Machine Learning-Enabled Smart Sensor Systems
Nam Ha, Kai Xu, Guanghui Ren, Arnan Mitchell, and Jian Zhen Ou*

Recent advancements and major breakthroughs in machine learning (ML) technologies in the past decade have made it possible to collect, analyze, and interpret an unprecedented amount of sensory information. A new era for “smart” sensor systems is emerging that changes the way that conventional sensor systems are used to understand the world. Smart sensor systems have taken advantage of classic and emerging ML algorithms and modern computer hardware to create sophisticated “smart” models that are tailored specifically for sensing applications and fusing diverse sensing modalities to gain a more holistic appreciation of the system being monitored. Herein, a review of the recent sensing applications, which harness ML enabled smart sensor systems, is presented. First well-known ML algorithms implemented in smart sensor systems for practical sensing applications are discussed. Subsequent sections summarize the practical sensing applications under two major categories: physical and chemical sensing and visual imaging sensing describing how the sensor technologies are coupled with ML “smart” models and how these systems achieve practical benefits. Finally, an outlook on the current trajectory and challenges that will be faced by future smart sensing systems and the opportunities that may be unlocked is provided.

1. Introduction

Sensors are essential components of any modern device in the 21st century. Practically, all modern systems with even modest complexity will rely on at least one sensor, and many will have dozens of embedded sensors. These types of sensors are found in diverse practical applications such as medical diagnostics,[1] sustaining human health and wellbeing,[2] environmental monitoring,[3] civil engineering,[4] agriculture farming,[5] among others. Sensors can be as sophisticated as Raman spectrometers[6] measuring distinct vibrations within the individual molecules that make up material to obtain a complex chemical fingerprint or can be as simple as a classic thermometer where liquid levels vary with respect to temperature.[7]

The fundamental function of any sensor is to detect and provide accurate information of its sensing target; and depending on the type of sensor, the outcome of the measurements may come in different forms such as electrical voltage signals from a gas sensor,[8] or digital images captured from a complementary metal-oxide-semiconductor (CMOS) image sensor array[9] in the camera of a microscope.

Conventional sensing systems typically monitor a single sensor output, for instance, a device that measures the resistance output change of a thermometer with respect to temperature. However, there are a growing number of systems that have inbuilt arrays of sensors with multimeasurement outputs such as an electronic nose[10] device that has inbuilt arrays of MOX gas sensors[8] or the variety of sensors integrated into an Internet of Things (IoT) device.[11] The increased number of sensors in devices will inherently generate higher data throughput, which poses a serious challenge in managing and processing the tremendous amount of sensory information. Furthermore, traditional processing techniques in conventional sensing devices are no longer suitable for systematically labeling, processing, and analyzing the exuberant amount of information.[12]

The recent advancement of machine learning (ML) algorithms and theories[13] has offered new opportunities and insight to address these challenges adequately. Real-world applications that are already benefiting from such ML algorithms include healthcare analytics,[14] business,[15] economics,[16] analytics and computer science,[17] among others. The strength of ML algorithms prowess lies in its ability to learn and automate the extraction of patterns and features from a given set of data, which traditionally requires a domain expert to identify. The versatility and robustness of ML algorithms enable them to adapt to virtually any application that has the basic requirement of an arbitrary dataset. These algorithms can be excellent candidates to replace traditional sensing approaches with expanding datasets and poorly understood system models.

A traditional conventional sensor system consists of a sensor element where a threshold limit is triggered for an application outcome (Figure 1a). For instance, a smoke alarm is triggered (application) when a certain amount (threshold limit) of CO₂ is detected by the CO₂ gas sensor (sensor) in the smoke furnes. However, such a simple sensor system may easily fail in a
gaseous environment where unknown gas species may crossstalk with the CO₂ measurements.\[18\] The threshold limit in a conventional sensing system is replaced with highly sophisticated “smart” models created from ML algorithms. The algorithms are based on mathematical and statistical methods that are implemented to analyze, process, and extract crucial information from large complex datasets. Most “smart” systems are evolving to take advantage of the movement toward parallel computation that was democratized with the introduction of modern graphical processing units (GPUs). The extraordinary parallel computational power enables algorithmic models to process thousands and even millions of data points simultaneously, and therefore significantly reduces the calculation time cost in data analysis and model training.\[19\] Consequently, the smart model becomes a crucial component and forms the foundation of an ML-enabled smart sensor system.

A general block diagram of an ML-enabled smart sensor system is shown in Figure 1b, the traditional threshold limit function from a conventional sensor system is replaced by the data preprocessing and smart model. The system preprocesses sensory measurements before being passed through to a trained ML-based smart model, which further processes and analyzes the sensory information that is specific to the application.

There are two types of smart models created for smart sensor systems, which are dictated by the sensor technology and the application requirements. The smart models fall under two categories in which they are trained to either solve for classification- or regression-based problems. A regression-based smart model implements predictive ML algorithms to correlate the relationships between sensory data and future values (e.g., future blood glucose level prediction based on past measurements\[20\]), whereas a classification-based smart model implements categorizing ML algorithms to identify and separate sensory data with discrete class labels (e.g., identifying cancer and noncancer patients from gas sensor measurements\[21\]).

A reliable smart model must be trained thoroughly with high-quality training datasets before being implemented into the smart sensor system.\[22\] A general block diagram representing the smart model training process is shown in Figure 1c. The model training development is split into two phases: data preprocessing and model training. Application-specific sensor data are compiled into unprocessed raw datasets and then processed by the data preprocessing step in Figure 1c-1, in which the unprocessed datasets will involve preprocessing tasks such as handling null values, standardization, or encoding discrete class labels among others. Feature extraction\[23\] is a crucial step in developing a smart model as it extracts the most relevant data that can provide useful information for model development. Features can be either manually selected by a domain expert or extracted by ML algorithms. Then the relevant features are annotated and further recompiled into two subdatasets for training, validating, and testing the smart model (Figure 1c-2). With the given training dataset from the previous step, the smart model can be trained with ML algorithms to solve either classification or regression problems (Figure 1c-3). The training stops when the smart model reaches acceptable levels in which the performance is validated against a validation and test dataset (Figure 1c-4). In addition, the application requirements may increase the complexity of the smart sensor system by training and combining one or more ML algorithms into the smart model, which may have a multiclassification or multiregression application outcomes.\[24,25\] Furthermore, the ideal ML algorithms selected to create smart models and data processing methods are dictated by sensor technology and application requirements.

In this Review, we first summarize the well-known ML algorithms used to create smart models in an ML-enabled smart sensor system for practical sensing applications. Under this section, the ML algorithms are divided into two categories: classic non-neural network (non-NN) algorithms (e.g., Principal Component Analysis [PCA], Support Vector Machine [SVM], Random Forest [RF]) and the state-of-the-art neural network (NN) algorithms (e.g., Backpropagation Neural Network [BP-NN], Recurrent Neural Network [RNN], Convolution Neural Network [CNN]). Furthermore, the general overview process of training a smart model for each ML algorithm is discussed with application examples. Subsequent sections summarize the novelty of ML-enabled smart sensor systems in practical sensing applications under two major categories that are based on their sensing principals:
physical and chemical sensors require some form of physical, chemical, optical, or biological response that results in a measurable output signal while visual imaging sensor capture 2D images which requires image processing at a pixel level. Under both sections, each application investigated is discussed in terms of sensor technology, generation of training and validation datasets, data processing and analysis, smart model training and development process, ML algorithms implementation, application outcomes, among others. Two tables are compiled summarizing the key elements of smart sensor systems investigated in this article under their respected categories. Finally, the review concludes by identifying the major trajectories and challenges faced by existing research activities and proposing opportunities for new research where gaps exist in current efforts.

2. ML Algorithms in Smart Sensor Systems

As shown in Section 1, the foundation of a smart sensor system is the ML-based smart models that are developed to solve classification or regression problems for specific sensing applications. The following related discussion is separated into two distinct ML groups: NN and non-NN algorithms. Non-NN algorithms can be as simple as linear regression (LR) models or as complex as statistical analysis methods such as PCA,[26] SVM,[27] RF,[28] among others.

