al.[83] applied dictionary learning to the Penn Haptic Adjective Corpus 2 data set.[84] They proposed a reduced extreme kernel dictionary learning algorithm and used a dictionary of size 300 to reduce the data size for extreme learning machine (ELM) classification algorithms.

An autoencoder is a feedforward neural network that learns to code itself in the middle layers and predicts original inputs at the output layer.[69] The former part of the neural network is called the encoder, which is always used for dimension reduction. The latter part of the neural network is called the decoder, which tries to recover the network input data using the output features of the encoder. With input as targets, autoencoders learn unsupervised to encode data efficiently and obtain representative data features for data compression or further processing.[61–63,85] The work[85] used a convolutional neural network autoencoder for feature extraction (Figure 2c). A bio-inspired optical tactile sensor was used to collect data, and the collected signal was resized to $28 \times 28 \times 1$. To extract features, the researchers trained the encoder and decoder in a way that the encoder extracted $2 \times 2 \times 4$ features from the tactile signals, and the decoder recovered $2 \times 2 \times 4$ features to approximate the original data.

Dimension reduction is useful for extracting features in data. However, following dimension reduction, it is impossible to preserve both global and local information of the data.[53] Therefore, information loss is unavoidable. This issue may become severe when analyzing tactile data that contains a lot of coupling information. Under this circumstance, researchers have to make the decisions as to whether the traits they are extracting will be relevant for later processing.

Unsupervised learning typically works for data analysis or as an intermediate step in the data processing process. As it attempts to discover underlying data patterns without human supervision, the noise would cause issues for algorithms to cluster or find features correctly, and sometimes could lead to contradicting conclusions. Preprocessing is, therefore, crucial as a step before unsupervised learning. There have also been attempts to apply unsupervised learning to recognition tasks, in addition to tasks involving clustering and data dimension, but the results have often been disappointing.[3] Nevertheless, unsupervised learning in tactile sensing has received considerable attention recently. With the further development of tactile sensors, unsupervised learning will continue to be a vital data processing tool.

2.2. Supervised Learning

Supervised learning is a type of machine learning that trains on labeled data and makes predictions on incoming, untrained data. In contrast to the unsupervised method, the labeled data require manual work to obtain an accurate label. Thus, the outputs can be validated as correct or wrong predictions. The data set used in supervised learning is usually divided into training, validation, and test set. Herein, supervised learning has several steps: training on the training set, assessing the validation set to tune hyperparameters, and evaluating on the testing set to select an optimal model. The model is adopted when it performs best compared with other models, and the results on the test set are satisfactory.

2.2.1. Statistical Algorithms

Algorithms based on statistical theory give methods for extracting features, creating models from data, and analyzing data. In statistical learning, it is hypothesized that data are independent and identically distributed. Researchers strive to determine the most suitable model from the training data set, which delivers the best prediction that fits the assessment criterion for both training data and test data. Typical algorithms used in tactile sensing include SVM, *K*-Nearest Neighbor (KNN), decision tree, random forest (RF), and linear discriminant analysis (LDA).

SVM was initially proposed to find the optimal separating hyperplane in a high-dimensional space that distinctly classifies two classes of data points.[86] The region bounded between two hyperplanes is a "margin." The optimal choice of hyperplane is realized by maximizing the margin, including soft margin and hard margin.[87] The "hard" margin follows the original design of SVM, which means that there are no single data points on the wrong side of the bound and no point located in the margin. While in practical usage, such harsh standards make it hard for data sets to fit in. Thus, the "soft" margin was designed, which slightly tolerates some fault data and allows some data points in the margin. The thought of SVM, i.e., maximize margin, can be formalized to find an optimized resolution of a convex quadratic problem. By introducing the one-against-all and one-against-one principle,[88] classical two-class SVM can be extended to a multiclass SVM classifier, which enables SVM to classify diverse categories. Originally, SVM was only capable of dealing with linear problems. With the help of the kernel method,[87] SVM was extended to solve nonlinear problems. Researchers achieved various classifications with SVM in tactile sensing, including object recognition, sign language translation, gesture recognition, and action classification.[44,45,77,89,90] SVM is also a popular choice

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

when different algorithms need to be tested and compared, which usually gets good classification accuracy.[90,91] As shown in **Figure 3**a, SVM was applied to classify objects with sixteen different geometrical features through grasping information on grippers.[44] The sensitive TENG sensors were embedded in the grippers. For each gripper, a long TENG is located along the gripper that can measure the length of the touch area when the gripper works. These sensors generated the raw data, then feature-extracted data from PCA were fed into the SVM model. After training, the model could predict real-time signals. In addition to solving classification problems, SVM holds the ability to estimate a continuous-valued multivariate function,[92] which enables it to deal with regression tasks, including vibration sensing,[93] and force estimation.[20] Barreiros et al.[20] reported a soft, optical, robotic flesh that was able to encode haptic stimuli and recognize contact force, position and gesture. SVM was used to regress the force, as well as classify touch location and gesture. The training subsystem consists of a feature extraction and a model selection module (Figure 3b). The SVM regression performed the best in force intensity estimation with a mean absolute error of 0.32 N. Generally, SVM is suitable for processing small-scale data as long as there are plenty of distinguishable features.[45] The SVM gets the advantage of high accuracy and robustness for both classification and regression. The drawback of SVM is that the performance of SVM deeply relies on the choice of the kernel function, whereas many kernel functions are not flexible. Furthermore, a subtle change of parameter in kernel function and soft margin would influence the model heavily.

KNN computes distances between data samples with no additional parameters. It is based on the assumption that similar data points are located near each other. The classification working principle is to get all samples, then calculate distances between data points and their neighbors, and pick k closest neighbor points. According to the voting or averaging mechanism, it can be used for classification or regression. Commonly used distance definitions are euclidean metrics and Minkowski metrics. The choice of k is an effective factor of performance. If k is chosen too small, the trained model would be complicated and easily get overfit. If k is too big, the classification results may be meaningless. A general approach to the choice of k is to pick a small k value, then verify it with cross-validation until finding the appropriate value. In the study of Van and his colleagues,[19] KNN achieved great performance in both classification and regression tasks compared with other methods. KNN was trained to classify whether the sensor was bent or twisted with a test error rate of 0. It also predicted the magnitude ranging from −80° to 90° in both bending and twisting. The regression model got the lowest mean absolute error of 0.06° (Figure 3c). Besides, Yu et al.[12] reported an ultrasensitive physicochemical skin-based interface encoded surface electromyography signals through the KNN method for remote robotic control. This KNN-assisted human–machine interactive multimodal sensing robotic system successfully achieved gesture recognition and gesture-controlled robotic hand control (Figure 3d). The KNN-powered interface-enabled gesture recognition provided a framework for online multidirectional robotic control with high-accuracy remote object manipulation. KNN was also applied to classify words in real-time silent speech systems[94] and electromyogram (EMG) signals.[95] Although the architecture of KNN is less complicated than other machine learning algorithms like the Bayesian method, what is counterintuitive is that the generalization error rate of KNN is no more than twice that of the Bayesian optimal classifier. A basic assumption induces the drawback of KNN: a data point can always be found at a close distance. It indicates that the training samples should be dense enough to find all neighbor points. However, naturally collected data sets usually do not meet the density requirement.

The decision tree is a multicharacter combined process based on a tree model. Generally, a decision tree consists of a single root node, leaf nodes covering all samples, i.e., the decision results, and many inertial nodes representing the attribution of leaf nodes. The flow path of the decision tree follows the classical strategy: divide and conquer. A decision tree has been generally used as an option in tactile processing and a comparison algorithm for model selection.[19,50,90] In most cases, it is not as competitive as methods like SVM and deep neural networks. However, when ensemble learning[96] was involved, many decision trees combined in bagging or boosting could improve the performance significantly. For general usage in tactile sensing, the bagging ensemble approach is RF.[97,98] In the research of Chunv and his colleagues, the RF was applied to detect scratching activities.[99] The RF was also used in object recognition through one grasp action (Figure 3e).[100] In the signal processing pipeline, the RF classifier returned scratching action predictions immediately, with an overall 89% accuracy (Figure 3f). Besides, the RF was used in fruit quality detection.[101] The RF method is fast and efficient when dealing with large databases and has the capability to handle high numbers of input variables. However, the RF tends to overfit if there is too much data noise.

LDA is to find the feature that separates different classes in the data set. By maximizing the variance of data points in different classes and minimizing the variance of data points in the same class, LDA is able to separate classes in the given data set. In tactile sensing, LDA was applied to classify surface electromyographic (sEMG) information around the mouth in the study of Wang et al.[91] Based on tattoo-like, wearable electrodes attached to the face, the subtle deformation of facial skin was recognized. Thus, the silent speech recognition system collected facial sEMG signals and used LDA to categorize 110 words into 13 classes with an accuracy of 92.64%. Moreover, LDA is a method of dimension reduction. Qu et al.[18] used LDA to acquire two-dimensional, visual data from high-dimensional data. After dimension reduction, the material classification accuracy and the material roughness prediction were significantly improved, from 52.7% to 96.8%. LDA was additionally applied in eye motion feature classification,[102] gesture recognition,[15] and objects classification.[100]

Besides, other traditional machine learning methods, including Naive Bayes (NB)[103,104] and linear regression,[105] were also applied to process tactile signals. Statistical algorithms in tactile applications can reach high accuracy and get admirable speed. However, all these satisfying performances require careful manual work in algorithm designing and parameter adjusting. Moreover, in conventional statistical machine learning, increasing data amount would lead the performance to a plateau. With the improvement of the resolution of tactile sensors and some specific requirements in real tasks, the artificial neural network was introduced.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

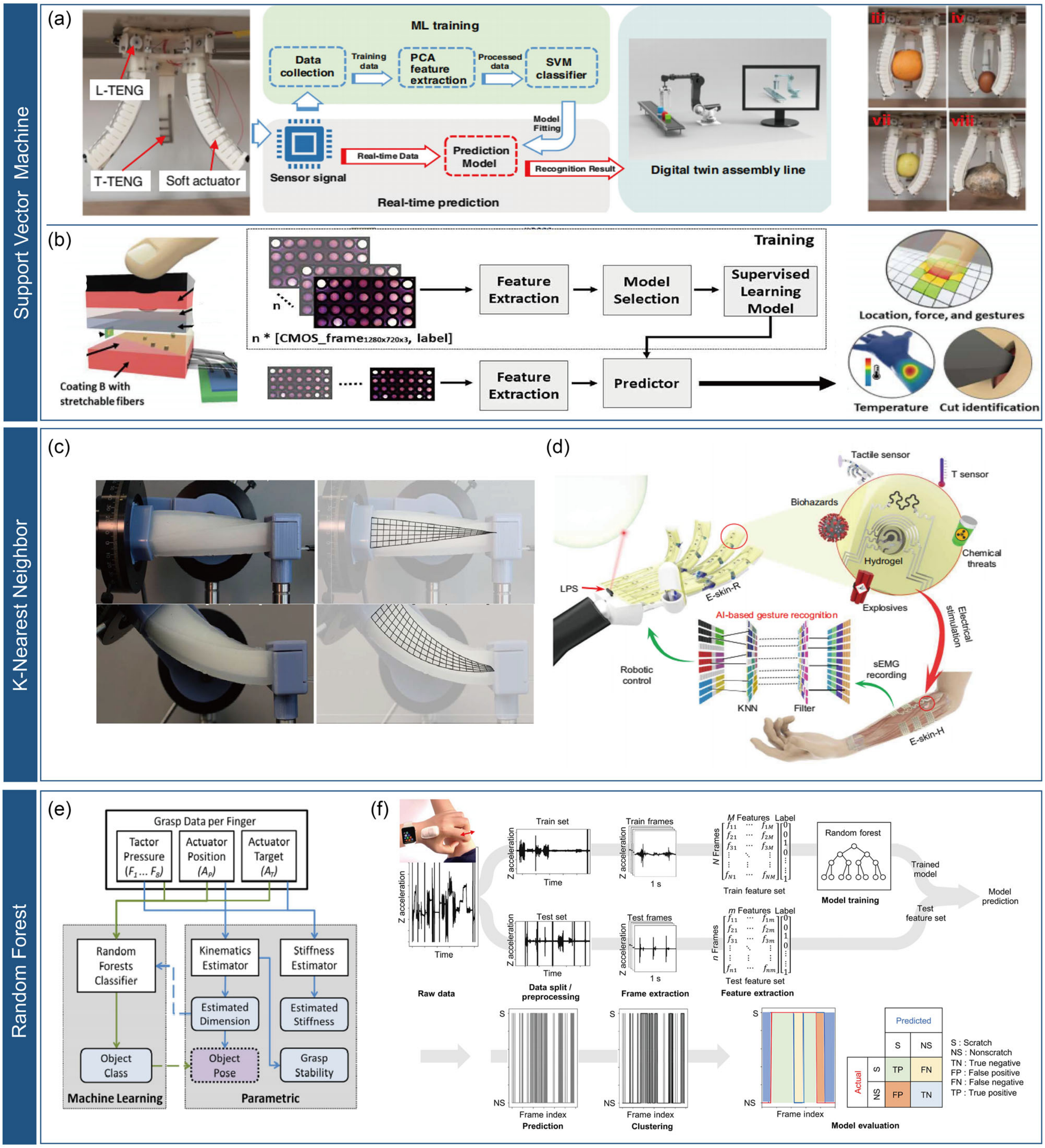

Figure 3. Statistical learning algorithms for tactile sensing. a) Triboelectric-based gripper data processed to classify different objects by SVM. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[44] Copyright 2020, The Authors, published by Springer Nature. b) Optical robotic flesh data with SVM to detect force, touch location, and gestures. Reproduced with permission.[20] Copyright 2022, AAAS. c) KNN-based foam posture recognition for detecting bend and twists and predicting angle. Reproduced with permission.[19] Copyright 2018, AAAS. d) KNN-based gesture recognition. Reproduced with permission.[12] Copyright 2022, AAAS. e) RF was applied in object recognition through grasping action. Reproduced with permission.[100] Copyright 2016, IEEE. f) To quantify symptoms of pruritus, RF was used in scratch prediction. Reproduced with permission.[99] Copyright 2021, AAAS.

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2.2.2. Neural Network and Deep Learning

Artificial neural network (ANN) algorithm simulates the mechanism of signal transmission in mammal nervous systems. The computational units are connected through weights, which serve the same role as the strengths of synaptic connections in biological organisms. Due to the advantages of parallelism, fault tolerance, and automatically extracting features rather than manual selection, deep learning provides a solution to discover the intricate structure of abstract and large data. The first conception of neural network is the artificial neuron. The neuron model (also called unit) simulates the activation and connection work principle of biological neurons. The algorithm is achieved by mathematical methods like functions. When a "neuron" gets an input that reaches a threshold, it will be "activated" through the activation function and send signals to other neurons in the next layer. An ideal activation function is a step function, mapping the input to binary output "0" or "1," representing neuron rest or activated state, respectively. However, due to the characteristic of discontinuity and nondifferentiable of a step function, several continuous functions are applied to replace it in practical applications, such as the sigmoid function, rectified linear unit, and *tanh* function. The artificial neural network computes a function of the inputs by propagating the computed values from the input neurons to the output neurons and applying the weights as intermediate parameters. Learning occurs by changing the weights connecting the neurons. By carefully adjusting the weights between neurons based on many input–output pairs with optimization methods, the function computed by the neural network is refined through time so that it provides more accurate predictions.