In contrast, NN algorithms are highly efficient in terms of feature learning and extraction and require less manual input when compared with non-NN ML algorithms. A NN learns abstract features from a given dataset by the activation of neuron nodes that are within a NN, similar to a biological neural circuit. Each neuron node is mathematically activated[29] by the input stimulus data that are propagated throughout the NN, in which the outcome of the NN is a result of active neurons that are stimulated by the dataset. NN algorithms[30] such as BP-NN,[31] RNN,[32] and CNN[33] have shown major advancement and breakthroughs in the past decade for demonstrating practical applications in the field of computer science,[34] medicine,[35] and engineering.[36]

In general, non-NN algorithms are more complex to configure and require more manual involvement to fine-tune ML parameters to achieve application outcomes. This can be attributed to the need for domain knowledge of the features in a given sensor dataset to accurately develop an application-specific non-NN smart model. Furthermore, the features can be either manually selected by a domain expert or “automatically” extracted by other ML algorithms.[23] In this section, we discuss the importance of training, validation, and testing datasets used in developing a smart model. Subsequent sections summarize the core concepts of the aforementioned non-NN and NN algorithms with the application.

2.1. Training, Validation, and Test Datasets

As discussed in Section 1, the generation of datasets is a crucial process in creating a smart model. As opposed to a conventional sensing system where unprocessed data are obtained directly from sensor measurements for threshold limit-based applications, the data generated from sensors are collected and compiled into an unprocessed dataset. The unprocessed dataset undergo a data preprocessing step (Figure 1c-1), where the features relevant to the application are extracted from the dataset, which will be used to create the smart model. Once adequate features are extracted, they are recompiled into two subdatasets, 80% reserved for training the smart model, whereas the remaining 20% for validating and testing the smart sensor system.

The smart models learn abstract features from the training dataset, which comprised annotated data labeled that references relevant to the smart sensor application, i.e., a label for a gas sensor response of CO₂ at specific t-time and correlates to a known type of analyte. Validation and test datasets are used to evaluate the performance of the smart sensor system by introducing data
that the smart model has never seen before. The smart sensor system performance is evaluated for classification accuracy and application functionality.

Furthermore, the overall quality of the dataset may affect the model performance and application outcome to solve classification or regression problems accurately. The amount of data required in a dataset involves a trade-off; insufficient or excessive training datasets may cause underfitting or overfitting in the smart model, which consequently affects the smart model’s ability to capture the underlying structure of the sensor measurement or fail to fit the sensor measurement, respectively.[37,38]

A typical process of generating labeled datasets from sensor measurement is shown in Figure 2, in which an example is shown of the training and validation and test datasets being generated from sensor measurements of two types of analytes denoted as analyte #1 and analyte #2 (Figure 2-1). The multiple sensor measurements of both analytes are compiled into unprocessed raw datasets (Figure 2-2) and undergo a data preprocessing step where relevant features to the application are extracted from the dataset (Figure 2-3). The extracted features are annotated and are to further recompiled into two subsets where the training dataset retains the annotated labels, while the validation and test dataset have annotated labels omitted (Figure 2-4).

### 2.2. Linear Regression

LR models the linear relationship between a sensor measurement and the sensing target, and it is the foundation for many other LR-based algorithms such as multivariate linear regression (MLR) and partial least squares (PLSs). A linear equation is calculated as the best-fit between the sensor measurement and the true sensing target, whereas MLR applies the same principals but with more than one sensor measurement.

An example in Figure 3a shows both MLR and PLS algorithms attempting to predict analyte concentrations from sensor data (Figure 3a-1). From Figure 3a-2, it is clear that MLR analyte prediction becomes less accurate as it attempts to fit multiple linear equations (red lines) in-between the sparsely scattered sensor data. In contrast, the PLS model transforms the sensor data into PLS component values that are based on the maximum variance between sensor data and predicted outcome. From Figure 3a-3, the transformed PLS components are linearly clustered together compared with MLR; thus, PLS provides a more accurate predictive model for analyte concentration.

### 2.3. Principal Component Analysis

PCA has been one of the commonly used dimension reduction algorithms for decades[39] and has now been configured to identify clusters of patterns.[26] By transforming the datasets into a new set of uncorrelated values, which are also called principal components (PCs), PCA can reduce the dimensions of large datasets, and then extract the main features without losing information. Typically, the first two or three PCs (PC1, PC2, and PC3) are the highest explained variances compared with subsequent PCs (PC4, PC5, PC6). They, therefore, can be visualized on a PCA biplot on a 2D or 3D plane.

From Figure 3b, a PCA algorithm-based smart sensor system identifies the two groups of analytes and a third unknown mixed sample using the analyte gas sensor measurement and training datasets from Section 2.1. The PCA model transforms analytes into PCs and ranks them into separated clusters on a 2D plane based on variance distance presented in PC1 and PC2 domain (Figure 3b-1).[26] As shown in Figure 3b-2, the complete model classifies analyte #1 and #2 as $Y_1$ and $Y_2$, respectively. In the
smart application, the trained PCA model is capable of identifying a third unknown analyte consisting of a mixture between analyte #1 and #2. The gas measurements from the unknown mixture are illustrated in red clusters and show the analyte relationship between analyte #1 and #2.

2.4. Support Vector Machine

SVM is an algorithm that solves binary classification problems. SVMs can be configured to solve “multiclassification” problems mapping one input to many classification outcomes.
or to tackle predictive problems which are typically one-to-one. The SVM approach separates input datasets, which are represented as nD support vectors, into two classes with an (n – 1)D plane called hyperplane by calculating the largest separation among the input datasets. The hyperplane divides the input vectors into groups linearly or nonlinearly with a well-chosen mathematical “kernel trick,” in which the boundary distance between the hyperplane divider and input vectors is also called margin. The larger the margin distance between the groups, the more accurately the datasets can be classified.

An example in Figure 3c shows an SVM model performing a classification of a 2D dataset (two analytes), in which the training scheme from Section 2.1 is applied. In this example, a binary classification of analytes #1 (blue) and #2 (orange) is used. SVM transforms the 2D datasets into support vectors that are separated by the initial (2 – 1)D hyperplane divider (Figure 3c-1). The hyperplane is optimized through the kernel trick algorithm, which maximizes the boundary distance between the divider and the two analytes (Figure 3c-2). Once an SVM model is trained, the binary classification output, denoted as Y1 and Y2 for analyte #1 and #2, respectively, is verified with validation and test datasets. The final smart model is then integrated into a smart sensor system which can analyze and identify new analyte measurements and automatically classified them into two distinct groups (Figure 3c-3).

2.5. Decision Trees and RF

As implied in the name, Decision Trees (DT) solves classification and regression problems through a decision-making process that is organized in a tree-like hierarchy. The highest-level decision node (root node) sits at the top level of the hierarchy and represents the entire dataset population. The root node splits the dataset into a subtree (n-branches), where n-decision nodes represent a portion of features found in the dataset and subsequent n-decision nodes split features until it cannot be further separated (n-leaf node). At each n-decision node, a threshold is calculated to proportionally split the most commonly found features based on the Gini Impurity metric.

The foundation of the RF algorithm is built upon n-DTs and inherits the classification and regression attributes from the DTs. At the highest hierarchical level in the RF model, the entire dataset population is randomly selected by a Bagging algorithm and distributed into subsets that are proportionally equal to n-DTs in the forest. The outcome of each n-DT is aggregated by the RF algorithm and used to solve classification problems by “majority votes” or predictive problems by averaging the result.

The training scheme from Section 2.1 is used to demonstrate an RF-based smart sensor system (Figure 3d), in which adds a third analyte (Figure 3d-1) to the training dataset for categorizing multiple types of analytes. The DTs in the RF are trained to learn features from analyte #1, #2, and #3 sensor datasets (Figure 3d-2), which are collected by the RF model to model the classification for the analytes. On completion of the training, the RF model classifies analytes #1, #2, and #3 as Y1, Y2, and Y3, respectively, and is verified with validation and test datasets. The final-trained RF model is embedded into the smart application where new analyte samples can be categorized either into analyte #1, #2, or #3 groups (Figure 3d-3).

2.6. Artificial Neural Network

The past decade has witnessed significant improvements in increased parallel computational power in GPUs used to train NN, which has resulted in the growing interest in deep learning algorithms and their practical applications. Artificial neural network (ANN) is inspired by biological neurons found in the physiology of the human brain and has become one of the most popular deep learning methods nowadays. The theory behind ANN is to have a network of interconnected logical computation nodes (or artificial neurons) that are individually activated by mathematical functions and weights that are determined by sample data feedforward into the ANN.

2.6.1. Feedforward Neural Network

A standard feedforward neural network (FNN) configuration, shown in Figure 4, comprises three layers: an input layer, a hidden layer, and an output layer. A typical application will generally initialize the FNN nodes with randomized weights and bias values. Data are fed into the input layer nodes of the FNN denoted as X1 and X2, where the data are propagated throughout the NN. In the hidden layer, each n-neuron node is activated by a mathematical function (Fi) based on the sum of the interconnecting nodes (Σi). The neuron nodes are active only when a specific threshold is reached and is determined by mathematical functions such as Hyperbolic Tangent or Logistic sigmoid. Eventually, the data will propagate to the nodes at the output layer, where further mathematical functions are applied to solve classification or regression problems. In addition, the performance of an FNN can be fine-tuned by manipulating the weights and bias in each neuron in the hidden layer, which is a less practical training strategy compared with other NN algorithms that will be discussed in the next section.

![Feedforward neural network](image-url)

**Figure 4.** A standard FNN comprised three layers: an input layer, a hidden layer, and an output layer. Sensor data are propagated throughout the FNN in one direction, where data enters into the input layer and comes out from the output layer. The neuron nodes are stimulated by the propagated data in the FNN, which activates the final nodes in the output layer where the results are shown.
2.6.2. Back Propagation Neural Network

The BP-NN[31] is built on the same foundation as that of an FNN and subsequently inherits the neuron nodes functionality and the feedforward data propagation attributes. The key difference between the FNN and the BP-NN is that a BP-NN model can be effectively trained using a technique called backpropagation, which can significantly reduce the training cost. A BP-NN propagates data forward from the input layer to the output nodes is identical to FNN. However, at the output layer, the errors in a BP-NN are measured with a loss function that compares the network outputs with the desired values, and then these errors propagate back through the network to reveal the error contributions of inputs at each neuron node until it reaches back to the input layer. With the error contributions computed, the BP-NN algorithm can automatically adjust the neuron weights across the network to reduce the error gradients until the output layer reaches acceptable convergence values for the classification or regression application.[46,47]

A BP-NN-enabled smart sensor system example is shown in Figure 5, which classifies two types of collision events from an accelerometer sensor. The model in this system is configured with $m$-hidden layers by $n$-neuron nodes with different types of activation functions and initial starting weights (Figure 5-2). The analytes from the training dataset scheme in Section 2.1 are replaced with an accelerometer dataset and are propagated into the BP-NN’s inputs as $X_1$ and $X_2$ (Figure 5-1). The BP-NN learns the collision features by activating hidden $n$-neuron nodes and backpropagating the predictive error (red arrows) back into the NN (Figure 5-3). When model training is complete when the predictive error rate is minimized, in which the model classification outcome denoted as $Y_1$ and $Y_2$ (Figure 5-4) is verified with validation and test dataset (Figure 5-5) for collision event #1 and #2, respectively. The improved training efficiency of the BP-NN has resulted in the adoption of NN-based smart sensing as a practical technology in real-world applications. This is particularly true of image-based sensing with rich input fields. The BP-NN has also formed the foundation of many other NN algorithms, such as RNN and CNN.

2.6.3. Recurrent Neural Network

A RNN shares the same attributes as an FNN. However, the difference is that in each $m$-recurrrent hidden layer, the internal recurrent $n$-neuron nodes behave like a memory element.[48] A sequential series of recurrent neuron nodes can retain activation information from the previous hidden states and combines these with the present input state. This preservation of the relationship between preceding elements and the current input allows the algorithm to perform exceptionally well with temporal or sequential data such as predictive texts or voice recognition applications.[49–51] Long short-term memory (LSTM) and gated recurrent units (GRUs) are state-of-the-art variants of RNNs due to their excellent performance in model training, and therefore have been widely applied into many smart applications.[52,53]

An example of a smart sensor system in Figure 6 shows a standard feedforward RNN using the accelerometer training dataset from Section 2.6.2 for a predictive model of collision events. The RNN training process is visualized by Greek letters, $\alpha$, $\beta$, $\delta$, $\gamma$, and $\zeta$ to represent the training data shown in (Figure 6-1). The RNN model highlighted in the orange box is configured to retain past information from six sequential data inputs denoted as $X_1[1,6]$. The expanded view of the RNN layer is shown in Figure 6-2, where the inner hidden RNN nodes are exposed. The first iteration $i = 1$ shown in Figure 6-3, concatenates ($C_1$) the weighted input data from $X_i$ with weighted input data of previous hidden state ($h_0$), where the initial value is zero (Figure 6-4). The concatenated sum of the product ($C_1$) is used for the activation of the function node ($f_i$) to produce a single prediction outcome denoted as output $Y_1$. The output result is retained for the next iteration hidden state $h_1$ (Figure 6-5). After several iterations, the RNN model learns to predict collision feature $\gamma$ by retaining information from three memory state $\alpha$, $\beta$, and $\delta$. The RNN model outcome is the ability to predict collision event $\gamma$ based on three sequential input of $\alpha$, $\beta$, and $\delta$ (Figure 6-6).

![Figure 5](image_url)

**Figure 5.** An example of a BP-NN-based smart sensor system. The flow diagram illustrates the overview process from creating the BP-NN smart model to a smart application. The BP-NN model learns abstract features by backpropagating the output error back into the NN, which adjusts the neuron nodes’ weights and ultimately reduces the unnecessary stimulated neuron nodes that are not relevant to the sensor application.
2.6.4. Convolution Neural Network

CNNs are by far the most successfully implemented NN algorithms, which has many real-world applications, particularly in image classification such as object detection, facial recognition, and image segmentation. In the past decade, CNN has seen several major breakthroughs in image processing applications and produced several specialized high-performance frameworks such as AlexNet, YOLO9000, U-net, GoogLeNet, VGG, among others. A standard CNN model comprised three building blocks, which are the data input, feature learning, and training and classification (Figure 7). A CNN excels at processing and analysis of images, in which the image data propagated into the NN are first decomposed and converted into a multidimensional array in which key abstract features are extracted. Subsequently, the abstract features are flattened and propagated into a NN to be trained for classification- or regression-based applications.

An example of a CNN-based smart sensor system, trained to identify and classify images of a dog and cat, is shown in Figure 7. The CNN model is trained with an image dataset (Figure 7-1) containing images of a variety of cats and dogs in the different breeds, colors, sizes, among others. During the feature extraction process, the CNN converts the training data image into a multidimensional array for the convolution layers to learn to extract significant features from the array by image convolution rectified linear units (ReLU) and then pooling (Figure 7-2). These two steps break the image down into smaller and more organized feature maps. The final arrays of feature maps are “flattened” before being propagated into the NN which is trained to classify the image based on these abstract features—in the case of Figure 7 to decide whether the image is that of a dog or a cat. The activated nodes stimulated throughout the NN by the abstract features are collected at the output layer where a classifier algorithm such as Softmax is implemented to determine the class of animals (Figure 7-3). Over a series of epochs, the CNN model learns to identify images of dogs or cats, and the final model can be used for real-time detection and classification (Figure 7-4).

3. ML-Enabled Smart Sensor Systems in Practical Sensor Applications

There have been many successful implementations of ML “smart” models with demonstrated data processing capability and analysis of a tremendous amount of data points. In this section, we present a summary of a wide range of practical sensor
applications that integrate ML algorithms with a variety type of sensor technologies to give insight into the novelty of various smart sensor systems. The section is divided into two parts based on the types of sensors used in the smart systems for discussion: 1) physical and chemical sensing for detecting the physical and chemical parameters of target objects or the surrounding environment; and 2) visual sensing that uses image sensors that focuses purely on processing images at the pixel level.

3.1. Smart Physical and Chemical Sensing Systems

In this section, practical smart sensor systems and their implementation of ML algorithms are categorized and discussed in three application areas: health and disease management, environmental pollution management, and precision agriculture and food science. The sensors in physical and chemical sensing systems require interaction with the target environment, which alter the chemical or physical properties of the sensors themselves to produce measurable signals. These measurable signals are accumulated into datasets that incorporate ML algorithms to processes and to develop smart models for the specific applications. In addition, the smart model performance can be further enhanced and improved by choosing ML algorithms for further extraction information from datasets that do not exhibit obvious patterns before smart model development. For each smart sensor system the choice of ML for each smart sensor system is assessed on an application-to-application basis in which the followings are considered: the type of sensor technology, application requirements, sensor dataset, smart model type, among others. In addition, the methodology developed for each case is unique to the practical sensing application and is most likely not applicable to other smart sensor systems.

A brief summary of the physical- and chemical-based smart sensor systems is shown in Table 1. The table highlights and compares the key characteristics of each smart sensor, such as the choice of ML algorithms used for specific sensor technology in an application area. The table lists the application areas (e.g., health and disease management), the sensor type (e.g., gas sensor or accelerometer), smart model type (e.g., classification or regression), and a short description of the application.