A widely used shallow neural network model is multilayer perceptron (MLP), which consists of fully-connected layers applied to do various work in tactile studies.[14,46,51,106,107] MLP is a feedforward artificial neural network. The training process is a process of adjusting connection weights and neuron thresholds. In tactile sensing, MLP is usually applied as an artificial intelligence part for the whole system and has achieved various tasks, such as object classification,[46] gesture recognition,[14,15,107,108] material sensing,[14,46,106] and super-resolution sensing.[22] Wei et al. developed an e-skin and implemented MLP as the main neural network for real-time sensing of materials with indistinct morphology and smooth surface solely via one touch (**Figure 4**a).[106] The system was able to recognize 12 materials with an accuracy of 98.9%, and the perception was displayed in real-time practice. A magnetic skin-like tactile hierarchical sensor was reported by Yan et al.,[22] and they applied a three-layer neural network to achieve super-resolution tactile sensing (Figure 4b). Shallow MLP has been widely used in tactile data processing. Compared with other deep neural networks, it can be embedded and is easy to train, and MLP has shown good performance in real-time tasks.

Inspired by the mammalian visual process neuron structure,[109] CNN was designed to work with grid-structured inputs, which have solid spatial dependencies in local grid regions. The convolution operation places the filter at each possible place in the input grid, so the filter fully covers the input grid (or hidden layer) and performs a dot product. A pooling operation usually follows the convolution layer, which summarizes the features presented in the region and reduces the size of feature maps. Dozens of works have applied CNN as the main classifier or regressor in data processing.[13,17,47,54,55,110–118] Due to the advantage of scalability and low cost, researchers applied piezoresistive sensors for human–environment interaction in many studies with CNNs.[48,55,115] A scalable glove was reported in 2019 that integrated 548 piezoresistive sensors to capture tactile signatures of action when grasping different objects (Figure 4c).[55] Compared with conventional tactile data, data in this research can be considered as large-area mapping. The interaction between the human hand and objects was recorded in large tactile mapping videos of 135 000 frames. Residual architecture ResNet-18[119] was applied to train the whole data set. Similarly, Luo et al. reported a piezoresistive fiber knitted textile to learn human motions, including sitting poses and other motions.[48] The interaction video contained more than a million frames. Figure 4d illustrates the gait recognition achieved by CNN.[112] The 1D-CNN-assisted sock was able to detect the dynamic gait cycle, then identify different participants. 1D-CNN was also applied to recognize different objects as a feature extractor in a TENG-based grasp system.[118] CNN can process series and grid-like data and fits a large amount of data. It is also easily combined with new concepts, like the attention mechanism.[120] Furthermore, the neural network can implement multimodel signal fusion. For instance, Zhu et al. applied CNN to process triboelectric and pressure signals.[45] Before the CNN algorithm, a triboelectric conditioner was used to optimize the signal readout. Then the data fusion was achieved by increasing the input dimension of CNN. Liu et al. used tactile and olfactory data fusion to implement rescue missions by classifying humans from objects.[115] The tactile data set was fed into a CNN, and the olfactory data set was fed into a fully connected neural network. Then two outputs were both processed in fully connected layers. Further discussions about multimodel signal fusion can be found in this review.[67] In the history of CNN, the convolution network was one of the first trained with backpropagation when it was once considered to have failed. As one of the most widely used network architectures, CNN has brought the most successful performance on grid topology and shown the capacity for tactile data processing. The performance benefits from the feature of auto-extraction and efficient sparsity. However, a black box architecture hinders the next step in the development of network structure.

The recurrent neural network (RNN) is a frequently used supervised method that fits sequence data in the tactile signal process procedure. Tactile analog data naturally contain time sequential dependency. Thus, RNN, as a time-related method, is introduced to tactile serial signal process. Serial signals can be processed by machine learning algorithms mentioned earlier, but in practical processing, they are divided into frames because other machine learning methods' parameter-sharing through time is limited and shallow. When the time series is much longer, nonseries specialized methods above easily get failed because they depend on external physical memories to store the historical data. RNN architecture memorizes input messages without external costs like physical memory space. Long-short-term memory (LSTM) is a typical and common model of RNN. LSTM is able to memorize the last output, circulate, update, and adapt through its architecture. An LSTM unit

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

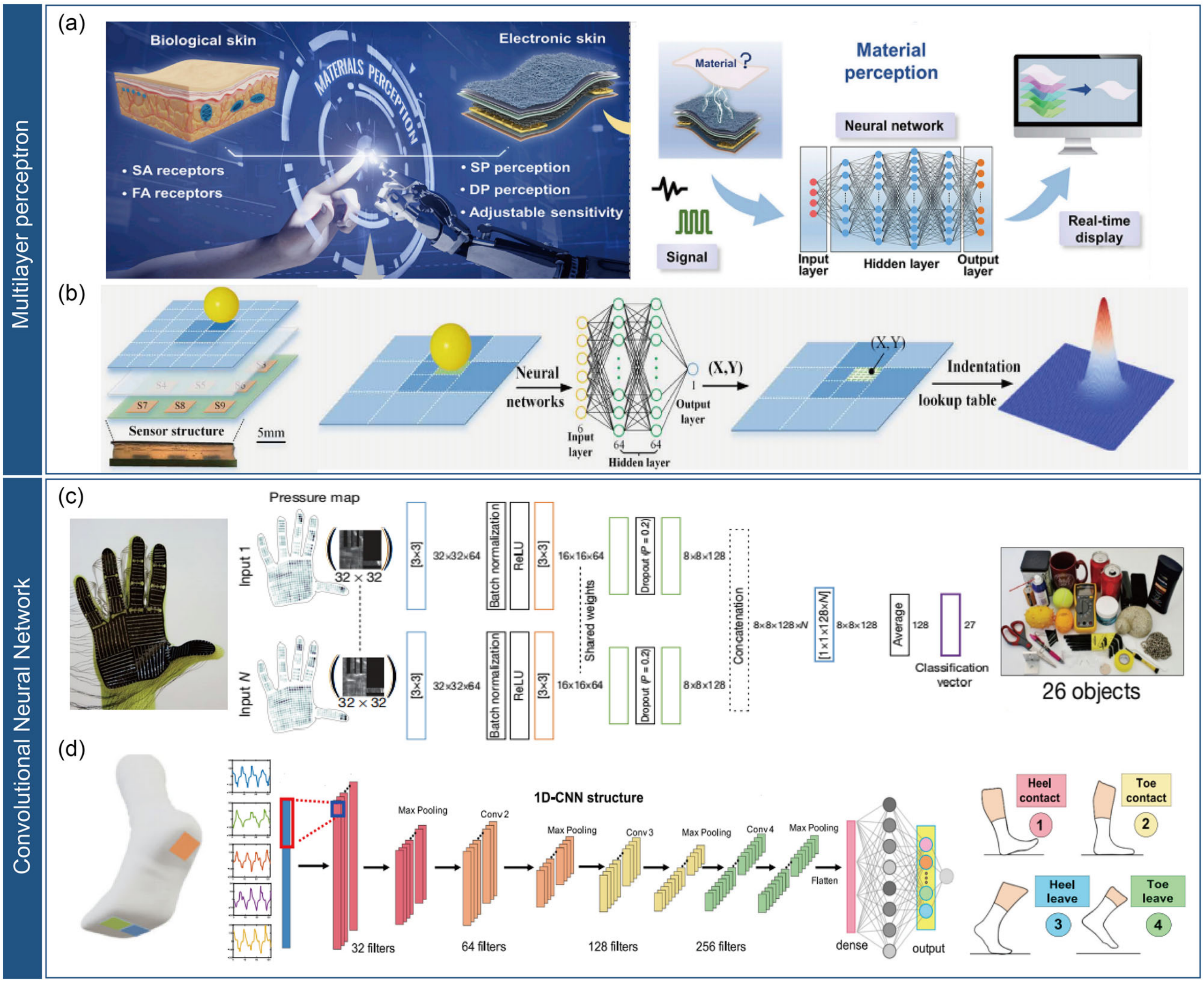

Figure 4. Artificial neural networks for tactile sensing. a) Material recognition with MLP. Reproduced with permission.[106] Copyright 2022, Elsevier. b) Super-resolution achieved by MLP. Reproduced with permission.[22] Copyright 2021, AAAS. c) Object recognition with scalable piezoresistive gloves using CNN as the classifier. Reproduced with permission.[55] Copyright 2019, Springer Nature. d) Gait identification by taking 1D-CNN as the predictor. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[112] Copyright 2020, The Authors, published by Springer Nature.

consists of an input gate, an output gate, and a forget gate (**Figure 5**a).

In Golestani and Moghaddam's research,[21] LSTM was implemented to recognize human activity. The system was based on magnetic induction signals generated from spatially distributed sensors over the human body (Figure 5b). Coupling nonpropagating magnetic field between the body nodes (i.e., wire coils) transmitted signals to the receive coil. Each pair of nodes placed at the joint can define a human bone. Due to the effect of distance and misalignment between coils, the voltage gain of this system and the magnetic-induced signals showed a strong relationship with the geo-translation of body segments. The signal process procedure is shown in Figure 5c. Compared with other machine learning algorithms, the LSTM with optimum hyperparameters outperformed other classifiers by a considerable margin on the generated synthetic magnetic-induced data. The advantages of using deep LSTM architecture in this paper could be summarized as follows: It enables the system to extract discriminative features and support multiple parallel temporal input data from different sensor modalities. The LSTM allows users to fine-tune the model with their data. In other applications, LSTM was also applied in a soft-embedded sensor robot to learn the time series mapping.[52] The time-lagged data needed LSTM for both kinematic and force estimation at the same time. RNN was also applied to do depth estimation[121] and slip detection in tactile-vision fused information.[49]

RNN is known to be turing complete. It means, theoretically, if given enough data and computational resources, RNN can simulate any algorithm. However, it is unrealistic because the requirement of data and resources is too harsh to realize. Furthermore, common issues occur, including gradient vanishing and exploding, and the training process is difficult. These problems increase with the length of time-series getting longer.

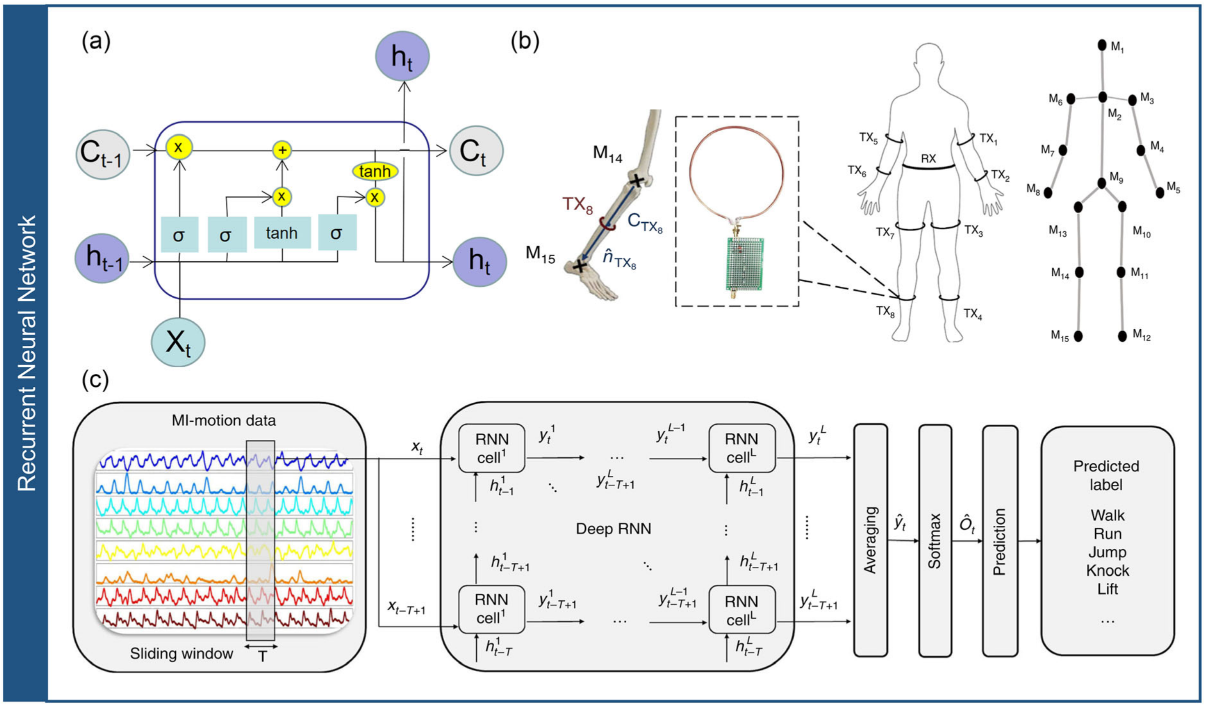

Figure 5. Recurrent neural network for human activity detection. a) Standard LSTM architecture. b) Magnetic sensor distribution on human body to collect signals for RNN. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[21] Copyright 2020, The Authors, published by Springer Nature. c) Signal processing procedure. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[21] Copyright 2020, The Authors, published by Springer Nature.

Thus, the usage of RNN is waiting for more appropriate and well-designed strategies in tactile analog signal process.

3. Machine Learning for Spike Signals

Unlike analog signals that are clock-based and synchronous, spike signals are event-based and asynchronous. Thereupon, compared with analog signals, spikes signals are appealing with the advantages of low power consumption and fast speed.[39,122] The high power efficiency of spike signals provides an exciting opportunity for edge computing, which is essential for tactile sensors to become edge devices for broad applications.[123] The low latency of spikes allows tactile sensors to perceive rapid changes and builds a basis for fast responses to those stimuli. Besides, spike signals have a physiological basis, which allows the imitation of biological signals and has great application potential in areas like physiology, neural prosthesis, and neural robotics.[124–128] Therefore, the increasing interest in spike signals has prompted the development of spike acquisition from tactile sensors, and the rise of tactile sensor spikes has driven the research in the corresponding data processing using machine learning.

3.1. Acquisition of Spike Signals

Though existing attempts to build optical tactile sensors with event-based cameras for spike signals,[129] most tactile sensors still produce analog signals. Nevertheless, some researchers have managed to obtain spikes by converting analog data from the analog tactile sensors, to utilize the advantages of spike signals like fast response and low power consumption, or to explore the applications in neural prosthesis, neural robotics, and spike-signal-related areas. Typical techniques are connecting with synaptic electronic devices, employing biological neuron models, and applying characteristics encoding strategies.