3.1.1. Health and Disease Management

Early disease detection and diagnostics are major areas of interest for health and disease management. The two most commonly used physical- and chemical-sensors in smart sensor systems are electronic chemical nose and wearable sensors. The electronic nose-based smart sensor systems under this section look at analyzing and detecting diseases such as cancer detection from exhaling breathe analysis. In contrast, smart sensor systems with wearable sensors aim at detecting and classifying movement disorders from physical movement.

Metal-oxide (MOX) gas sensing elements are the key building blocks of the electronic chemical nose. Surface properties of MOX enable gas sensing elements to be specifically sensitive to target gases and volatile organic compounds (VOCs), although their cross-selectivity with other interfering analytes is relatively low. Thus, forming an electronic chemical nose composed of multiple MOX gas sensing elements provides a relatively comprehensive identification of many gases and VOCs, which are biomarkers correlated to diseases such as repository diseases, diabetic diseases, bowel cancer, and prostate cancer.

When coupled with ML algorithms, such an electronic chemical nose can diagnose these types of diseases in a noninvasive, convenient, and economical manner compared with those of traditional methods. Saidi et al. fabricated an electronic nose-based smart sensor system consisting of 6 MOX sensing elements to identify chronic kidney disease (CKD), diabetes mellitus (DM), high creatinine (HSHC), and healthy subjects with low creatinine (HSLC) patients from their breathe samples. The model developed for this system implements PCA to discriminate among the four disease subjects, whereas gas chromatography-mass spectrometry (GC-MS) is used to validate the results. A total of 264 breath samples were collected from two groups at clinical trials. In the first group, 216 breathe samples from healthy subjects and patients with CKD and DM were used.
Table 1. Practical sensing application summary of physical and chemical sensing in smart sensor systems.

| Sensor type                                      | ML/a) | Application area | Smart model type | Application description                                                                 | Ref. |
|-------------------------------------------------|--------|------------------|------------------|----------------------------------------------------------------------------------------|------|
| Accelerometer, gyroscope                        | LSTM/b) |                  | Classification   | Detecting tremor signatures found in PD patients                                        | [62] |
| Blood glucose, blood pressure, heart rate       | RNN/c) |                  | Classification   | Real-time classification diabetic patient and future glucose prediction                  | [20] |
| Accelerometer                                   | RNN/d) |                  | Regression       | Smartwatch, Patient prediction of fall detection from a smartwatch                      | [63] |
| MOx gas sensor                                  | PCA/e) |                  | Classification   | Identifying patients with chronic kidney from exhaled breath                            | [21] |
| MOx gas sensor                                  | FNN/f) |                  | Classification   | Screening for lung cancer from exhaled breath                                           | [64,65] |
| Gold-nanoparticle gas sensor                    | MLR/g) |                  | Classification   | Detecting 17 types of diseases from breath samples                                      | [66] |
| MOx gas sensor                                  | PCA/h) |                  | Classification   | Identifying asthma patients by analyzing VOCs from breath profiles using electronic nose| [67] |
| Gold-nanoparticle gas sensor                    | PCA/i) |                  | Classification   | Classifying 3 and stage 4 lung cancer patients from healthy controls                    | [68] |
| Raman spectroscopy                              | PCA/j) |                  | Classification   | Identifies types of gastric diseases from blood serum                                    | [69] |
| Accelerometer, gyroscope                        | LSTM/k) |                | Classification   | Predicting the severity of Parkinsonian tremors in free body movement                   | [70] |
| Accelerometer, gyroscope                        | RF/l)  |                  | Classification   | Real-time detection and classification of FoG events                                     | [71] |
| MOx gas sensor                                  | PCA/m) |                | Classification   | Classifying storage quality of Chinese pecans and predicting fatty acids                 | [72] |
| MOx gas sensors                                  | SVM/n) |                  | Classification   | Beef quality assessment using electronic nose                                          | [73] |
| NIR spectroscopy                                 | PCA/o) |                  | Regression       | Quality assessment of potato in long-term storage                                        | [74] |
| Electrochemical biosensor                       | FNN/p) |                  | Regression       | Predicting nutrients in the soil (nitrate and phosphorus) using NIR wavelength (700–2500 nm) | [75] |
| MOx gas sensor                                  | BPNN/q) |                  | Classification   | Identifying of wine vintage, fermentation process, production, and varietals from expelled odor | [76] |
| RGB–NIR sensor                                  | CNN/r) |                  | Regression       | Predicting crop yield from data obtained from RGB–NIR spectrum                          | [77] |
| MOx gas sensor                                  | SVM/s) |                  | Classification   | Identifying and predicting the number of tainted milk adulterants (formalin, hydrogen peroxide, and sodium hypochlorite) | [78] |
| MOx gas sensor                                  | BPNN/t) |                | Regression       | Classification of common indoor gas pollution (formaldehyde and ammonia)                | [79] |
| Temperature, humidity, pressure, wind speed     | RF/u)  |                  | Regression       | Predicting PM$_{2.5}$ and PM$_{10}$ pollution across provinces in China                 | [80] |
| Electrochemical biosensor                       | FNN/v) |                  | Classification   | Identifying water pollutants Escherichia coli, Shewanella oneidensis, and Methylcoccus capsulatus | [81] |
| MOx gas sensor temperature, humidity, precipitation, radiation, wind speed | FNN/w) |                | Regression       | Short-term prediction of NO$_2$ and NO$_x$ concentration based on past meteorological sensors and NO and O$_3$ found in urban cities | [82] |
| NIR optical sensor                               | PCA/x) |                  | Classification   | Identifying urban traffic pollution from reflectance of the leaf (Carpinus betulus L. saplings) | [83] |
| MOx gas sensor, temperature, humidity           | FNN/y) |                  | Classification   | Classification of three types of pollution: traffic, urban and photochemical pollution   | [84] |

a) Machine learning algorithm; b) Electronic nose; c) Health and disease management; d) Long short-term memory recurrent neural network; e) Principal component analysis; f) Feedforward neural network; g) Multiple linear regression; h) Recurrent neural network; i) Random forest; j) Precision agriculture and food science; k) Near-infrared spectroscopy; l) Backpropagation neural network; m) Red–Green–Blue–NIR spectroscopy; n) Convolutional neural network; o) Environmental pollution management; p) Particulate matter 2.5/10.

For training data, and in the second group, 48 samples were collected to test and evaluate the system. Eighteen variables were selected to train the PCA model, which consists of three features (i.e., dynamic conductance slope, area under the curve [AUC], and delta conductance) extracted from 6 MO$_x$ sensing elements. The PCA results in Figure 8a shows a clear separation between training dataset and blind dataset from CKD, DM, and healthy groups.

Apart from the use of traditional tin dioxide (SnO$_2$) as sensitive materials, Peng et al. [68] fabricated nine gold...
nanoparticle-based gas sensing elements functionalized with different organic materials for a smart sensor system. Their system can discriminate against healthy subjects and lung cancer patients by analyzing specific VOCs that are correlated to lung cancer. Breathe samples were taken from 40 patients with stage-3 and stage-4 lung cancer, and 56 healthy individuals for

Figure 8. Smart applications use electronic nose and gas sensors in health and disease management. a) PCA model and electronic nose used to identify patients with kidney disease, diabetes, and healthy controls. Reproduced with permission.[21] Copyright 2018, Elsevier B.V. b) Heatmap of extracted sensing features from 20 custom gold nanoparticle sensors. The heatmap shows the sensitivity of each feature against disease class. Reproduced with permission under the terms of ACS AuthorChoice license.[66] Copyright 2016, The Authors. Published by American Chemical Society.
training the PCA model, in which delta resistance over delta baseline was extract from the sensing elements as the input feature for PCA model. The outcome of the PCA model showed two distinct clusters between control and cancer patients. Furthermore, a GC-MS was used to identify VOCs from to healthy and lung cancer patients to verify the simulation data cluster in the PCA model.

Smart sensor systems composed of arrays of modified gold nanoparticle-based gas sensing elements can also be used to diagnose disease beyond cancers.[66] Figure 8b shows the heatmap of extracted sensing features obtained from 20 different sensing elements, some of which show higher sensitivity to types of diseases than others. The final system based on discriminant factor analysis benchmarks in a blind study where the smart sensor system can achieve a classification accuracy rate of 86% for each class.

Wearable physical sensors are devices that are generally attached to the end-user to extract movement data, which are composed of accelerometers, gyroscopes, and linear accelerators sensing elements. Wearable physical sensors can be used for smart health management[93] and disease[94] associated with movement. In particular, they are highly suitable candidates for the detection and monitoring of Parkinson’s disease (PD).[95] Many applications utilize NN to detect early signs and classify patients with PD. Hssayeni et al.[70] develop a system that can automatically estimate the severity of Parkinsonian tremors in a user by monitoring their free body movement. Their system in Figure 9a shows an LSTM RNN model trained to detect PD tremors from sampling accelerometer and gyroscope measurements in 5 s intervals, in which 78 distinct features were selected to predict the PD severity.

Figure 9. Smart applications use wearable sensors in health and disease management. a) LSTM NN is used to detect tremors in PD patients from body and ankle sensors. Reproduced under the terms and conditions of the CC-BY license.[70] Copyright 2019, The Authors. Published by MDPI, Basel, Switzerland. b) Block diagram of a real-time system that predicts future blood glucose levels and classifies diabetic patients using LSTM NN. Reproduced under the terms and conditions of the CC-BY license.[20] Copyright 2018, The Authors. Published by MDPI, Basel, Switzerland.
In addition, the freezing of gait (FoG) is a common motor symptom associated with PD patients. Alrich et al. used a waist-worn triaxial accelerometer to detect FoG episodes experienced by PD patients. FoG data collected from 15 patients were used to develop an SVM model to classify FoG movements. The final SVM model was tested with an additional five patients achieving classification accuracy of 98.7%. Another example brings the detection of FoG events online in which real-time detection of FoG episodes can be assessed by a triaxial accelerometer found in smartphones. Movement data are not just limited to FoG motor symptoms; other illnesses such as Alzheimer’s disease can be identified by monitoring accelerometer sensor data. Varatharajan et al. successfully demonstrated this capability by creating an SVM model that can detect early signs of Alzheimer’s disease using wearable sensors attached to participants.

In addition to the wearable sensors for measuring the physical parameters of the human body, there are a few cases to apply wearable biochemical sensors into the smart sensor systems. An electrochemical blood glucose sensor was used by Alfant et al. to develop a cloud-based smart sensor system to provide real-time diabetes classification and prediction of future glucose levels (Figure 9b). Heart rate, blood pressure, and blood glucose measurements were sent to the cloud, where a trained LSTM RNN model processed the data and made diabetes classification and prediction for future glucose levels.

Health management systems look at identifying physical activities associated with movement, which are a part of “smart assistive technologies” found in rehabilitation and health support applications. Physical activities such as sitting, walking, and lying are among the number of movements identified by a smart model created by Jiang and Yin. The system converts the raw measurements of a linear accelerometer, accelerometer, and gyroscope into an image before training an NN. The 1D raw sensor data are stacked adjacent in rows to form a 2D signal image, in which the image is further processed by a discrete Fourier transform to create the activity images. The final activity images are used to train a CNN model to classify the three types of physical movement (sitting, walking, and lying). Other forms of smart sensor systems exploit accelerometer data from a smartwatch such as an RNN-based smart sensor system that predicts fall events for the elderly. The system implements a GRU type RNN that was trained to predict fall events in 1.28 s sampling window. The system shown in Figure 10a deploys the predictive model into the cloud for the real-time fall classification decision.

Stereotypical motor movements (SMMs), which are commonly associated with autism spectrum disorders, can be detected in real time. The real-time measurements from IMU are sent into the cloud, where an NN model makes the classification decision of SMMs (Figure 10b). The CNN/LSTM model was trained with physical activity measurements from IMUs in which a CNN was used to extract features from IMU datasets. The extract features were fed into an RNN in which the data were further processed for the final decision of classifying SMMs events.

Wearable physical sensors also have the potential of assisting self-rehabilitation caused by sports injuries. A smart sensor system created by Lin et al. identifies the physical activities relating to a frozen shoulder injury and assists with the rehabilitation process. The system uses a wireless sensor node (WSN) that has a built-in triaxial accelerometer and gyroscope, which are attached to the upper arm and wrist. Their BP-NN model classifies into five specific exercises: scapula exercise, Codman’s pendulum exercise, finger wall-climbing exercise, back shoulder circling exercise, and towel exercise. Classification accuracy achieved by the BP-NN model reaches up to 95% for basic swinging and stretching exercises; however, the smart model reached as low as 60% for exercises that involved spiral rotation.

3.1.2. Environmental Pollution Management

Environmental pollution management enabled by smart sensors systems focus on modeling air pollutants and greenhouse gases such as NOx, NO, NO2, CO2, CO, CH4, SO2, and O3. Other hazardous particulates matter such as particulate matter 10 μm (PM10) and PM2.5 are correlated to disrupt the functions of lung and heart.

A system developed by Chen et al. predicts the PM2.5 pollution across provinces within China. The system uses a combination of sensors and an RF model to estimate PM2.5. The model is trained with measurements from temperature, relative humidity, barometric pressure, wind speed, and aerosol optical depth sensors. In addition, wavelength data from hyperspectral images of provinces across China are used to calculate normalized difference vegetation index (NDVI) values, which are also a training parameter. Three categories of pollution: urban, traffic, and photochemical pollution are classified by an NN model developed by Barakeh et al. The NN model takes both sensor data (gas, temperature, and humidity) and time parameters (weekday/weekend) to categorize the pollution type.

In addition, Zhang et al. fabricated an array of MOx composites with reduced graphene oxide-based gas sensors to detect common indoor pollutants such as NH3 and HCHO. The block diagram of the system (Figure 11b) shows the successful combination of BP-NN and graphene MOx sensors to identify and quantify the NH3 and HCHO. Long exposure time to toxic levels of NOx gas concentrations in built-up urban environments has detrimental health effects that can cause children and adults to develop severe repository health problems.

Forecasting and modeling the NOx gas concentrations is crucial to understanding and deploying preventative measures to mitigate the pollutants. An NN system reported by Rahimzadeh predicts the NOx concentration found in Tabriz, Iran, from meteorological (wind speed, wind direction, precipitation, vapor pressure, barometric pressure, temperature, humidity, and radiation). NOx and O3 gas data are used to train the NN model.

A unique approach proposed by Brackx et al. uses plant leaves to measure pollution in an urban environment. The system uses the reflectance measurements from leaf species Carpinus betulus L. saplings to identify different pollution levels in the city environment (Antwerp, Belgium). A PCA model is successfully trained to identify the levels with the reflectance measurements obtained from the Vis–near infrared (NIR) sensor. The PCA results in Figure 11c shows the PCs of high and low traffic pollution that are measured from leaf reflectance at a wavelength range between 450 and 2400 nm.
3.1.3. Precision Agriculture and Food Science

Smart sensor systems in precision agriculture rely on hyperspectral wavelength measurements to calculate the NDVI and enhanced vegetation index (EVI) values. The indexes are key indicators used to evaluate precision farming practices such as crop management, soil assessment, and livestock management.\[5\]

Pantazi et al.\[109\] showed an NN model based on Supervised Kohonen Network framework to estimate future crop growth in wheat fields. The NN model predicts crop yield based on soil sensors that measure total nitrogen content, moisture content, phosphorus levels, pH level, calcium levels, magnesium levels, and cation exchange capacity. In addition, the calculated NDVI value obtained from the hyperspectral images also used as a feature to train the NN. The A color code heatmap illustrating the prediction yield of the wheat field is shown in Figure 12a, the left image shows actual measure yield and the right image shows the prediction from the NN. In comparison, the images show very high spatial similarity to the measured yield, in which the SKN model achieved the highest wheat prediction accuracy of 81.65% compared with the benchmarked counter-propagation ANN and XY-fusion NN models.

Furthermore, a soil sensor-driven PCA model developed by Sithole et al.\[112\] can detect soil organic nitrogen after different stages of N-fertilizer treatments. Unwanted foreign objects in the wheat field, such as barley, maize, deer, and rabbit dropping, are identified by a smart sensor system developed by Ravikanth et al.\[113\]. The system uses NIR measurements captured by hyperspectral image camera to classify the foreign object found in the wheat farm. An SVM was trained with different reflectance data of the unwanted objects collected at the wavelength range between 1000 and 1600 nm. Coopersmith et al.\[114\] evaluated the performance of two ML algorithms: k-nearest neighbor and FNN for binary classification of soil assessment for agriculture planning. The model uses data from solar radiation,
humidity, wind, and temperature sensors to calculate potential evaporation and precipitation. Both models use the same variables (24 h aggregated precipitation, 48 h aggregated rainfall, and 24 h potential evapotranspiration) to classify “wet” or “dry” soil condition with 91–94% accuracy.