Synaptic electronic devices connect synaptic electronic components with tactile sensors to generate spike signals. Memristors are one popular kind of artificial synapse. Researchers used memristors in series with tactile sensors to transform analog data into spike signals (**Figure 6**a).[32,33,35,36] It is worth mentioning that the memristors used in the works[33,35] were temperature dependent, and the researchers both achieved data fusion of temperature and pressure into output spikes for multimodal tactile perception. Besides memristors, researchers also used synapse transistors to create spike signals for further processing.[34,128,130,131]

The biological neuron model approach originated from neuron biological activities, which emit spikes when hidden membrane voltages exceed a defined threshold. These models take analog inputs as currents and charge the membrane voltages with selected neuron parameters based on their equations. After spike emission, the neuron models have resetting on the voltages. Izhikevich neuron model,[132] with its reasonable biological plausibility, efficient computational cost, and high neuron dynamics, has been used commonly to transform analog tactile signals into spikes.[127,133–140] Integrate-and-fire model[141,142]

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

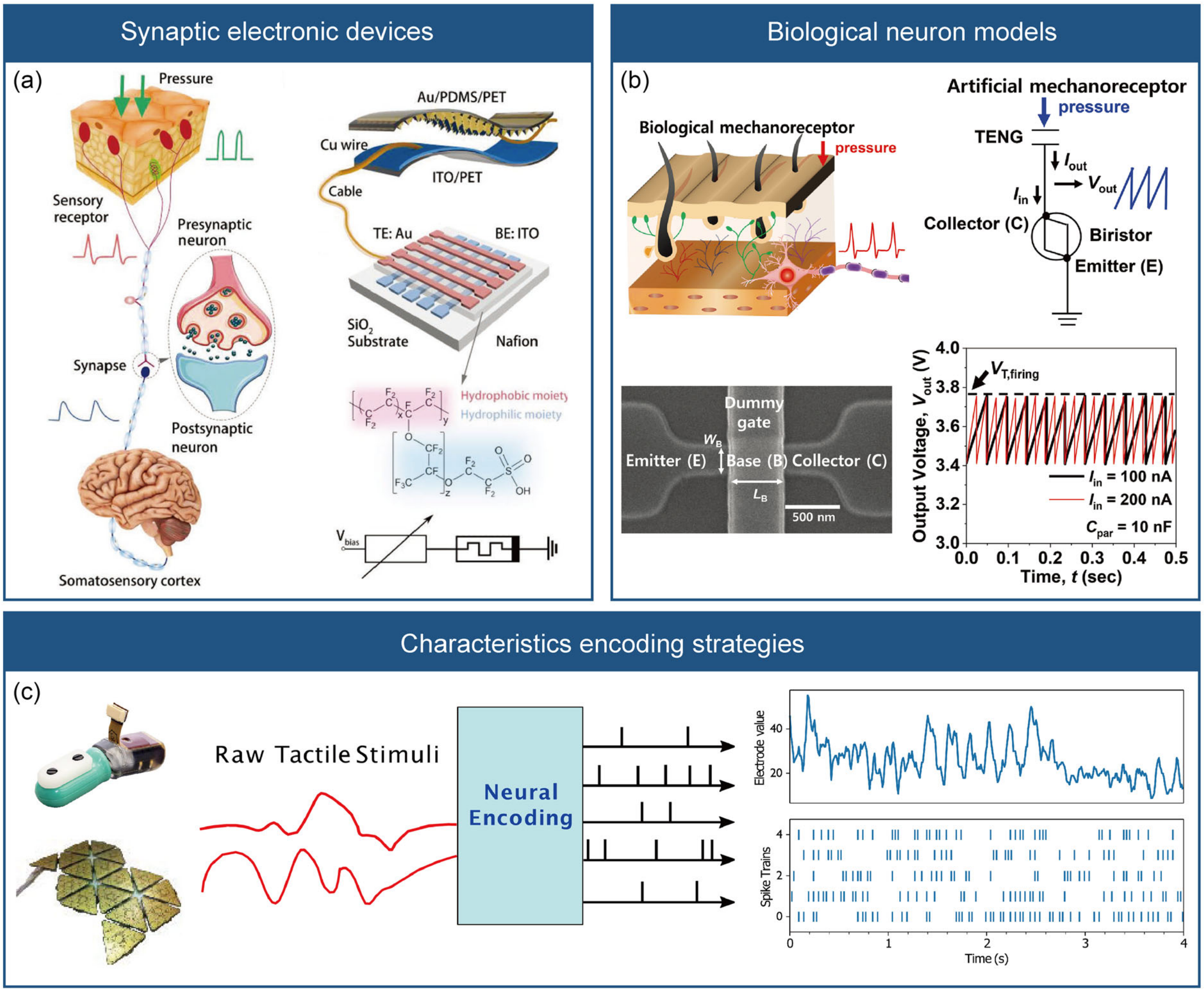

Figure 6. Convert analog signals to spikes. a) Synaptic electronic devices. Reproduced with permission.[32] Copyright 2019, Wiley-VCH. b) Biological neuron models. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[123] Copyright 2022, The Authors, published by Wiley-VCH. c) Characteristics encoding strategies. Reproduced with permission.[144] Copyright 2020, IEEE.

and spike response model,[143] with more efficiency on computation cost than Izhikevich, have also been applied to signal conversion,[123,144–149] although they have poor biological plausibility. Usually, biological neuron models are implemented at a software level, whereas a biristor neuron was fabricated with a triboelectic nanogenerator sensor to mimic the integrate-and-fire function (Figure 6b).[123] Beside the above mentioned ones, more neuron models could be found in these papers[132,143] for readers' reference.

The characteristics encoding technique is more accessible to implement than synaptic electronic devices or biological neuron models because this method needs no extra electronic components or careful neuron parameter selection. A threshold encoding strategy was employed to encode raw data from BioTac and RoboSkin sensors into spikes for further texture classification tasks (Figure 6c).[144] On the basis of a spiking neural network, the work achieved 94.6% accuracy with the BioTac sensor data and 92.2% accuracy with the RoboSkin data. Readers may refer to the following review materials for more encoding strategies like rate and time encoding.[150–152]

3.2. Spike-based Machine Learning

Unlike analog signals, spikes are asynchronous, relatively sparse, and generally memory-based. Thereupon, traditional machine learning algorithms are not appropriate for processing spike signals if researchers want to take advantage of these signals to the fullest. Many researchers have conducted studies from various directions, trying to deal with spike signals as human brains do. Among those attempts, a spiking neural network (SNN) is one popular research field and dominant technique to process spike signals under the scope of machine learning, with the advantages of temporal coding processing, power efficiency, and fast response. SNNs have the same network structures as the NNs mentioned above, while neurons in SNNs are biological neurons, not nonlinear but continuous function approximators in artificial neural networks. With different neurons, training in SNNs is unlike training in artificial neural networks, which generally use gradient descent for backpropagation. The training schemes of SNNs can be roughly categorized into three groups: local learning at synapses, learning by error backpropagation, and ANN-to-SNN conversion.

3.2.1. Local Learning at Synapses

Local learning at synapses is relatively biologically realistic training,[39] which updates weights based on spatially or temporally local signals by applying local learning rules. A typical local learning rule is spike-timing-dependent plasticity (STDP). STDP is a biologically plausible synaptic learning rule, which changes synaptic connections on the basis of relative spike timing of presynaptic and postsynaptic spikes using the STDP function.[153,154] An example of STDP in processing tactile information is the work of Bucci et al.[133] Bucci and his colleagues designed a neuromorphic robot that interacted through touching sensing by an array of 16 trackball sensors and visual signaling on its surface. To decode the tactile sensory data, they constructed an unsupervised SNN of two bionic groups of neurons. One group included 256 thalamic input neurons, and the other contained 680 somatosensory cortex neurons. The somatosensory cortex neurons had 544 excitatory neurons and 136 inhibitory neurons. Each somatosensory cortex neuron received 100 inputs that were randomly selected from the thalamic input neurons and other somatosensory cortex neurons. The neurons took Izhikevich neuron models as the dynamics. The output connections of the excitatory neurons were subject to STDP. Input patterns of left and right moves across the robot were fed into the SNN for training. Then the researchers tested the SNN using the additional left and right moves. The algorithm performed flawlessly as the areas under the receiver operator characteristics (ROC) curves approached 1. The analysis of the correlation of neuron firing activity between movements also demonstrated that similar classes of hand movements were highly correlated after training.

Usually, STDP learns unsupervised, while tempotron, a variant of STDP, is a supervised learning rule for classification problems.[155] For classification tasks with two classes, the task of the tempotron is to respond to an input pattern from the positive class by emitting at least one spike, and the tempotron stays quiescent when faced with an input pattern from the negative class. Tempotron takes a leaky integrate-and-fire neuron as the neuron model. The algorithm learns to conduct the task by a rule that depresses synaptic efficacies when the synapses contribute to a wrong output spike on a negative pattern and potentiates the efficacies when they fail to spike on a positive pattern. Navaraj and Dahiya utilized the tempotron classifier system to discriminate between the loop and hook textured surfaces (**Figure 7**a).[145] They designed a tactile-sensing architecture composed of a floating electrode-based capacitive structure and a piezoelectric structure and covered this susceptible sensing skin with fingerprint-like patterns. They attached the biomimetic tactile sensor to the end effector of a UR5 industrial arm and collected signals by sliding on the textures that were wrapped on various surfaces. Two hundred scans for each texture were recorded, with 160 scans for training and 40 for testing. Due to its ability to capture temporal variation and biologically plausible wavelets, a windowed Gabor wavelet transform was used to process the tactile information. An integrate-and-fire neuron model was utilized for encoding the normalized wavelet amplitudes to get spikes. Then, the spike trains were given to the tempotron neurons as inputs. Two tempotrons were trained to detect hook and loop textures separately using a biologically plausible STDP algorithm that aimed at eliciting a spike output for matching target classification. The synaptic efficacy of each tempotron was modified when the tempotron as the target category emitted no output spike. This approach offered 99.45% accuracy in distinguishing hook texture and loop texture.

Local learning at synapses usually focuses on dealing with small samples of data and simple classification tasks. It is attractive in tactile sensing with its ability of detecting spatio-temporal patterns and the potential for high hardware-efficient training.[39] However, it faces the problem of convergence when learning on a complex data set with a large-scale model. Besides, it strongly relies on network structure to achieve outstanding performance.[39] Nevertheless, with its root in biology, local learning at synapses is intriguing for researchers who seek to process tactile information as animals do.

3.2.2. Learning by Error Backpropagation

When dealing with more complex tasks, researchers prefer to employ the approach of learning by error backpropagation. Because spikes are undifferentiated, conventional backpropagation using gradient descent is not applicable for SNN. Adapted training approaches have been proposed to solve this; the surrogate function method is commonly used among those approaches. The surrogate function method replaces the spike function with a differentiable surrogate function which removes the barrier for error backpropagation based on gradient descent. In tactile information processing, the sigmoid activation function is a popular surrogate function for weight updating in SNN training due to its spike-like curve and differentiable property. Han et al. used the sigmoid activation function as the neuron spiking function for weight updating to get backpropagation gradients.[123] A self-powered artificial mechanoreceptor was proposed with a TENG for pressure sensing and energy harvesting. A biristor functioned as an integrate-and-fire neuron to encode spike signals. The researchers built a three-layer SNN that contained 784 input artificial mechanoreceptors, 100 hidden neurons, and ten output neurons for pattern recognition simulation. Backpropagation was utilized for training the SNN with the sigmoid activation function as the surrogate function. The pixel intensity of the MNIST data set[156] was projected to the tactile perception, and the SNN was trained for nine epochs. A classification accuracy of 85.8% was obtained. Then, they implemented hardware breath monitoring by putting a wind pressure measuring TENG near a subject's nose and a bending pressure measuring TENG on the abdomen. A spiking neural network was designed for classifying exhalation and inhalation. It was composed of an input layer with two artificial mechanoreceptors corresponding to the two sensors and an output layer corresponding to the two states. The system observed spiking in one output neuron for exhalation and another for inhalation and accomplished the classification task.

Besides the surrogate function method, another method, Spike LAYer Error Reassignment (SLAYER), was also a fashion

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

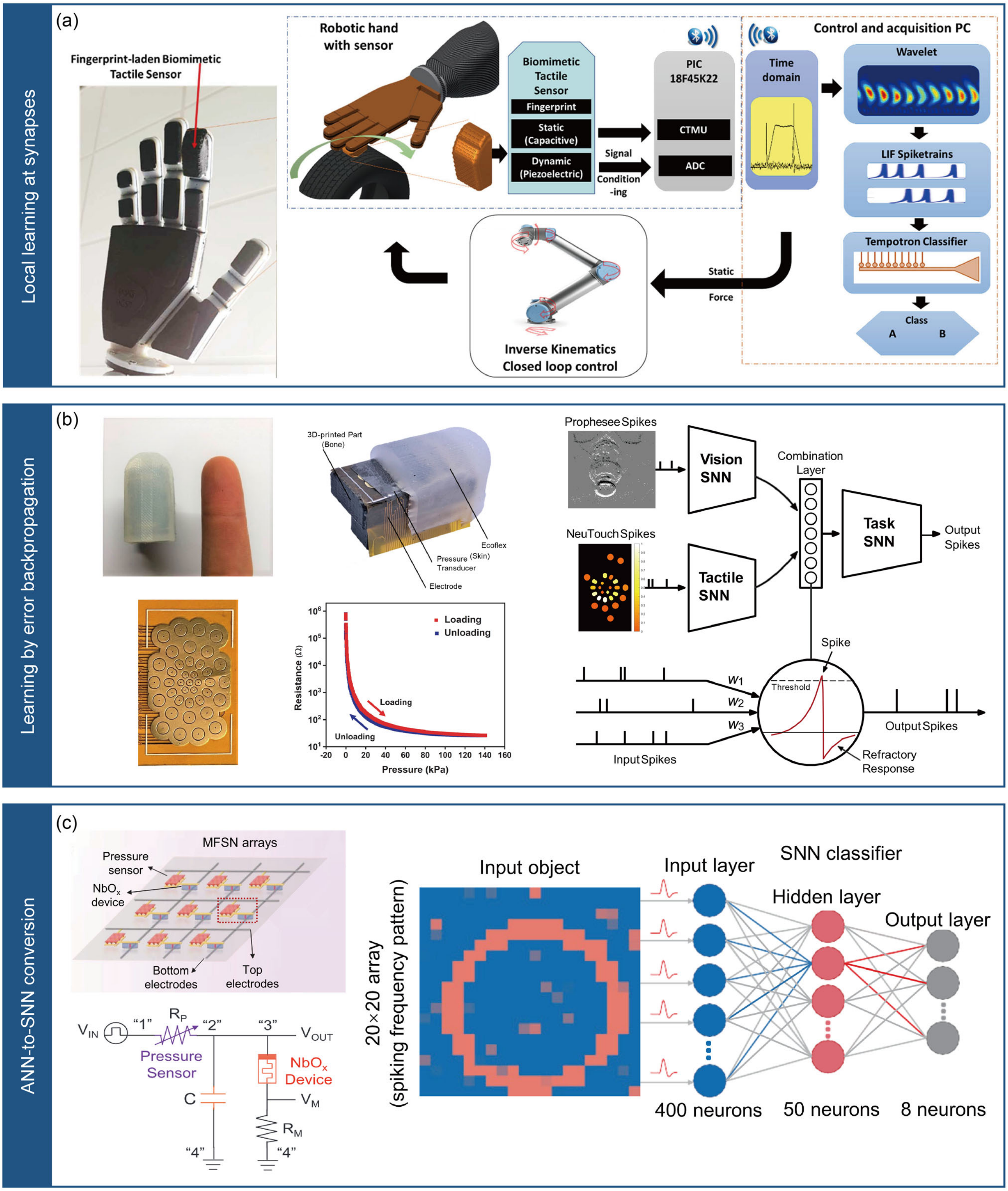

Figure 7. Training schemes of SNNs. a) Local learning at synapses. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[145] Copyright 2019, The Authors, published by Wiley-VCH. b) Learning by backpropagation. Reproduced with permission.[148] Copyright 2020, The Authors, published by Robotics: Science and Systems. c) ANN-to-SNN conversion. Reproduced with permission.[33] Copyright 2022, Wiley-VCH.