Electronic chemical noses are commonly used to assess the quality of food in this category. Jiang et al.[72] developed two models to classify the freshness quality of Carya cathayensis (Chinese pecans) at different storage times and quantitative predictions of its fatty acid profiles. Five distinct features...
(10th-second value, 75th-second value, area values, max response curve, and min response curve) from the 10 MOX gas sensing measurements were used to train a PCA algorithm for build a storage time classification model. A single feature (75th-second value) was selected for the RF regression model to predict fatty acid profiles (oleic, linoleic, palmitoleic, linolenic, palmitic, and stearic acids). A PCA score plot in Figure 12b, therefore, shows the “freshness” cluster of pecans over 20 days. The final model indicates a strong correlation between electronic chemical noses data and GC-MS of acid profiles. Such systems can also be used to assess the quality of wine and identify wine spoilage, alcohol fermentation, or monitor age, in which electrochemical- or MOX-gas sensors can develop a PCA smart model to distinguish and classify a variety of grapes (Figure 12c).[110]

Application in early detection of pathogenic diseases found in fruits is vital to stop the contamination and spreading of the disease. Using the measurements from an electronic chemical nose, PCA can separate the healthy control and fungal-diseased strawberries infected with Botrytis sp., Penicillium sp., and Rhizopus sp.[111] The final system implements an NN model to discriminate between healthy control and fungal-diseased strawberries, in which the model is trained with diseased strawberries and healthy control samples. The results of the model in Figure 12d shows the classification outcome of the bacteria groups and controls from the electronic chemical nose.

Chemicals such as formalin, hydrogen peroxide, and sodium hypochlorite are added to raw milk to increase shelf life. However, food regulators in many countries prohibited these ingredients as it leaves consumers’ health at risk.[115] An electronic chemical nose was used to classify the prohibited adulterants found in milk samples[79] in which a trained SVM model successfully identified the illegal substances (formalin, hydrogen peroxide, and sodium hypochlorite) with a classification accuracy rate of 94.64%.

3.2. Smart Visual Imaging Sensing Systems

Compared with physical and chemical sensing, visual sensing has shown highly successful practical implementation in real-world applications. The input datasets utilized image-based smart models purely focus on analyzing and processing images at a pixel level. In comparison with physical and chemical sensor
datasets, visual imaging datasets primarily comprises 2D images that are typically captured and generated from two types of image sensor: charged-couple device (CCD) or CMOS.[116]

In the past, classic non-NN ML algorithms such as SVM and PCA implement for image processing are complex and inefficient for solving image classification problems.[117,118] However, recent advancements in the past decade have introduced the highly efficient CNN frameworks such as AlexNet,[33] YOLO9000,[54] U-net,[57] VGG,[119] among others. The frameworks are highly specialized methods that process different types of images for a variety of applications such as object detection, image recognition, or image segmentation.

The sophistication of what can be achieved with practical NN algorithms has grown dramatically with the advent of modern high-performance GPUs’ parallel computation. NNs intrinsically require iterative application of very large-scale matrix operations with order proportional to number of neuron nodes, which can be vast. Conducting these computations with a traditional serial CPU can take several days, whereas a GPU can take advantage of parallelization to accelerate this to a couple of hours.[120] CNNs are particularly well suited for image-based smart sensor systems.

The practical sensing applications of visual imaging sensing in smart sensor systems under this section are discussed in four application areas: Object Detection and Recognition, Precision Agriculture, Medical Image Diagnosis and Analysis, and Biomedical Image Diagnostics and Analysis. Furthermore, a brief summary of image-based smart sensor systems is shown in Table 2. The table outlines and compares the key attributes such as the image type and the practical sensing application with other systems that are reviewed in this section. The table lists the application areas (e.g., Precision Agriculture), the image types (e.g., magnetic resonance imaging (MRI) or Fundus), and a short description of its applications.

### 3.2.1. Object Detection and Recognition

Object detection and recognition by smart sensor systems can be found in a variety of real-world practical applications such as surveillance, autonomous vehicles, robotics, consumer electronics, among others. Fast region-based CNN (R-CNN) is a generic object detection framework developed by Girshick[210] and is superseded by the recently improved version known as Faster R-CNN[211] Ji and Learned-Miller[212] applied this framework to solve face recognition problems instead of object detection. The NN is trained on a WIDER dataset[213] that contains 12,880 images with 159,424 faces and benchmarked on the FDDB dataset[214] with 5171 annotated faces in a set of 2845 images. Figure 13a shows an example of facial recognition which highlights faces in red squares. YOLO9000[24] introduced in 2017 is the successor to the YOLO framework and is reintroduced as YOLOv2.[216] YOLOv2 addresses many of the shortcomings, such as large localization and recall errors compared with other Fast R-CNN[210] object detection models. YOLO9000 is capable of detecting and categorizing over 9000 objects by combining ImageNet and COCO datasets hierarchy and optimizing the training of the model on the combined datasets.[217,218]

Figure 13b shows output example of smart sensor system implementing the YOLO9000 framework that detects a wide range of objects and labels it in real-time. Visual sensing systems found in civil engineering such as reported by Cha et al.[215] show that a CNN model can be trained to classify and identify concrete defects from high-resolution images. The smart model’s architecture is shown in Figure 13c and illustrating the model training process for classifying types of concrete damages. The CNN model was trained with cropped images (256 × 256 pixels) of both damage and nondamage concrete. The CNN model achieved up to 98% accuracy of identifying crack defects in high-resolution images (5888 × 3584 pixels). Other types of structural damages found in civil infrastructures can be automatically identified with Fast R-CNN, in which an example of a system shown in Figure 13d identifies two levels of steel corrosion (i.e., medium and high), bolt corrosion, and steel delamination. The system is trained with 2365 images to identify five types of damage and scored classification accuracy of above 85%.[198]

### 3.2.2. Precision Agriculture

Crop segmentation is an important tool in precision agriculture for identifying objects such as crops, land types, foreign objects, among others. The smart sensor system developed by Kussul et al.[193] demonstrates a smart model that identifies major crop types (maize, soybeans, sugar beet, sunflower, and wheat) from remote sensing images. The remote sensing data are made of multitemporal and multisource images captured from satellites in which 19 captured images from vegetation seasons are used to train the several models for classification. A CNN-based model achieved the highest accuracy (85%) in classifying maize, soybeans, sugar beet, sunflower, and wheat types compared with other models in the study. A visual illustration of a CNN smart model for the classification of crops and land types in an agriculture setting is shown in Figure 14a.[219] Studies have shown plants and crops that undergo a prolonged period of stress caused by environmental factors can lead to decreased crop growth.[211] A typical hyperspectral sensor provides wavelength measurements of the sensing target in which a system proposed by Behmann et al.[220] utilizes the wavelength measurements obtained from the hyperspectral images to build an SVM-PCA model that identifies the senescence stages and classifies drought stress plants. A visualization of the vitality map of untreated versus fertilized plants are shown in the left image in Figure 14b, and the PCA classification plot of stress and nonstressed plant images grouped by their type of treatment is shown on the right.

Other types of smart precision farming applications identify foreign objects and unwanted plants. Mohanty et al.[222] showed a system that is capable of identifying different types of weed among soybean crops. The smart sensor system is based on a CNN model utilizing the CaffeNet framework that was trained with 400 annotated images, classified into four types of weed (soil, soybean, weed, and grass). The training data are split in 70:10:20 ratio, in which 70% of the images from the database are used for training, 10% for model validation, and 20% for performance testing. A training image dataset of different types of grass and weeds used to train the CNN model is shown in Figure 15a-i. A visualization showing color-coded heatmaps superimposed on the two soybean plantations compares the classification performance between CNN- and SVM-based smart
model is shown (Figure 15a–ii). The far-left images show two original images of soybean plantations, in which the middle images display the classification results of the CNN-based smart model, and the far-right images show the classification results from the classic non-NN SVM smart model. The CNN-based smart model outperformed the SVM-based model, in which it achieved a classification accuracy of 98% across weed class types (soil, soybean, weed, and grass).[191]