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

in training SNNs for tactile information processing. SLAYER was invented by Shrestha and Orchard.[157] It is a learning algorithm with the superiority of considering temporal dependency between input and output signals of a neuron, handling the non-differentiable issue of spike function, and not prone to the dead neuron problem. Taunyazov et al. collected data from BioTac and RoboSkin sensors by sliding on 20 material textures and encoded the data into spike trains using the thresholding technique.[144] Using the Spike Response Model as the neural model, the researchers trained with the SLAYER algorithm and achieved 94.6% accuracy on a BioTac data set and 92.2% on an iCub data set with a short inference time. SLAYER was also used to train an SNN for fusing visual and tactile data by Taunyazov et al. (Figure 7b).[148] The researchers employed Prophesee Onboard for vision and NeuTouch for tactile sensing. They mounted the sensors on a Franka Emika Panda robotic arm and used the arm to grasp four containers with five different weight levels. Training with Spike Response Model and SLAYER, the researchers achieved 81% accuracy in classifying these twenty object classes with combined sensor data. Taunyazov and his colleagues also tested the rotational slip classification capacity, and the algorithm achieved 100% accuracy in detecting slips within 0.001 s.

With researchers' continuous efforts, learning by error backpropagation in SNNs can achieve comparable accuracy in dealing with tactile data compared with conventional ANNs and have a shorter inference time than traditional ANNs. Though it waits to see whether this method is an optimal choice to train SNNs, learning by error backpropagation is still the mainstream of training schemes, with its strong capacity in tuning network parameters and successful experience in ANNs.

3.2.3. ANN-to-SNN Conversion

ANN-to-SNN conversion is an indirect supervised learning approach for developing SNNs. By learning in ANNs and converting the learned ANNs to SNNs, the method avoids the difficulty of direct training in SNNs. There are requirements for both ANNs and SNNs to enable the conversion. For example, because neurons in SNNs have only non-negative firing rates, ANNs should have non-negative activation neurons, like neurons with rectified linear-unit (ReLU) activation functions, or SNNs need to have two spiking neurons for each ANN neuron to cover positive and negative activation parts separately.[158] Another limitation is that max-pooling operations, a common data-processing technique in ANNs, are hard to realize in SNNs[159] because the maximum operation is nonlinear and cannot be calculated on a spike basis.[39] Even though some constraints exist, the ANN-to-SNN conversion is appealing, as advances in ANNs can be applied to SNNs directly using the conversion approach.[37] Zhu et al. published a paper in which they used ANN-to-SNN conversion to learn a spiking neural network for enhanced pattern recognition (Figure 7c).[33] Zhu and his colleagues integrated a 3 × 3 array of multimode-fused spiking neurons containing a pressure sensor to process pressure and an NbO_x-based memristor to sense temperature. With the spiking neurons, they fused multisensory information into one spike train. They simulated a 20 × 20 array of neurons to collect data for classifying eight cups of different shapes, temperatures, and weights and obtained 800 data samples. An SNN of 400 input neurons, 50 hidden neurons, and eight output neurons was built. The neural network was trained in ANN mode with the ReLU activation function, and the weights were updated by error backpropagation using gradient descent. After training of 250 epochs, the ANN was converted into the SNN with linear leaky-integrate and fire (LIF) neurons. The researchers tested the SNN with linear LIF neurons and found the classification results by finding the first firing neuron in the output layer. The SNN achieved 93% accuracy in classifying the eight cups with fused information.

Although there still exist constraints for neural networks and training tools, and this method fails to meet the promise of low-power inference,[39] ANN-to-SNN conversion avoids the difficulty of direct training of SNNs. It achieves a remarkable performance, which makes this approach a strong candidate for spike-based machine learning.

3.2.4. Extreme Learning Machine

Besides the three training schemes of spike-based machine learning, an alternative to train a spike-based neural network is to use an ELM. ELM works for generalized single-hidden layer feedforward networks,[160] where input weights and biases are randomly chosen and fixed after initialization. ELM adapts only output weights by multiplying the inverse of the hidden layer output matrix with the target matrix, which avoids computing the gradients. After training, the classification decision is determined by the output neuron of the highest activation value. An attempt to utilize the ELM to train SNN has been made by Rasouli et al.[137] Rasouli and his colleagues fabricated a biomimetic tactile sensor array comprising ridge-shaped structures with piezoresistive material sandwiched between two conductive layers. They slid the fingertip equipped with the sensors on ten different textures and obtained analog signals. Izhikevich neuronal model was utilized to convert the tactile signals into spikes. Then a window counter counted the number of spikes in a moving time window as the analog features. A total of 128 features functioned as the inputs. With 128 hidden layer nodes and ten output nodes, the system could predict the correct texture label with a precision of 92% after training.

3.3. Convert into Analog Features

Researchers sometimes extract analog features from spikes to analyze the signals for ease of convenience because spike-based machine learning is still underdeveloped. In some cases, it is still tedious to build a spike-based machine learning model. Given spike signals, this conversion approach extracts analog features from a spike sequence of a certain time duration and then applies the above-mentioned analog-based algorithms (e.g., SVM, KNN, and MLP) for application tasks. Researchers have tried different feature extraction methods on spike sequences, and typical methods are single-channel characteristics extraction, multiple-channel distance computation, and unsupervised learning feature extraction.

Single-signal characteristics extraction extracts feature by selecting specific signal characteristics and computing the relative characteristics of a single signal channel. Average spike rate,

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

sometimes referred to as spike frequency, spike counts, or firing rate, is a common signal characteristics value used in machine learning for tactile spike analysis.[35,127,129,135,161] The frequencies of spike signals from a serial connection of piezoresistive sensor and VO_2 volatile memristor were combined with minimum and maximum spike voltage values to construct inputs of a three-layer MLP (**Figure 8**a).[35] This work collected the signals under 11 situations of different pressures and temperatures and trained the MLP with spike features for 200 epochs to classify these situations, which attained an accuracy of 91.35% after training. Sankar et al. integrated a 3 × 3 flexible textile neuromorphic tactile sensor array on a soft biomimetic finger to palpate thirteen textures and used average spike rates and average interspike intervals as features to feed into SVM with a linear kernel.[127] Performing a *k*-fold cross-validation procedure with *k* equaling to four, the work achieved an overall accuracy of 99.62% in classifying different textures.

As for quantitatively comparing multiple signal channels for machine learning, researchers usually use spike train distance as the analog feature. Spike train distance, also known as spike train synchrony, is a measurement of similarity between spike trains and is usually used to replace conventional Euclidean distance in the KNN algorithm. The researchers in ref. [162] selected Victor-Purpura distances[163] as features for a KNN with data collected from an array of 69 fast-adapting receptors on Asynchronously Coded Electronic Skin. They demonstrated a task on classifying local curvatures and hardness and accomplished a 97% accuracy in a short duration of less than 7 ms. Multineuron Victor-Purpura distance[163,164] was used in the work of Yi et al. (Figure 8b).[139] They slid a biomimetic polyvinylidene difluoride fingertip on surfaces of different roughness and converted the tactile signals to spike trains with the Izhikevich neuron model. Then, they computed multineuron Victor-Purpura spike train distances and applied a KNN algorithm for surface roughness categorization. The algorithm had a performance of 77.25% accuracy in classifying eight levels of roughness. In addition to Victor-Purpura distance, there exist other ways to define spike train distance, and more introductions and comparisons of spike train distances can be found in these reviews.[165–167] A comparison between the two feature extraction methods was conducted by Rongala et al. They compared single-channel characteristics and multiple-channel spike train distances as feature selection methods for naturalistic texture categorization.[135] They utilized a MEMS piezoresistive sensor to collect data on ten different textures. With the Izhikevich algorithm, they transformed the signals into neuromorphic spike outputs. They chose spike rate and

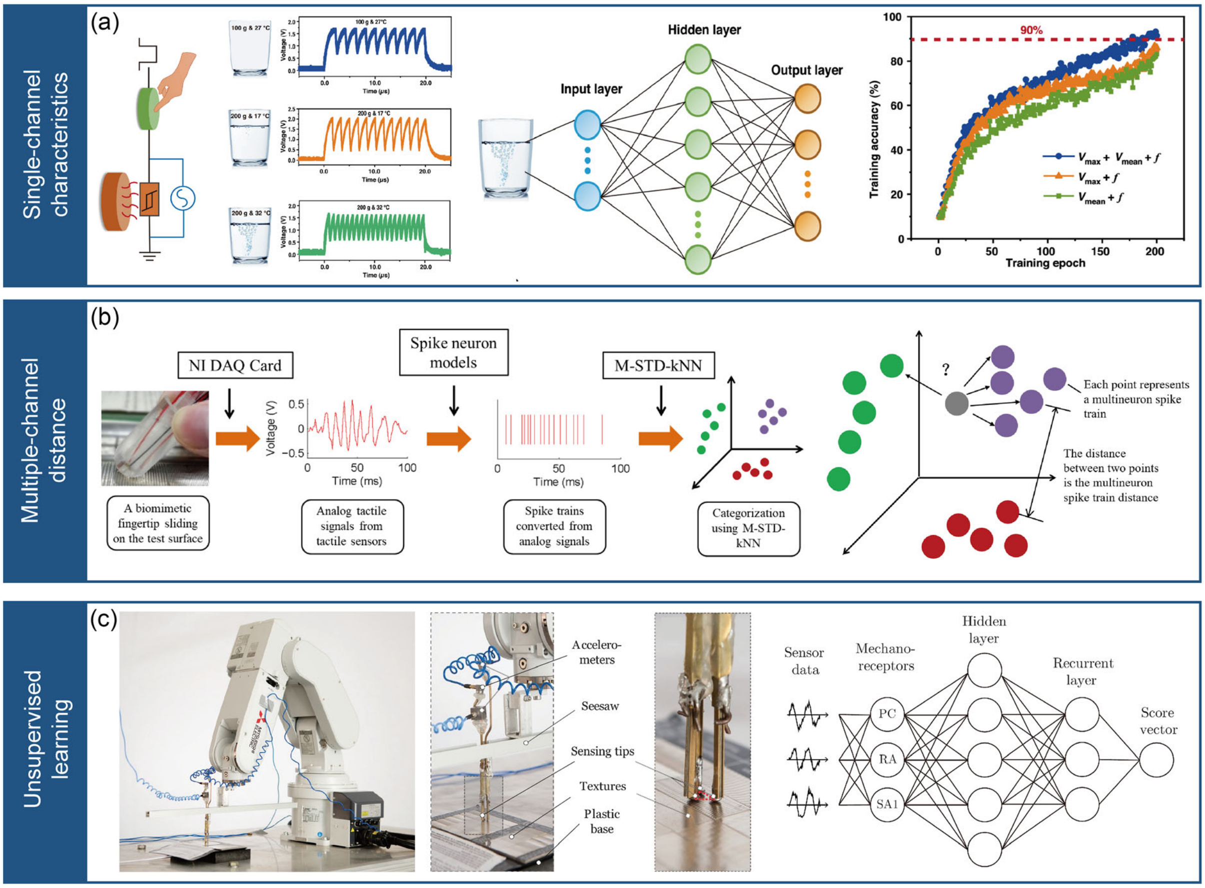

Figure 8. Extract analog features from spikes. a) Single-channel characteristics extraction. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[35] Copyright 2022, The Authors, published by Wiley-VCH. b) Multiple-channel distance computation. Reproduced with permission.[139] Copyright 2021, IEEE. c) Unsupervised learning feature extraction.[149] Reproduced with permission.[149] Copyright 2016, IEEE.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

coefficient of variation of the interspike interval as signal features for KNN and obtained an overall precision of 78%. As for spike train distance, the researchers calculated Victor-Purpura distances and decoded spike stimulus using KNN with 93% accuracy. The results of this work indicated that Victor-Purpura distances carried some valuable information that was not present in the spike rate and variation of the interspike interval.

Besides single-channel signal characteristics and multichannel spike train distance computation, unsupervised learning is applied to extract spike signal features. One attempt was made by Friedl et al. Friedl et al. used a semi-supervised approach for spike signals, which trained an unsupervised two-layer SNN for feature extraction and supervised SVM for texture classification (Figure 8c).[149] They mounted three piezoceramic acceleration sensors and two one-axis PCS Piezoelectrionics C65 on a robotic arm and collected signals by sensing tips through sliding on surface textures. They employed adaptive leaky integrate-and-fire neuron models to obtain spike trains and used these to train an unsupervised SNN with leaky integrate-and-fire neurons for features. Then classification was conducted with SVM, and an overall precision of 65.6% was obtained in determining eighteen surface textures.

Extracting analog features of spike trains allows the use of analog-based machine-learning techniques to deal with spike signals. This provides researchers with a choice to utilize spike signals while not being bothered by the hard work of training spike-based models. However, there exists information loss when extracting features, and this approach sacrifices spikes' advantages of low-power consumption and fast responsive time.

4. Applications of Machine Learning for Tactile Sensing

As previously indicated, the four main tasks used in tactile sensing are classification, regression, clustering, and dimension reduction. Given tactile data, researchers select machine learning algorithms based on their purposes. Applications of these tasks and related algorithms are summarized in **Table 1** and **2** for analog signals and **Table 3** for spike signals.

Table 1. List of unsupervised learning algorithms applied in tactile sensing for analog signals.

Algorithm	Task	Application	References
K-means clustering	Clustering	Data redundancy minimization	[55]
–	–	Data cleaning	[42]
Principal component analysis	Dimension reduction	Feature extraction	[44,45,55,60,64,77,94,103]
–	–	Data visualization	[17,48,78]
Dictionary learning	Dimension reduction	Feature extraction	[60,82,83]
Autoencoder	Dimension reduction	Feature extraction	[61–63,85]

4.1. Applications of Supervised Learning

Supervised learning processes data with labels, and classification and regression are the main tasks. Classification addresses problems with discrete labels, whereas regression addresses issues with continuous labels. Tactile sensors have been applied widely in human activity monitoring and robot tactile perception. Machine learning, the upgraded signal processing, provides tactile sensors with a broader prospect in these topics.

4.1.1. Supervised Learning in Human Activity Monitoring

Classification and regression have been widely targeted as tasks among tactile sensing in applications on human activity monitoring and solved by supervised learning due to their robustness, efficiency, and effectiveness. Among all human activities, hand gesture has received the most attention with tactile sensors mounted on hands or arms. Gesture classification is a prevalent demonstration task to verify tactile sensors' ability to detect hand motions, like the bending of knuckles. This task has been solved by supervised learning in many works using sensors on hands or tactile gloves (**Figure 9**a).[14,17,55,77,128,146] Sensors on arms are also capable of monitoring the motions of hands because the movement of hand joints is associated with mechanical signals of arm skin deformation.[168] Therefore, attempts with tactile sensors on arms have also been made to classify hand signs (Figure 9b).[12,15,95,107] Moreover, regression of bending fingers and corresponding angles have been proposed with sensors attached to arms (Figure 9c).[110] Arm movement monitoring, such as arm pronation/supination and metacarpophalangeal angle estimation, has also been regressed using arm strain gauge sensor sleeves.[107] Interactions through hands have also been monitored using supervised learning with tactile sensors on hands, for example, classifying touched patterns,[34] recognizing grasping objects,[45] estimating the weights of grasping objects,[55] and distinguishing hand scratching activities.[99] Hand motion, and even arm motion, with the flexibility and diversity, would definitely continue being a hot application of tactile sensors. Moreover, the monitored information could be further used in translating sign languages,[77] controlling robots (Figure 9d),[12] virtual reality, and other scenarios.

Besides hand motion monitoring, gait analysis is another popular topic in human activity monitoring. With sensors on socks or insoles, supervised learning has been applied to accomplish the tasks of human activity classification (Figure 9e),[112] participant recognition,[112,169] and human pose prediction using regression.[48] Leg movement sensing with a wearable system was also proposed to estimate energy expenditure with a linear regression model.[105] Attempt to analyze gait using sensors on mats has also been proposed (Figure 9f),[170] and the work has accomplished the tasks of user recognition and walking pass status classification.

Other activities have also been researched and monitored with supervised learning, such as breath classification with sensors on the nose and abdomen,[123] eye vergence recognition with sensors on the face (**Figure 10**a),[90] respiration pattern recognition with a on-mask sensor network (Figure 10b),[171] speech

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 2. List of supervised learning algorithms applied in tactile sensing for analog signals.

Algorithm	Task	Application	References
Random forest	Classification	Object classification	[100,101]
–	–	Human scratching activity detection	[99]
Linear discriminant analysis	Classification	Human action recognition	[15,91,94]
–	–	Object classification	[18,100]
–	–	Slip detection	[50]
Naive Bayes	Classification	Speech recognition	[91]
–	–	Object classification	[100,101,103]
Linear regression	Regression	Human movement energy expenditure estimation	[105]
–	–	Sensor deforming angle estimation	[19]
Support vector machine	Classification	Object classification	[44,93,100,174]
–	–	Human action recognition	[45,77,89–91,94]
–	–	Sensor deformation detection	[19]
–	–	Touch location detection	[20,93]
–	–	Slip detection	[50,93]
	Regression	Contact estimation	[20,93]
–	–	Sensor deforming angle estimation	[19]
K-nearest neighbor	Classification	Sensor deformation detection	[19]
–	–	Human action recognition	[12,90,94,95]
–	–	Slip detection	[50]
	Regression	Sensor deforming angle estimation	[19]
Decision tree	Classification	Sensor deformation detection	[19]
–	–	Slip detection	[50]
–	–	Eye vergence detection	[90]
	Regression	Temperature estimation	[20]
–	–	Sensor deforming angle estimation	[19]
Multilayer perception	Classification	Object classification	[14,46,100,106,175]
–	–	Human action recognition	[14,107,169]
–	–	Sensor deformation detection	[173]
–	–	Slip detection	[93]
–	Regression	Sensor deforming angle estimation	[19]
–	–	Contact estimation	[22,51,54,93]
–	–	Human arm movement estimation	[107]
Convolutional neural network	Classification	Object classification	[45,47,48,55,115,116]
–	–	Contact force estimation	[54]
–	–	Human action recognition	[13,17,55,95,111–113,117,170,171]
	Regression	Object weight estimation	[55]
–	–	Contact estimation	[114,172]
–	–	Human joint angle estimation	[48]
Recurrent neural network	Classification	Object classification	[93,121,179]
–	–	Human action recognition	[17,21,110]
–	–	Slip detection	[49,93,174]
–	Regression	Finger bending angle regression	[110]
–	–	Sensor kinematics estimation	[52]
–	–	Contact estimation	[52,93,114]

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 3. List of machine learning algorithms applied in tactile sensing for spike inputs.

Algorithm	Task	Application	References
STDP learning rule	Clustering	Pattern recognition	[133]
Unsupervised SNN	Dimension reduction	Feature extraction	[149]
Tempotron	Classification	Texture classification	[145]
SNN with surrogate function	Classification	Breath classification	[123]
SNN with SLAYER	Classification	Object classification	[144,148]
–	–	Slip detection	[148]
ANN-to-SNN conversion	Classification	Object classification	[33]
SNN with ELM	Classification	Texture classification	[137]

recognition with face tattoo-like sensors (Figure 10c),[91] and action recognition with sensors on human bodies (Figure 10d).[21,117]

Human activity monitoring has been a prevalent topic of tactile sensors with excellent prospects. The data processing using machine learning undoubtedly helps decode the tactile signals and monitor the corresponding human motions, which underpins the enormous promise of tactile sensors. With decoded information, tactile sensors can be further applied in human intention estimation, teleoperation, virtual or augmented reality, internet of things, smart home or environment, healthcare, and other scenarios. The research in this field holds great industry potential.

4.1.2. Supervised Learning in Robot Tactile Perception

Besides working for human activity monitoring, supervised learning has been commonly applied in robot tactile perception with classification and regression tasks. Perception of tactile stimulus is essential for robots, and supervised learning enables a more accurate estimation of force intensity and contact position. Contact force intensity has been estimated using regression learning (**Figure 11**a),[20,52,54,114] and so has force contact position.[22,54,93,172] In addition, the relative orientation of tactile sensor to the contacted object surface has also been regressed.[172] Touch location can also be achieved using classification learning, with each class representing an area of contact positions. An example was proposed in the work of Barreiros et al.,[20] which divided the sensor into square segments and classified the contact information into the related locations.

In addition, proprioception of tactile sensors has been studied and presented. Examples include sensor deformation mode classification (Figure 11b),[19,173] sensor deformation angle regression,[19] and sensor-mounted robot kinematics estimation.[52]

Perception of objects is an important research field in robot cognition, and tactile sense enables robots to get to know some physical properties of objects for manipulation or interaction. Textures are one physical property often identified through supervised learning algorithms by tactile sensors on robots' end effectors, where signals were collected by robots sliding on textured surfaces or pressing on the materials.[18,127,129,135,137,144,145,149,174,175] Roughness recognition has also been implemented to validate the ability of tactile discrimination with biomimetic fingertip sensors.[136,139,140] Hardness and local curvature were classified with supervised learning to verify the tactile perception ability of sensors.[162] Material classification is another popular kind of perception task accomplished with tactile perception (Figure 11c).[14,106,116] Tactile sensors using a supervised learning algorithm have also demonstrated the ability to handle object classification, an integrated task for object perception, with signals collected by robots interacting with objects.[33,35,44,46,47,93,100,147,148,161] Other attempts on object perception tasks, like edge orientation classification[138] and foreign object depth estimation,[121] have also been made using supervised learning.

Moreover, tactile perception is significant for robots when robots conduct manipulation actions on objects. A typical classification task is slip detection, which has been researched for stable robot grasp.[49,50,93] Shear force direction classification was also presented as a machine learning object for a better grasp.[161]

Furthermore, applications of human–robot interactions have been introduced with supervised learning, such as classifying human tapping actions,[111] identifying hand movement directions,[133] and recognizing hand-written characters.[32,36,131]

Tactile perception enables robots with tactile sensing to better reasoning of and reaction to outside dynamic changes. With machine learning, tactile sensors on robots could achieve an enhanced perception field and precision that exceeds hardware limits. Proprioception could also be accomplished through supervised learning. More importantly, object perception under machine learning helps robots with a better understanding of manipulation, which boosts the development of garbage sorting (Figure 11d),[46] automatic sorting, manufacturing, and others. This also could be applied in other various situations, such as smart care (Figure 11e),[118] digital twin applications,[44,118] and virtual reality. Besides object perception, environment monitoring has also been achieved.[12] These sensing abilities also open the door of prosthesis applications for tactile sensors, which holds a vast development prospect. Also, monitoring through manipulation using machine learning provides robots with predictions of slips or other cases, which offers robots opportunities to take flexible strategies to deal with different situations. Moreover, sensing during human–robot interactions is another topic with great potential. With its prominent data processing ability, machine learning would undeniably prompt this area's development.

4.2. Applications of Unsupervised Learning

Regarding unsupervised learning, clustering predicts groups of data points without assigned labels, whereas dimension reduction identifies a low-dimensional representation for a high number of dimensions. Tactile sensing usually has a specific goal, and thereby unsupervised learning for clustering or dimension reduction always works as an intermediary step of data processing or as an approach to analyzing data. When working as an intermediary step, unsupervised learning on clustering plays the role of data cleaning, which filters out data outliers based on clustering results.[42] Besides, clustering helps reduce continuous data redundancy by compressing data with clusters.[55] Moreover, clustering is commonly used to visualize data for analysis.[17,48,78] Dimension reduction, sometimes referred to as squeezing or feature extraction, functions by reducing data

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

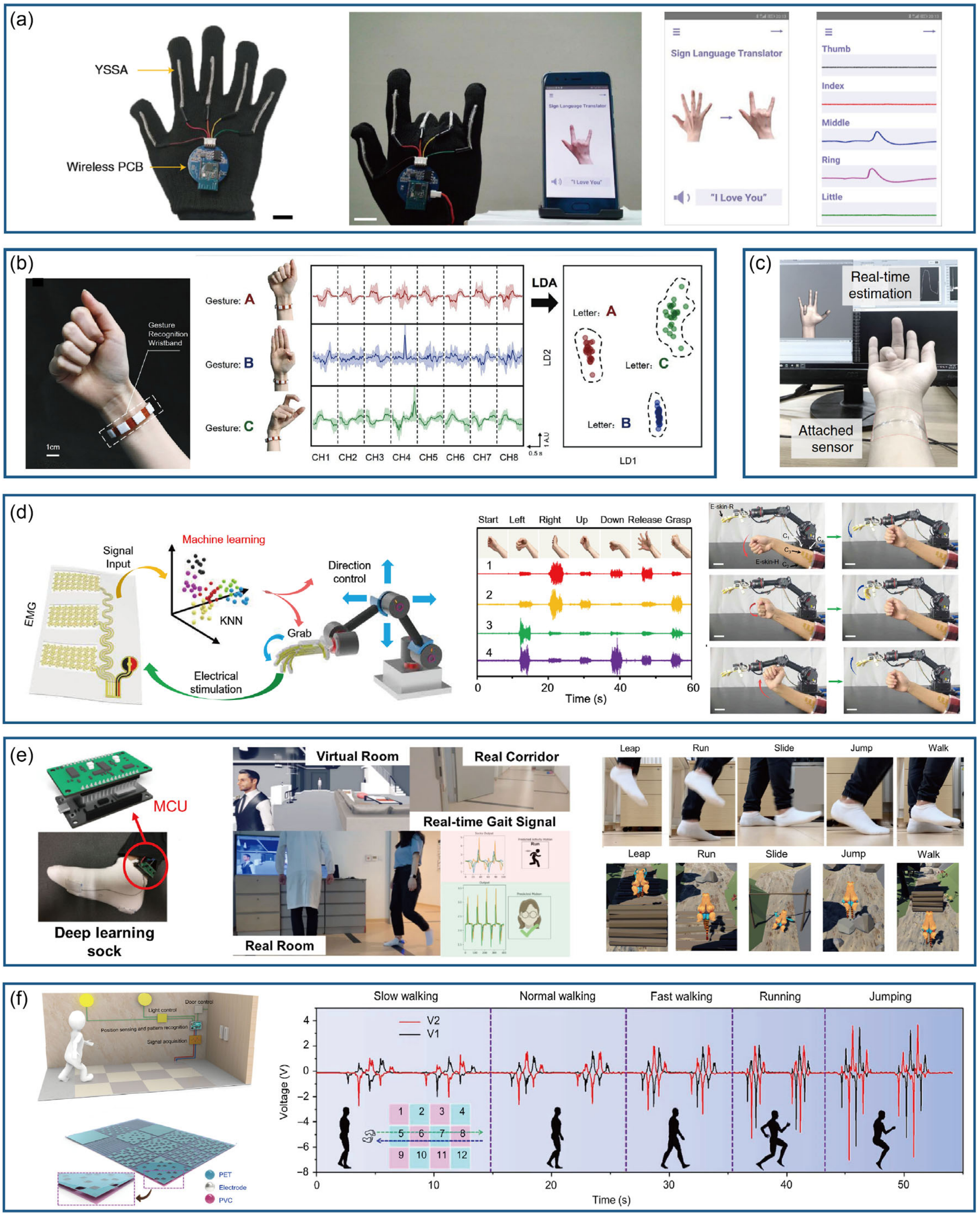

Figure 9. Tactile applications in hand motion monitoring and gait analysis with supervised learning. a) Sign-to-speech translation using tactile gloves. Reproduced with permission.[77] Copyright 2020, Springer Nature. b) Hand gesture recognition using sensors on arms. Reproduced with permission.[15] Copyright 2022, Wiley-VCH. c) Regression of finger movements using attached sensor on the arm. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[110] Copyright 2020, The Authors, published by Springer Nature. d) Controlling robotic hand with gestures recognized using sensors on the arm. Reproduced with with permission.[12] Copyright 2022, AAAS. e) Gait analysis with sensors on socks for human activity classification and participant recognition. Also, application of controlling characters in virtual reality was demonstrated. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[112] Copyright 2020, The Authors, published by Springer Nature. f) Gait analysis using mat tactile sensors for user recognition and walking status classification. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[170] Copyright 2020, The Authors, published by Springer Nature.