Smart visual sensing systems in agriculture visually detect and identify defects and anomalies in fresh produces and plants.[192,194,223–225] Ferentinou[192] created a system that identifies Black Sigatoka diseases found in banana leaves. The system uses the images of healthy and diseased banana leaves to create a smart model based on a CNN using the VGG framework.[119] The model is trained with an open database containing 87,848 images of 25 different types of plants that are annotated healthy and diseased. The image classification results shown in Figure 15b visualizes the system’s classification process with a percentage certainty highlighted in each image. The final model based on CNN-VGG achieved the highest plant species

| Image type | ML | Application area | Smart model type | Application description | Ref. |
|------------|----|------------------|------------------|-------------------------|------|
| MRI[5]     | CNN| MIDA[5]          | Segmentation     | Detecting, classifying and segmentation of tumors in human a brain | [121–125] |
| Dermoscopy | CNN| Classification   | Classification and identification of multiple types of skin cancer | [126–131] |
| Fundus images | CNN| Classification   | Detecting and identifying hemorrhages, diseases, and abnormalities in the human retina | [132–137] |
| X-ray, MRI, US[4,5] | CNN| Classification   | Breast cancer tumor, tissues, and lesion classification and segmentation | [138–154] |
| X-ray      | CNN| Classification   | Detection and identification of chest anomalies, diseases, and illnesses (tuberculosis, nodule detection) | [155–160] |
| CT[4]      | CNN| Classification   | Nodule detection and Interstitial lung disease classifications | [157,161–168] |
| CT, US, MRI[4,6] | CNN| Classification   | Identifying and Segmentation of diseases and healthy organs (bladder, prostate, liver, kidneys, pancreas, and colon) | [169–190] |
| Hyper spectral images | CNN| PA[4]          | Classification   | Identifying and separating unwanted plants (weeds, grass, and soil) from soybean crop | [191] |
| Photographs | CNN| Classification   | Diagnosis and classification types of diseases in plants | [192] |
| Satellite images | CNN| Classification   | Detecting and segmentation of land and crop types (water, grassland, bare land, winter wheat, soybeans, maize, sunflowers, and sugar beet) | [193] |
| Photographs | CNN| Classification   | Identifying damaged sour lemons | [194] |
| Photographs | CNN| Segmentation      | Automatically segment apples from images and predicts the yield | [195] |
| Photographs | CNN| Regression       | Predicting agricultural waste (vegetable, livestock, and straw) | [196] |
| Photographs | CNN| ODR[4]          | Classification   | Object detections and facial recognition | [54,55] |
| Satellite images | CNN| Classification   | Detection and classification of oil spills found in open oceans | [197] |
| Photographs | CNN| Classification   | Classification of civil structural damage types (crack, corrosion, and rust) | [198] |
| Video      | CNN| Classification   | Real-time classification of pedestrians | [199] |
| TEM[6]     | CNN| BIDA[6]          | Classification   | Classification of 15 types of viruses | [200] |
| FL[4]      | CNN| Classification   | Reconstructed holograms used to detect the antigens attached to HSV | [201] |
| H&E, RTI, PC[4,5] | CNN| Classification   | Detection of mitosis cell images for diagnosis breast cancer applications | [202–206] |
| IFL[6]     | CNN| Classification   | Human epithelial-2 cell image classification for the diagnosis of autoimmune diseases | [207] |
| H&E[5]     | CNN| Classification   | Detection and classification of different of infrared nuclei for colon cancer diagnosis | [208] |
| H&E[5]     | CNN| RNN             | Classification   | Detection of cancerous cells in prostate, skin, and axillary lymph nodes using whole image slides | [209] |

[a]Magnetic resonate imaging; [b]Medical image diagnosis and analysis; [c]Ultrasound image; [d]Computed tomography; [e]Precision agriculture; [f]Object detection and recognition; [g]Transmission electron microscopy images; [h]biomedical image diagnosis and analysis; [i]FM images; [j]Hematoxylin and eosin staining microscope images; [k]Real-time images; [l]Phase contrast microscope images; [m]Immunofluorescent microscope images.
classification accuracy of 99.53% compared with other CNN models benchmarked in the study (AlexNetOWTBn, Overfeat, AlexNet, and GoogLeNet).

3.2.3. Medical Image Diagnosis and Analysis

Cancerous cells without early detection can lead to serious health issues that could result in death, and it is crucial to detect and diagnose cancerous cells at the very early stages when it is treatable and has not spread throughout the body. Medical practitioners face daily challenges in identifying and diagnosing patients with cancer-related symptoms, where the limitation of traditional techniques often results in detecting cancer at a much later stage. The smart sensor system can potentially replace medical practitioners in the diagnosis of diseases such as the skin cancer classiﬁcation model. This method significantly reduces the training time, and classiﬁcation errors as the GoogleNet model has previously learned how to separate 1000 object categories from 1.28 million images that are not related to skin cancer. A dataset contains 129 450 clinical images of skin lesions with 2032 annotated disease skin disease classes, in which 127 463 images are reserved for training and 1942 biopsy-labeled images for model validation. The CNN model in this system manages to benchmark its classiﬁcation performance against 21 certiﬁed dermatologist practitioners, in which the model successfully diagnoses and classiﬁes the ﬁve types of skin lesions on par with the dermatologist. An illustration in Figure 16 shows a CNN-based smart visual sensing system mapping the skin cancer from an image of the skin. The severity of the melanoma mapped by a color-coded distance map which highlights the severity based on features extracted from skin lesions.

CNN algorithms can be applied to mammograms with the potential to aid oncologists in early breast cancer detection that is not visible or present even for the trained eye. A smart sensor system developed by Ribli et al. demonstrated an intelligent model that screens and classiﬁes breast cancer lesions from X-ray mammogram images. The model uses the Faster R-CNN framework to train and classify malignant, benign, and normal tissues.

Figure 13. Smart sensor system for object detection and recognition. a) Fast R-CNN based smart sensor system demonstrating facial recognition and detection, in which the detected faces are shown in red boxes. The volunteers consented to their photos being taken and published as part of the figure. b) An example of smart sensor system based on the YOLO9000 framework which identiﬁes a range of objects and classiﬁes it in real-time. c) Block diagram of a smart sensor system that identiﬁes concrete cracks using CNN. Reproduced with permission. 

Reproduced with permission. Copyright 2017, Wiley-VCH.
in mammogram images. The model training process was expedited by pretraining with the ImageNet dataset\cite{ILSVRC15} before introducing the mammogram's image dataset. The breast cancer training images were obtained from three separate databases with annotated ground truths, in which 2620 mammogram images were obtained from Digital Database for Screening Mammography\cite{DDSM} and 847 Full-Field Digital Mammography (FFDM) from Department of Radiology at Semmelweis University in Hungary. The breast cancer classification architecture is shown in Figure 16b, where the left image represents the input mammograph image to the model, whereas the middle image identifies the potential region of cancerous tissues, and the right is the application output that locates the tumor. The model performance is benchmarked on two public challenges, achieving classification results of AUC = 0.85 on the Digital Mammography DREAM Challenge dataset which contains 86,000 images and an AUC score of 0.95 on the INbreast dataset which contains 15 FFDM images.\cite{INbreast}

Image segmentation is extensively studied and explored in the field of medical image analysis.\cite{ImageSegmentation} Cha et al.\cite{Cha2017} developed a CNN system that outlines bladder tumors from computerized tomography (CT) scans. The segmentation outlines in the model...
are used to accurately calculate the tumor size in response to pre- and post-treatment for bladder cancer prediction. The model was trained with 62 pretreatment CT scans of the bladder and tested on 62 post-treatment CT images. Segmentation results in Figure 16c show the tumor boundary after post-treatment. The tumor boundary is outlined from each model with CNN represented in light blue, AI-CAL in pink and radiologist in dark blue. The CNN model shows acceptable performance by highest prediction AUC score of 0.73 compared with AI-CAL and radiologist diagnosis.

Another area of interest is detecting and screening for bone fractures from medical images, where deep learning can greatly aid orthopedic doctors in identifying hairline fractures and bone deformity from these images. A hybrid NN system developed by Tomita et al.[23] trained a CNN/RNN model to identify osteoporotic vertebral fractures in CT images. The hybrid model is shown in Figure 17a-i first implements CNN-ResNet[238] framework to extract radiological features from 2D CT slices. The extracted feature vectors were then aggregated in the second part of the hybrid in which a LSTM NN classifies vertebral fractures. The model was trained with 1432 CT images containing 713 positive and 719 negative osteoporotic vertebral fractures that were split in the 80:10:10 ratio, 80% reserved for training, 10% for validation, and 10% for test data. The results from the CNN–LSTM hybrid model is shown in Figure 17a-ii and illustrates a color heatmap the identifying the probability of fractures in thoracic vertebrae by LSTM model. The hybrid classification model achieved the highest accuracy of 89.2% compared with three models and the conclusion drawn by a radiologist.