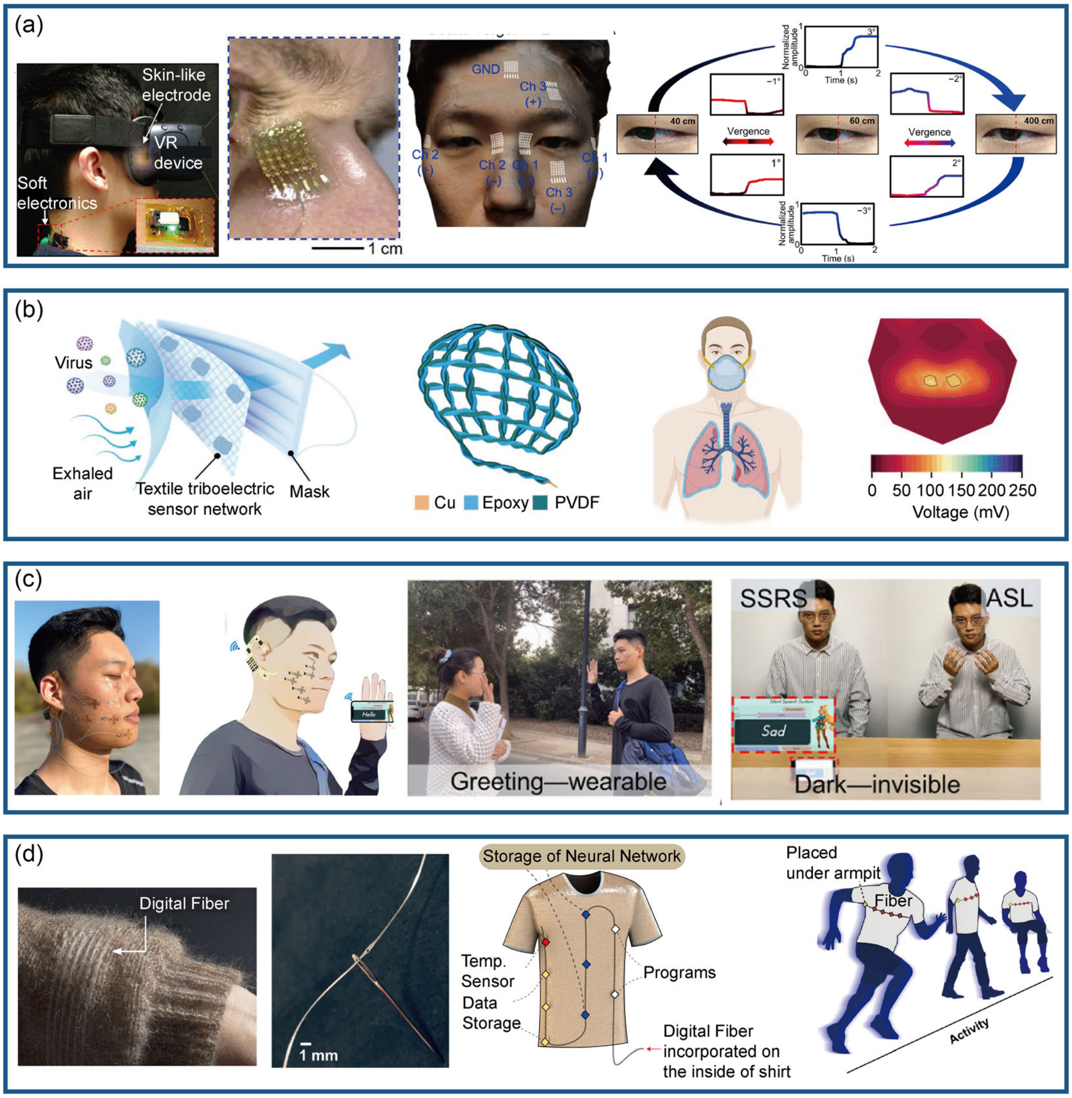

Figure 10. Besides hand motion monitoring and gait analysis, other typical human activity monitoring applications with tactile perception using machine learning are shown here. a) Eye vergence recognition with skin-like electrodes. Reproduced with permission under the terms of CC BY-NC Creative Commons Attribution NonCommercial License 4.0 (https://creativecommons.org/licenses/by-nc/4.0/).[90] Copyright 2020, The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. Published by AAAS. b) Respiration pattern recognition with on-mask sensors. Reproduced with permission.[171] Copyright 2022, Wiley-VCH. c) Speech recognition with face tattoo-like sensors. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[91] Copyright 2021, The Authors, published by Springer Nature. d) Action recognition with digital fibers. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[117] Copyright 2021, The Authors, published by Springer Nature.

size for further processing. Some machine learning algorithms are hard to deal with large-scale data, and dimension reduction is recruited to process data before data are input to those algorithms.[55,60–64,77,82,83,85]

5. Conclusion and Perspective

In this work, we have presented how tactile data processes from raw sensor output to aim results in analog and spike modality and introduced how unsupervised and supervised learning is applied in data process procedure. Emerging tactile sensors and machine learning technology bring a revolution in robotic perception, human–machine interfaces, electric skin, and health care monitoring. However, there are some challenging tasks worth mentioning that require next-step development.

5.1. Sensors Design

In the past decades, most tactile intelligent systems were sensor-centralized or structure-centralized designed, applied machine learning assisted. This design strategy virtually set a limit on optimal machine learning performance and introduced some

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

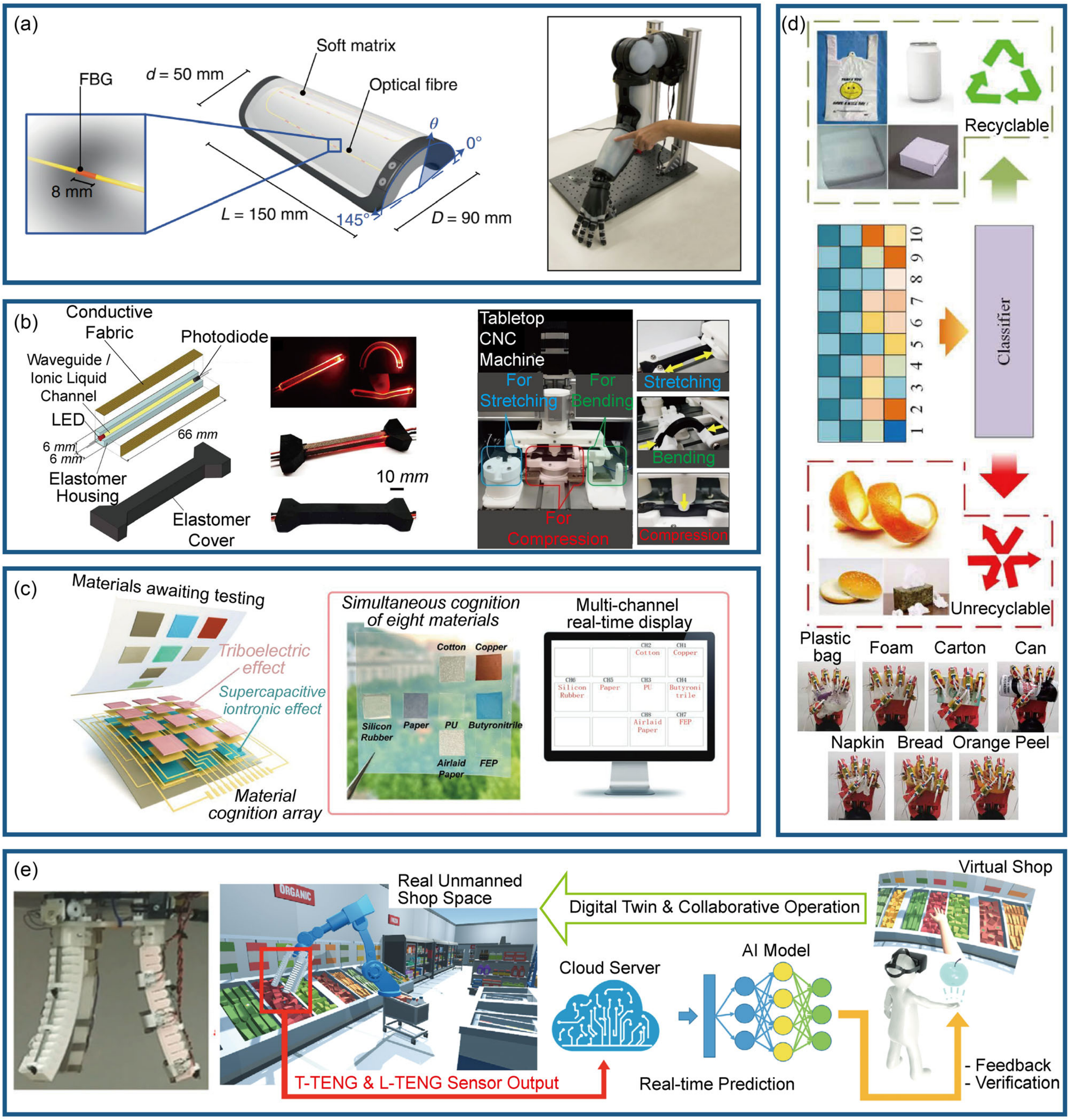

Figure 11. Supervised learning for robot tactile perception. a) Regression of contact force intensity and location. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[54] Copyright 2022, The Authors, published by Springer Nature. b) Proprioception of sensor deformation mode classification. Reproduced with permission.[173] Copyright 2020, AAAS. c) Material classification using tactile sensors by pressing. Reproduced with permission.[14] Copyright 2022, Wiley-VCH. d) Garbage sorting with tactile sensors on robotic hands. Reproduced with permission.[46] Copyright 2020, AAAS. e) Object recognition with gripper tactile sensors for digital twin application and collaborative operation. Reproduced under the terms of the CC-BY Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).[118] Copyright 2021, The Authors, published by Wiley-VCH.

inevitable noise. Thus machine learning-enabled sensor design could be introduced. By considering the algorithm selection and design of the signal process procedure, the desired performance is set firstly. In the iteration of sensor design, the target performance guides the redesign of sensors. It can be optimized in data dimension, input signal, e.g., processed in frames or time-related series. Thus some shortcomings can be overcome. The defined algorithm avoids complex model selection and

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

hard-characterized training data. Besides, feature optimization can be partly achieved at the hardware level. Furthermore, transfer learning would be easier achieved by sharing the same design strategy systems. Following this strategy, the next iteration of the intelligent sensor design will use the weights and hyperparameters determined in the previous iteration to initialize the sensing algorithm, thus simplifying the design process.

5.2. Data Quality and Quantity

As a data-driven method, machine learning requires high-quality data. Large data set also benefits knowledge discovery from data. However, unlike images or texts, tactile data are much more expensive. The quality of data heavily depends on the sensitivity and stability of sensors and transformation in the intelligent system. Large amounts of data provide a foundation to mine useful information behind them. While in the tactile data area, it is hard to reach. On the one hand, sensors in pixel points strategy require high stability. However, the data is hard to process and reconstruct interactions because of the geometrical complexity and redundant information. On the other hand, due to the resolution distribution of the human body, it is unnecessary to achieve high resolution on every point of the body except sensitive areas like fingertips, which require subtle recognition. Thus, some recent research in robotic body and arm[20,54] prefer to set sparse data points under multilayer electric skin to realize super-resolution recognition.

5.3. Algorithm Development

Currently, most machine learning methods used in tactile sensing are derived from algorithms for vision, natural language processing, and other areas, either by adopting algorithms to deal with tactile signals or by transforming tactile signals into ideal inputs that the algorithms can process. However, transforming tactile signals to corresponding algorithm inputs would either increase data redundancy or lose some tiny signals that may contain information. Adopting algorithms for tactile signals also sacrifices some advantages that are specifically designed for images and texts. Specially designed algorithms for tactile signals are still lacking. It is worth investigating this kind of algorithms to advance the understanding of tactile data. Spiking neural network is a potential choice, but it is still underdeveloped to catch up with the performance of other conventional machine learning methods.

5.4. Validation Benchmark

Though there are standard measurements to quantify the quality of tactile sensors, no benchmarks exist to compare the integrated performances of sensors and algorithms. Benchmarks like YCB benchmarks[176] in robotic manipulation are needed to be established in tactile fields, which could contain samples of textures and materials, and objects of different shapes, sizes, and weights. These benchmarks could cover tasks of tactile sensing and would enable a direct comparison of different sensors with their data processing methods.

Acknowledgements

Z.H. and L.L. contributed equally to this work. This research was supported by the Shenzhen Science and Technology Program (RCYX20200714114736115), the National Natural Science Foundation of China (61903317), the Guangdong Basic and Applied Basic Research Foundation (2022A1515011523), the Science and Technology Innovation Council of Shenzhen (KQTD20170810105439418, JCYJ 20200109114237902), and the Guangdong-Hongkong Joint Funding Program (2021A0505110015).

Conflict of Interest

The authors declare no conflict of interest.

Keywords

artificial intelligence, electronic skin, machine learning, signal processing, tactile perception

Received: October 21, 2022
Revised: February 7, 2023
Published online:

[1] C. Wang, L. Dong, D. Peng, C. Pan, *Adv. Intell. Syst.* **2019**, *1*, 1900090.
[2] J. C. Yang, J. Mun, S. Y. Kwon, S. Park, Z. Bao, S. Park, *Adv. Mater.* **2019**, *31*, 1904765.
[3] M. Wang, T. Wang, Y. Luo, K. He, L. Pan, Z. Li, Z. Cui, Z. Liu, J. Tu, X. Chen, *Adv. Funct. Mater.* **2021**, *31*, 2008807.
[4] M. Wang, Y. Luo, T. Wang, C. Wan, L. Pan, S. Pan, K. He, A. Neo, X. Chen, *Adv. Mater.* **2021**, *33*, 2003014.
[5] J. Park, M. Kim, Y. Lee, H. S. Lee, H. Ko, *Sci. Adv.* **2015**, *1*, 1500661.
[6] H. Oh, G.-C. Yi, M. Yip, S. A. Dayeh, *Sci. Adv.* **2020**, *6*, abd7795.
[7] T. Kim, J. Kim, I. You, J. Oh, S.-P. Kim, U. Jeong, *Sci. Robot.* **2022**, *7*, abl5761.
[8] Y. Li, J. Long, Y. Chen, Y. Huang, N. Zhao, *Adv. Mater.* **2022**, *34*, 2200517.
[9] S. Choi, G. S. Kim, J. Yang, H. Cho, C.-Y. Kang, G. Wang, *Adv. Mater.* **2022**, *34*, 2104598.
[10] M. I. Jordan, T. M. Mitchell, *Science* **2015**, *349*, 255.
[11] D. W. Otter, J. R. Medina, J. K. Kalita, *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 604.
[12] Y. Yu, J. Li, S. A. Solomon, J. Min, J. Tu, W. Guo, C. Xu, Y. Song, W. Gao, *Sci. Rob.* **2022**, *7*, abn0495.
[13] Y. Jung, J. Choi, W. Lee, J. S. Ko, I. Park, H. Cho, *Adv. Funct. Mater.* **2022**, *32*, 2201147.
[14] H. Niu, H. Li, S. Gao, Y. Li, X. Wei, Y. Chen, W. Yue, W. Zhou, G. Shen, *Adv. Mater.* **2022**, *34*, 2202622.
[15] P. Tan, X. Han, Y. Zou, X. Qu, J. Xue, T. Li, Y. Wang, R. Luo, X. Cui, Y. Xi, L. Wu, B. Xue, D. Luo, Y. Fan, X. Chen, Z. Li, Z. L. Wang, *Adv. Mater.* **2022**, *34*, 2200793.
[16] H. S. Wang, S. K. Hong, J. H. Han, Y. H. Jung, H. K. Jeong, T. H. Im, C. K. Jeong, B.-Y. Lee, G. Kim, C. D. Yoo, K. J. Lee, *Sci. Adv.* **2021**, *7*, abe5683.
[17] F. Wen, Z. Zhang, T. He, C. Lee, *Nat. Commun.* **2021**, *12*, 5378.
[18] X. Qu, Z. Liu, P. Tan, C. Wang, Y. Liu, H. Feng, D. Luo, Z. Li, Z. L. Wang, *Sci. Adv.* **2022**, *8*, abq2521.
[19] I. Van Meerbeek, C. De Sa, R. Shepherd, *Sci. Robot.* **2018**, *3*, aau2489.
[20] J. A. Barreiros, A. Xu, S. Pugach, N. Iyengar, G. Troxell, A. Cornwell, S. Hong, B. Selman, R. F. Shepherd, *Sci. Rob.* **2022**, *7*, abi6745.
[21] N. Golestani, M. Moghaddam, *Nat. Commun.* **2020**, *11*, 1551.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[22] Y. Yan, Z. Hu, Z. Yang, W. Yuan, C. Song, J. Pan, Y. Shen, *Sci. Rob.* **2021**, *6*, abc8801.

[23] S. K. Ravi, N. Paul, L. Suresh, A. T. Salim, T. Wu, Z. Wu, M. R. Jones, S. C. Tan, *Mater. Horiz.* **2020**, *7*, 866.

[24] S. Zhang, L. Suresh, J. Yang, X. Zhang, S. C. Tan, *Adv. Intell. Syst.* **2022**, *4*, 2100194.

[25] M. Liu, Y. Zhang, T. H. Tao, *Adv. Intell. Syst.* **2022**, *4*, 2100280.

[26] A. Yamaguchi, C. G. Atkeson, *Adv. Rob.* **2019**, *33*, 661.

[27] F. Liu, S. Deswal, A. Christou, Y. Sandamirskaya, M. Kaboli, R. Dahiya, *Sci. Rob.* **2022**, *7*, abl7344.

[28] S. Dai, Y. Zhao, Y. Wang, J. Zhang, L. Fang, S. Jin, Y. Shao, J. Huang, *Adv. Funct. Mater.* **2019**, *29*, 1903700.

[29] H.-L. Park, Y. Lee, N. Kim, D.-G. Seo, G.-T. Go, T.-W. Lee, *Adv. Mater.* **2020**, *32*, 1903558.

[30] J.-W. Han, M. Meyyappan, *IEEE Electron. Device Lett.* **2018**, *39*, 1457.

[31] J.-K. Han, M. Seo, W.-K. Kim, M.-S. Kim, S.-Y. Kim, M.-S. Kim, G.-J. Yun, G.-B. Lee, J.-M. Yu, Y.-K. Choi, *IEEE Electron. Device Lett.* **2019**, *41*, 208.

[32] C. Zhang, W. B. Ye, K. Zhou, H.-Y. Chen, J.-Q. Yang, G. Ding, X. Chen, Y. Zhou, L. Zhou, F. Li, S.-T. Han, *Adv. Funct. Mater.* **2019**, *29*, 1808783.

[33] J. Zhu, X. Zhang, R. Wang, M. Wang, P. Chen, L. Cheng, Z. Wu, Y. Wang, Q. Liu, M. Liu, *Adv. Mater.* **2022**, *34*, 2200481.

[34] C. Wan, G. Chen, Y. Fu, M. Wang, N. Matsuhisa, S. Pan, L. Pan, H. Yang, Q. Wan, L. Zhu, X. Chen, *Adv. Mater.* **2018**, *30*, 1801291.

[35] Q. Duan, T. Zhang, C. Liu, R. Yuan, G. Li, P. Jun Tiw, K. Yang, C. Ge, Y. Yang, R. Huang, *Adv. Intell. Syst.* **2022**, *4*, 2200039.

[36] H. Tan, Q. Tao, I. Pande, S. Majumdar, F. Liu, Y. Zhou, P. O. Persson, J. Rosen, S. van Dijken, *Nat. Commun.* **2020**, *11*, 1369.

[37] J. K. Eshraghian, M. Ward, E. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, W. D. Lu, (Preprint) arXiv:2109.12894, v4, submitted: Jan. **2022**.

[38] A. L. Hodgkin, A. F. Huxley, *J. Physiol.* **1952**, *117*, 500.

[39] M. Pfeiffer, T. Pfeil, *Front. Neurosci.* **2018**, *12*, 774.

[40] H. R. Nicholls, M. H. Lee, *Int. J. Rob. Res.* **1989**, *8*, 3.

[41] V. E. Abraira, D. D. Ginty, *Neuron* **2013**, *79*, 618.

[42] J. H. Lee, J. S. Heo, Y.-J. Kim, J. Eom, H. J. Jung, J.-W. Kim, I. Kim, H.-H. Park, H. S. Mo, Y.-H. Kim, S. K. Park, *Adv. Mater.* **2020**, *32*, 2000969.

[43] Y. Wan, Y. Wang, C. F. Guo, *Mater. Today Phys.* **2017**, *1*, 61.

[44] T. Jin, Z. Sun, L. Li, Q. Zhang, M. Zhu, Z. Zhang, G. Yuan, T. Chen, Y. Tian, X. Hou, C. Lee, *Nat. Commun.* **2020**, *11*, 5381.

[45] M. Zhu, Z. Sun, Z. Zhang, Q. Shi, T. He, H. Liu, T. Chen, C. Lee, *Sci. Adv.* **2020**, *6*, aaz8693.

[46] G. Li, S. Liu, L. Wang, R. Zhu, *Sci. Rob.* **2020**, *5*, abc8134.

[47] K. Pang, X. Song, Z. Xu, X. Liu, Y. Liu, L. Zhong, Y. Peng, J. Wang, J. Zhou, F. Meng, J. Wang, C. Gao, *Sci. Adv.* **2020**, *6*, abd4045.

[48] Y. Luo, Y. Li, P. Sharma, W. Shou, K. Wu, M. Foshey, B. Li, T. Palacios, A. Torralba, W. Matusik, *Nat. Electron.* **2021**, *4*, 193.

[49] J. Li, S. Dong, E. Adelson, in *Proc. 2018 IEEE Int. Conf. on Robotics and Automation*, IEEE, Brisbane, QLD, Australia 21–25 May **2018**, p. 7772.

[50] M. Levins, H. Lang, *IEEE Sens. J.* **2020**, *20*, 13284.

[51] G. Li, S. Liu, Q. Mao, R. Zhu, *Adv. Sci.* **2022**, *9*, 2104969.

[52] T. G. Thuruthel, B. Shih, C. Laschi, M. T. Tolley, *Sci. Rob.* **2019**, *4*, aav1488.

[53] J. G. Greener, S. M. Kandathil, L. Moffat, D. T. Jones, *Nat. Rev. Mol. Cell Biol.* **2022**, *23*, 40.

[54] L. Massari, G. Fransvea, J. D'Abbraccio, M. Filosa, G. Terruso, A. Aliperta, G. D'Alesio, M. Zaltieri, E. Schena, E. Palermo, E. Sinibaldi, C. M. Oddo, *Nat. Mach. Intell.* **2022**, *4*, 425.

[55] S. Sundaram, P. Kellnhofer, Y. Li, J.-Y. Zhu, A. Torralba, W. Matusik, *Nature* **2019**, *569*, 698.

[56] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, A. Rodriguez, in *Proc. 2021 IEEE Int. Conf. on Robotics and Automation*, IEEE, Xi'an, China 30 May–05 June **2021**, p. 6437.

[57] R. B. Hellman, C. Tekin, M. van der Schaar, V. J. Santos, *IEEE Trans. Haptic.* **2017**, *11*, 61.

[58] A. Church, J. Lloyd, R. Hadsell, N. F. Lepora, *IEEE Rob. Autom. Lett.* **2020**, *5*, 6145.

[59] H. Liang, L. Cong, N. Hendrich, S. Li, F. Sun, J. Zhang, *IEEE Robot. Autom. Lett.* **2021**, *7*, 1174.

[60] Y. Chebotar, K. Hausman, Z. Su, G. S. Sukhatme, S. Schaal, in *Proc. 2016 IEEE Int. Conf. on Intelligent Robots and Systems*, IEEE, Daejeon, Korea (South) 09–14 October **2016**, p. 1960.

[61] H. Van Hoof, N. Chen, M. Karl, P. van der Smagt, J. Peters, in *Proc. 2016 IEEE Int. Conf. on Intelligent Robots and Systems*, IEEE, Daejeon, Korea (South) 09–14 October **2016**, p. 3928.

[62] W. Liang, Q. Ren, X. Chen, J. Gao, Y. Wu, in *Proc. 2021 IEEE Int. Conf. on Robotics and Automics*, IEEE, Xi'an, China 30 May–05 June **2021**, p. 6308.

[63] Z. Wu, W. Lian, V. Unhelkar, M. Tomizuka, S. Schaal, in *Proc. 2021 IEEE Int. Conf. on Robotics Automation*, IEEE, Xi'an, China 30 May–05 June **2021**, p. 6214.

[64] Y. Zheng, F. F. Veiga, J. Peters, V. J. Santos, *IEEE Trans. Rob.* **2022**, *38*, 2734.

[65] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, F. L. Lewis, *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 2042.

[66] B. Singh, R. Kumar, V. P. Singh, *Artif. Intell. Rev.* **2022**, *55*, 945.

[67] S. Duan, Q. Shi, J. Wu, *Adv. Intell. Syst.* **2022**, *4*, 2200213.

[68] S. Lloyd, *IEEE Trans. Inf. Theory* **1982**, *28*, 129.

[69] S. Nasiriany, G. Thomas, W. Wang, A. Yang, J. Listgarten, A. Sahai, *A Comprehensive Guide to Machine Learning*, Department of Electrical Engineering and Computer Sciences University of California, Berkeley, Berkeley, CA **2019**.

[70] C. Amendola, J.-C. Faugere, B. Sturmfels, *J. Algebr. Stat.* **2016**, *7*, 14.

[71] A. P. Dempster, N. M. Laird, D. B. Rubin, *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1977**, *39*, 1.

[72] C. J. Wu, *Ann. Stat.* **1983**, *11*, 95.

[73] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ **1957**.

[74] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton Legacy Library, Princeton University Press, Princeton, NJ **1961**.

[75] K. Pearson, *London Edinburgh Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559.

[76] H. Hotelling, *J. Educ. Psychol.* **1933**, *24*, 417.

[77] Z. Zhou, K. Chen, X. Li, S. Zhang, Y. Wu, Y. Zhou, K. Meng, C. Sun, Q. He, W. Fan, E. Fan, Z. Lin, X. Tan, W. Deng, J. Yang, J. Chen, *Nat. Electron.* **2020**, *3*, 571.

[78] 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, J. M. Rabaey, *Nat. Electron.* **2021**, *4*, 54.

[79] S. G. Mallat, Z. Zhang, *IEEE Trans. Signal Process.* **1993**, *41*, 3397.

[80] S. S. Chen, D. L. Donoho, M. A. Saunders, *SIAM Rev. Soc. Ind Appl. Math.* **2001**, *43*, 129.

[81] A. M. Bruckstein, D. L. Donoho, M. Elad, *SIAM Rev. Soc. Ind. Appl. Math.* **2009**, *51*, 34.

[82] J.-P. Roberge, S. Rispal, T. Wong, V. Duchaine, in *Proc. 2016 IEEE Int. Conf. on Robotics and Automation*, IEEE, Stockholm, Sweden 16–21 May **2016**, p. 2675.

[83] H. Liu, J. Qin, F. Sun, D. Guo, *IEEE Trans. Cybern.* **2016**, *47*, 4509.

[84] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, T. Darrell, K. J. Kuchenbecker, *Rob. Auton. Syst.* **2015**, *63*, 279.

[85] M. Polic, I. Krajacic, N. Lepora, M. Orsag, *IEEE Rob. Autom. Lett.* **2019**, *4*, 3671.

[86] O. Chapelle, P. Haffner, V. N. Vapnik, *IEEE Trans. Neural Netw. Learn. Syst.* **1999**, *10*, 1055.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[87] B. E. Boser, I. M. Guyon, V. N. Vapnik, in *Proc. 5th Annual Workshop Computational Learning Theory*, Association for Computing Machinery Pittsburgh, Pennsylvania, USA 27–29 July **1992**, p. 144.
[88] C.-W. Hsu, C.-J. Lin, *IEEE Trans. Neural Netw. Learn. Syst.* **2002**, *13*, 415.
[89] Y. Shao, *Tactile Sensing, Information, and Feedback Via Wave Propagation*, Springer, Cham **2022**.
[90] S. Mishra, Y.-S. Kim, J. Intarasirisawat, Y.-T. Kwon, Y. Lee, M. Mahmood, H.-R. Lim, R. Herbert, K. J. Yu, C. S. Ang, W.-H. Yeo, *Sci. Adv.* **2020**, *6*, aay1729.
[91] Y. Wang, T. Tang, Y. Xu, Y. Bai, L. Yin, G. Li, H. Zhang, H. Liu, Y. Huang, *npj Flexible Electron.* **2021**, *5*, 20.
[92] A. J. Smola, B. Schölkopf, *Stat. Comput.* **2004**, *14*, 199.
[93] T. Taunyazov, L. S. Song, E. Lim, H. H. See, D. Lee, B. C. Tee, H. Soh, in *Proc. 2021 IEEE Int. Conf. on Intelligent Robots and Systems*, IEEE, Prague, Czech Republic 27 September–01 October **2021**, p. 1755.
[94] Y. Tong, Z. Feng, J. Kim, J. L. Robertson, X. Jia, B. N. Johnson, *Nano Energy* **2020**, *75*, 104973.
[95] Y.-T. Kwon, Y.-S. Kim, S. Kwon, M. Mahmood, H.-R. Lim, S.-W. Park, S.-O. Kang, J. J. Choi, R. Herbert, Y. C. Jang, Y.-H. Choa, W.-H. Yeo, *Nat. Commun.* **2020**, *11*, 3450.
[96] D. Opitz, R. Maclin, *J. Artif. Intell. Res.* **1999**, *11*, 169.
[97] T. K. Ho, in *Proc. 3rd Int. Conf. on Document Analysis and Recognition*, IEEE, Montreal, QC, Canada 14–16 August **1995**, p. 278.
[98] L. Breiman, *Mach. Learn.* **2001**, *45*, 5.
[99] K. S. Chun, Y. J. Kang, J. Y. Lee, M. Nguyen, B. Lee, R. Lee, H. H. Jo, E. Allen, H. Chen, J. Kim, L. Yu, X. Ni, K. Lee, H. Jeong, J. Lee, Y. Park, H. U. Chung, A. W. Li, P. A. Lio, A. F. Yang, A. B. Fishbein, A. S. Paller, J. A. Rogers, S. Xu, *Sci. Adv.* **2021**, *7*, abf9405.
[100] A. J. Spiers, M. V. Liarokapis, B. Calli, A. M. Dollar, *IEEE Trans. Haptic.* **2016**, *9*, 207.
[101] P. Ribeiro, S. Cardoso, A. Bernardino, L. Jamone, in *Proc. 2020 IEEE Int. Conf. on Intelligent. Robots and Systems*, IEEE, Las Vegas, NV, USA 25–29 October **2020**, p. 8875.
[102] S. Mishra, J. J. Norton, Y. Lee, D. S. Lee, N. Agee, Y. Chen, Y. Chun, W.-H. Yeo, *Biosens. Bioelectron.* **2017**, *91*, 796.
[103] T. Corradi, P. Hall, P. Iravani, in *Proc. 2015 IEEE Int. Conf. on Robotics and Automation*, IEEE, Seattle, WA, USA 26–30 May **2015**, p. 3909.
[104] N. F. Lepora, M. Evans, C. W. Fox, M. E. Diamond, K. Gurney, T. J. Prescott, in *Proc. 2010 Int. Joint Conf. on Neural Networks*, IEEE, Barcelona, Spain 18–23 July **2010**, p. 1, https://doi.org/10.1109/IJCNN.2010.5596360.
[105] P. Slade, M. J. Kochenderfer, S. L. Delp, S. H. Collins, *Nat. Commun.* **2021**, *12*, 4312.
[106] X. Wei, H. Li, W. Yue, S. Gao, Z. Chen, Y. Li, G. Shen, *Matter* **2022**, *5*, 1481.
[107] O. A. Araromi, M. A. Graule, K. L. Dorsey, S. Castellanos, J. R. Foster, W.-H. Hsu, A. E. Passy, J. J. Vlassak, J. C. Weaver, C. J. Walsh, R. J. Wood, *Nature* **2020**, *587*, 219.
[108] S. Duan, Y. Lin, C. Zhang, Y. Li, D. Zhu, J. Wu, W. Lei, *Nano Energy* **2022**, *91*, 106650.
[109] K. Fukushima, S. Miyake, *Competition and Cooperation in Neural Nets*, Springer, Berlin Heidelberg **1982**, p. 267.
[110] K. K. Kim, I. Ha, M. Kim, J. Choi, P. Won, S. Jo, S. H. Ko, *Nat. Commun.* **2020**, *11*, 2149.
[111] K. Park, H. Yuk, M. Yang, J. Cho, H. Lee, J. Kim, *Sci. Rob.* **2022**, *7*, eabm7187.
[112] Z. Zhang, T. He, M. Zhu, Z. Sun, Q. Shi, J. Zhu, B. Dong, M. R. Yuce, C. Lee, *npj Flexible Electron.* **2020**, *4*, 29.
[113] F. Wen, Z. Sun, T. He, Q. Shi, M. Zhu, Z. Zhang, L. Li, T. Zhang, C. Lee, *Adv. Sci.* **2020**, *7*, 2000261.
[114] G.-H. Lee, J.-K. Park, J. Byun, J. C. Yang, S. Y. Kwon, C. Kim, C. Jang, J. Y. Sim, J.-G. Yook, S. Park, *Adv. Mater.* **2020**, *32*, 1906269.
[115] M. Liu, Y. Zhang, J. Wang, N. Qin, H. Yang, K. Sun, J. Hao, L. Shu, J. Liu, Q. Chen, P. Zhang, T. H. Tao, *Nat. Commun.* **2022**, *13*, 70.
[116] X. Wei, B. Wang, Z. Wu, Z. L. Wang, *Adv. Mater.* **2022** 2203073.
[117] G. Loke, T. Khudiyev, B. Wang, S. Fu, S. Payra, Y. Shaoul, J. Fung, I. Chatziveroglou, P.-W. Chou, I. Chinn, W. Yan, A. Gitelson-Kahn, J. Joannopoulos, Y. Fink, *Nat. Commun.* **2021**, *12*, 3317.
[118] Z. Sun, M. Zhu, Z. Zhang, Z. Chen, Q. Shi, X. Shan, R. C. H. Yeow, C. Lee, *Adv. Sci.* **2021**, *8*, 2100230.
[119] K. He, X. Zhang, S. Ren, J. Sun, in *Proc. IEEE Conf. on Computer Vision Pattern Recognition*, IEEE, Las Vegas, NV, USA 27–30 June **2016**, p. 770.
[120] B. Fang, X. Long, F. Sun, H. Liu, S. Zhang, C. Fang, *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1.
[121] B. Xiao, W. Xu, J. Guo, H.-K. Lam, G. Jia, W. Hong, H. Ren, *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1791.
[122] W. Maass, *Neural Netw.* **1997**, *10*, 1659.
[123] J.-K. Han, I.-W. Tcho, S.-B. Jeon, J.-M. Yu, W.-G. Kim, Y.-K. Choi, *Adv. Sci.* **2022**, *9*, 2105076.
[124] L. E. Osborn, A. Dragomir, J. L. Betthauser, C. L. Hunt, H. H. Nguyen, R. R. Kaliki, N. V. Thakor, *Sci. Rob.* **2018**, *3*, aat3818.
[125] Y. Kim, A. Chortos, W. Xu, Y. Liu, J. Y. Oh, D. Son, J. Kang, A. M. Foudeh, C. Zhu, Y. Lee, S. Niu, J. Liu, R. Pfattner, Z. Bao, T.-W. Lee, *Science* **2018**, *360*, 998.
[126] Y. Lee, Y. Liu, D.-G. Seo, J. Y. Oh, Y. Kim, J. Li, J. Kang, J. Kim, J. Mun, A. M. Foudeh, Z. Bao, T.-W. Lee, *Nat. Biomed. Eng.* **2022**, https://doi.org/10.1038/s41551-022-00918-x.
[127] S. Sankar, D. Balamurugan, A. Brown, K. Ding, X. Xu, J. H. Low, C. H. Yeow, N. Thakor, *Soft. Rob.* **2021**, *8*, 577.
[128] L. Shan, Y. Liu, X. Zhang, E. Li, R. Yu, Q. Lian, X. Chen, H. Chen, T. Guo, *Nano Energy* **2021**, *88*, 106283.
[129] B. Ward-Cherrier, N. Pestell, N. F. Lepora, in *Proc. 2020 IEEE Int. Conf. on Robotics and Automation*, IEEE, Paris, France 31 May–31 August **2020**, p. 2654.
[130] L. Chen, C. Wen, S.-L. Zhang, Z. L. Wang, Z.-B. Zhang, *Nano Energy* **2021**, *82*, 105680.
[131] K. Lee, S. Jang, K. L. Kim, M. Koo, C. Park, S. Lee, J. Lee, G. Wang, C. Park, *Adv. Sci.* **2020**, *7*, 2001662.
[132] E. M. Izhikevich, *IEEE Trans. Neural Netw. Learn. Syst.* **2003**, *14*, 1569.
[133] L. D. Bucci, T.-S. Chou, J. L. Krichmar, in *Proc. 2014 IEEE Int. Conf. on Robotics and Automation*, IEEE, Hong Kong, China 31 May–07 June **2014**, p. 1909.
[134] S. Roy, P. P. San, S. Hussain, L. W. Wei, A. Basu, *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 1572.
[135] U. B. Rongala, A. Mazzoni, C. M. Oddo, *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *28*, 819.
[136] Z. Yi, Y. Zhang, *Neurocomputing* **2017**, *244*, 102.
[137] M. Rasouli, Y. Chen, A. Basu, S. L. Kukreja, N. V. Thakor, *IEEE Trans. Biomed. Circuits Syst.* **2018**, *12*, 313.
[138] D. Kumar, R. Ghosh, A. Nakagawa-Silva, A. B. Soares, N. V. Thakor, *Neurocomputing* **2020**, *407*, 246.
[139] Z. Yi, T. Xu, S. Guo, W. Shang, X. Wu, *IEEE Trans. Autom. Sci. Eng.* **2020**, *18*, 1835.
[140] L. Qin, Y. Zhang, *Neural. Comput. Appl.* **2021**, *33*, 14793.
[141] L. Lapique, *J. Physiol. Pathol.* **1907**, *9*, 620.
[142] N. Brunel, M. C. Van Rossum, *Biol. Cybern.* **2007**, *97*, 337.
[143] W. Gerstner, W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, Cambridge, England **2002**.
[144] T. Taunyazov, Y. Chua, R. Gao, H. Soh, Y. Wu, in *Proc. 2020 IEEE Int. Conf. on Intelligent Robots and System*, IEEE, Las Vegas, NV, USA 25–29 October **2020**, p. 9890.
[145] W. Navaraj, R. Dahiya, *Adv. Intell. Syst.* **2019**, *1*, 1900051.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[146] D. Sengupta, M. Mastella, E. Chicca, A. G. P. Kottapalli, *ACS Appl. Electron. Mater.* **2022**, *4*, 308.
[147] F. Gu, W. Sng, T. Taunyazov, H. Soh, in *Proc. 2020 IEEE Int. Conf. on Intelligent Robots and System*, IEEE, Las Vegas, NV, USA 25–29 October **2020**, p. 9876.
[148] T. Taunyazoz, W. Sng, H. H. See, B. Lim, J. Kuan, A. F. Ansari, B. Tee, H. Soh, in *Proc. Robotics: Science and Systems*, Robotics: Science and Systems, Virtual Event, 12–16 July **2020**.
[149] K. E. Friedl, A. R. Voelker, A. Peer, C. Eliasmith, *IEEE Rob. Autom. Lett.* **2016**, *1*, 516.
[150] F. Ponulak, A. Kasinski, *Acta Neurobiol. Exp.* **2011**, *71*, 409.
[151] D. Auge, J. Hille, E. Mueller, A. Knoll, *Neural Process Lett.* **2021**, *53*, 4693.
[152] B. Petro, N. Kasabov, R. M. Kiss, *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 358.
[153] G.-q. Bi, M.-m. Poo, *J. Neurosci.* **1998**, *18*, 10464.
[154] S. Song, K. D. Miller, L. F. Abbott, *Nat. Neurosci.* **2000**, *3*, 919.
[155] R. Gütig, H. Sompolinsky, *Nat. Neurosci.* **2006**, *9*, 420.
[156] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *Proc. IEEE* **1998**, *86*, 2278.
[157] S. B. Shrestha, G. Orchard, *Adv. Neural Inf. Process Syst.* **2018**, *31*, 1419.
[158] J. A. Pérez-Carrasco, B. Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, S. Chen, B. Linares-Barranco, *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2706.
[159] J. Y. Angela, M. A. Giese, T. A. Poggio, *Neural Comput.* **2002**, *14*, 2857.
[160] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, *IEEE Trans. Syst. Man Cybern. B Cybern.* **2011**, *42*, 513.
[161] B. Ward-Cherrier, J. Conradt, M. G. Catalano, M. Bianchi, N. F. Lepora, in *Proc. 2020 IEEE Int. Conf. on Intelligent Robots and System*, IEEE, Las Vegas, NV, USA 25–29 October **2020**, p. 9883.
[162] W. W. Lee, Y. J. Tan, H. Yao, S. Li, H. H. See, M. Hon, K. A. Ng, B. Xiong, J. S. Ho, B. C. Tee, *Sci. Rob.* **2019**, *4*, eaax2198.
[163] J. D. Victor, K. P. Purpura, *J. Neurophysiol.* **1996**, *76*, 1310.
[164] D. Aronov, *J. Neurosci. Methods* **2003**, *124*, 175.
[165] T. Kreuz, J. S. Haas, A. Morelli, H. D. Abarbanel, A. Politi, *J. Neurosci. Methods* **2007**, *165*, 151.
[166] T. Kreuz, D. Chicharro, R. G. Andrzejak, J. S. Haas, H. D. Abarbanel, *J. Neurosci. Methods* **2009**, *183*, 287.
[167] E. Satuvuori, T. Kreuz, *J. Neurosci. Methods* **2018**, *299*, 22.
[168] U. Proske, S. C. Gandevia, *Physiol. Rev.* **2012**, *92*, 1651.
[169] Q. Zhang, T. Jin, J. Cai, L. Xu, T. He, T. Wang, Y. Tian, L. Li, Y. Peng, C. Lee, *Adv. Sci.* **2022**, *9*, 2103694.
[170] Q. Shi, Z. Zhang, T. He, Z. Sun, B. Wang, Y. Feng, X. Shan, B. Salam, C. Lee, *Nat. Commun.* **2020**, *11*, 4609.
[171] Y. Fang, J. Xu, X. Xiao, Y. Zou, X. Zhao, Y. Zhou, J. Chen, *Adv. Mater.* **2022**, *34*, 2200252.
[172] J. Lloyd, N. F. Lepora, *IEEE Trans. Rob.* **2021**, *38*, 1201.
[173] T. Kim, S. Lee, T. Hong, G. Shin, T. Kim, Y.-L. Park, *Sci. Robot.* **2020**, *5*, abc6878.
[174] T. Taunyazov, H. F. Koh, Y. Wu, C. Cai, H. Soh, in *Proc. 2019 IEEE Int. Conf. on Robotics and Automation*, IEEE, Montreal, QC, Canada 20–24 May **2019**, p. 4269.
[175] K. Kim, M. Sim, S.-H. Lim, D. Kim, D. Lee, K. Shin, C. Moon, J.-W. Choi, J. E. Jang, *Adv. Sci.* **2021**, *8*, 2002362.
[176] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, A. M. Dollar, *Int. J. Rob. Res.* **2017**, *36*, 261.
[177] H. Tan, Y. Zhou, Q. Tao, J. Rosen, S. van Dijken, *Nat. Commun.* **2021**, *12*, 1120.
[178] Y. R. Lee, T. Q. Trung, B.-U. Hwang, N.-E. Lee, *Nat. Commun.* **2020**, *11*, 2753.
[179] Y. Yan, Z. Hu, Y. Shen, J. Pan, *Adv. Intell. Syst.* **2022**, *4*, 2100076.

Zhixian Hu received a B.Eng. in electronic information engineering and an M.Phil. in computer information engineering from the Chinese University of Hong Kong, Shenzhen. Her research interest includes human–robot interaction and robot perception and cognition.

Lan Lin is currently studying for a master's degree at Shenzhen University. She received her bachelor's degree at East China Normal University in computer science and technology. Her main research interests are in the fields of algorithms in tactile data process.

Waner Lin is a Ph.D. student in School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. She obtained her master degree from Shanghai Jiao Tong University in 2022 and bachelor degree from Shenzhen University. Her research thrust includes flexible electronics for robotic tactile perception.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Yingtian Xu received his bachelor's degree in mechanical engineering from Tianjin University and his M.S. degree in robotics from Johns Hopkins University. He is currently working toward the Ph.D. degree in the tactile sensing and sensor-based robot control with Chinese University of Hong Kong, Shenzhen.

Xuan Xia received the Ph.D. degree in instrument science and technology from Shanghai Jiao Tong University, in 2017. He is currently a Researcher with the Shenzhen Institute of Artificial Intelligence and Robotics for Society. His research areas include pattern recognition, anomaly detection, defect detection, generative model, and image-to-image translation.

Zhengchun Peng received his B.S. in mechanical engineering from Beijing Institute of Technology, China, and Ph.D. degree in MEMS from the Georgia Institute of Technology, USA. He then joined Intel Corp. as a Senior R&D Engineer, developing Wafer Sort and E-Test technologies for 14 nm and 10 nm processors. He is currently a distinguished professor at Shenzhen University. His main research interests are in the fields of micro/nanosensors, flexible/stretchable electronics, and their application in robotics, human–machine interface, and healthcare.

Zhenglong Sun received the B.Eng. degree in biomedical engineering, M.Eng. and Ph.D. in mechatronic system and design from the Nanyang Technological University (NTU). Currently, he is an Assistant Professor at the Chinese University of Hong Kong, Shenzhen. His research interests include medical/surgical robot, haptic sensing and feedback, and human–machine interaction. He has published over 70 journal and conference papers.

Ziya Wang is currently an assistant professor of Physics and Optoelectronic Engineering at Shenzhen University. She received her B.S. and Ph.D. degree in physics from University of Science & Technology Beijing. Her research interests are in smart functional materials, soft sensors and electronics, and robotic tactile perception.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202200371 by Shenzhen University, Wiley Online Library on [15/03/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License