Van Grinsven et al.[132] explored examining retinal fundus images for the detection of diseases and anomalies in human eyes. The system implements the Fast-CNN model and integrates selective sampling, which improves the training efficiency of
the model. The model was trained with 6679 selected images (Kaggle database), which contain annotated labels of four levels of diabetic retinopathy severity. The training scheme was 60-20-20, in which 60% reserved for training, 20% reserved for validation, and reserved for 20% testing. Additional 1200 images from the Messidor database were exclusively used to test the performance of the model after training completion. The trained CNN model can classify hemorrhage regions in the retina by superimposing a color-coded probability heatmap over the original retina fundus image. Another application for fundus images involves segmenting intricate details of retina vessels in the image. Guo et al.²³⁶ reported this capability of the CNN-based smart sensor system, in which the segmentation results of the retina vessel are shown in Figure 17b.

Tumors in the human brain can be detected, segmented, and identified from MRI scans, in which Havaei et al.²³⁷ reported a brain tumor segmentation system that uses the CNN framework to focus on segmenting glioblastomas cancer in MRI images. The CNN model InputCascadeCNN was built upon a unique architecture that comprises two concatenated CNN models to greatly reduce the segmentation processing time. The model was trained with MRI scans to obtain from the 2013 BRATS dataset, which contains 30 MRI images with pixel-accurate ground truths that are separated in five classes (nontumor, necrosis, edema, nonenhancing tumor, and enhancing tumor). Further additional ten images from the dataset without ground truths were reserved for testing. The image classification results are shown in Figure 17c, comparing the brain tumor segmentation performance of ground truth and InputCascadeCNN. The final InputCascadeCNN CNN model segments brain tumors from MRI images 30 times faster compared with those of the MFCascadeCNN and the Tustison model in the study.

3.2.4. Biomedical Image Diagnosis and Analysis

Biomedical images are images taken from specialized equipment such as a transmission electron microscope (TEM) or a fluorescent microscope (FM). The captured images generated by the equipment are significantly magnified in which intricate details of the analyzed sample are shown. The smart sensor systems under this section use biomedical images in practical biomedical applications to identify cancerous cells and viruses for diagnosing diseases and illnesses.

Matuszewski and Sintorn²⁰⁰ showed a system enabling classification of viruses species from TEM images. The CNN model based on the U-Net framework focuses on improving training time performance by tuning four hyperparameters and decreasing the number of trainable weights used in U-Net. The model was trained with a TEM image dataset²¹⁹ consisting of 1077 images categorized in 15 virus classes. The dataset was split into a
70–30 ratio, where 70% of images are used for training and 30% reserved for testing. The classification results in Figure 18a superimpose a color heatmap to show the identification accuracy of the 15 virus classes. The finely tuned CNN smart model managed to achieve an acceptable classification accuracy of 82% utilizing 7.8 million training weights.

Biomedical smart sensor systems are not limited to CNN smart models that solve classification and segmentation problems; they can be applied to reconstruct images taken by a lower-resolution image sensor. Wu et al.\[201\] leveraged CNN to reconstruct localized microparticles from 20 mm² holographic images captured by a chip-based digital holographic microscope. Their system demonstrates real-time cluster detection and counting of herpes simplex virus (HSV) by monitoring the antibody microparticles attached to it. The microparticle reconstruction system architecture illustrated in Figure 18b shows a hologram image.
captured microparticles from the holographic image sensor and was reconstructed by a CNN in real time. The CNN model in the system was trained once with 12,960 annotated microparticle images and deployed directly to the application, in which the system is capable of analyzing HSV samples within 60 s and has a detection limit of 5 cpmL⁻¹.

Images from whole images slide for biopsy samples, stained with hematoxylin and eosin (H&E) can be used in a smart sensor.
system to detect prostate cancer. A system created by Campanella et al.\textsuperscript{[209]} trained two different types of NN for the classification of prostate cancer cells. The whole slide images captured from the microscope were processed by CNN based on ResNet34\textsuperscript{[238]} framework, where they are cropped into smaller tiles and used for training to identify cancerous tiles. The outcome of CNN had predicted tile scores that rank with the highest probability of cancer, where each tile score was fed into an RNN to classify the whole slide image as cancerous. The model used 44 732 images compiled from four datasets that contained prostate (in-house and external), skin, and axillary lymph nodes whole image slides. Each dataset’s image slide was randomly split into a 70–15–15 ratio, in which 70% of data was for training, 15% for validation, and 15% reserved for testing. The system was then evaluated against 44 372 images, and the model achieved an AUC score of 0.98 for all cancer types. Breast cancer can also be identifiable from stained H&E histology images with the implementation of a CNN model. A system created by Araujo et al.\textsuperscript{[240]} identified four types of tissue defects in H&E images that are associated with breast cancer (normal tissue, benign lesion, in situ carcinoma, and invasive carcinoma). The CNN model achieved a respectable 77.8% accuracy rate of identifying tissues across the four classes. A visualization of the hidden layers in the CNN smart model is shown in Figure 18c in which a sample training data of an H&E stained image is normalized in Figure 18c-i. The image was processed by the CNN model where the hidden activation of first (Figure 18c-ii) and second (Figure 18c-iii) layers in convolution were visualized. Finally, a t-distributed stochastic neighbor embedding (t-SNE) plot was implemented to visualize the CNN training patches and activation in a 2D Euclidean space. The first image showing the internal visualization of the CNN training patch is shown in Figure 18c-iv, which is followed by visualization of the last convolutional layer (Figure 18c-v) and finally the visualization of the second fully connected layer (Figure 18c-vi).

Other areas applications of microscope slide classification include the identification of the white blood cells. Kutlu et al.\textsuperscript{[241]} created a system classifying different types of white blood cell types (lymphocytes, monocytes, eosinophils, basophils, and neutrophils). The model based on the Regional-CNN ResNet50 framework used 5000 slide images compiled from two datasets—Blood Cell Images (BCCD) and Leukocyte (LISC).\textsuperscript{[242,243]} The datasets were split into 80–20 ratio, in which 80% of data for training and 20% for testing and benchmarking against multiple R-CNN frameworks. The white blood cell classification results shown in Figure 19a compare 4 R-CNN frameworks, which are AlexNet, VGG16, GoogLeNet, and ResNet50. The smart model based on ResNet50 framework achieved the highest accuracy in identifying all white blood cell types, and the classification accuracy of lymphocyte cells reached up to 99.52%, 98.40% for monocyte, 98.48% for basophil, 96.16% for eosinophil, and 95.04% for neutrophil.

Segmentation application provides accurate detailing of the physical outline of the desired target, for instance, images

| Full Training CNN Architectures |
|------------------|------------------|------------------|------------------|------------------|
| Images            | AlexNet          | VGG 16           | GoogLeNet        | Resnet50         |
|                   | ![Image](image1) | ![Image](image2) | ![Image](image3) | ![Image](image4) |

**Figure 19.** Smart applications for medical image diagnosis and analysis. a) Classification results of 4 CNN frameworks identify five types of white blood cells from the microscope slide image. The system is capable of distinguishing five types of white blood cells: lymphocytes, monocytes, eosinophils, basophils, and neutrophils cells. Adapted with permission.\textsuperscript{[241]} Copyright 2020, Elsevier Ltd. b) A smart sensor system uses images of mesenchymal stem cells to count individual amounts of cells in a stem cell cluster. The system based on a CNN smart model. Adapted with permission.\textsuperscript{[244]} Copyright 2019, Elsevier Ltd.
obtained from a focused-ion beam electron microscopy were used to develop an automated system that is capable of segmenting mitochondria cells from other cells.[245] Furthermore, CNN can also be used to quantify objects in images, as demonstrated in a system created by Hassanlou et al.[244] The system counts liquid droplets in the mesenchymal stem cells cultures obtained from optical microscope images. The CNN model was trained with 200 images, which were then cropped into smaller images with single-pixel annotations. The classification and cells count results are shown in Figure 19b. The CNN model achieved an average counting accuracy of 94%, which outperformed to state-of-the-art cell counting methods: ImageJ and Cohen’s method.

4. Conclusions

ML algorithms are emerging as crucial elements in modern sensing systems for the burgeoning demand for high-throughput data analysis, which is evident in the reported systems in this Review. The ML algorithms and sensors technologies are key components of a ML-enabled smart sensor system, in which the most suitable ML is implemented in the smart model based on the sensor technology and specific application requirements. The smart models in these systems efficiently process and analyze the tremendous amount of data generated from the sensors, in which crucial sensory information, such as clusters of patterns or predictive trends is extracted for practical applications. The ML algorithms in these systems are divided into classic non-NN and NN algorithms and are discussed in the perspective of a sensor device.

Physical and chemical and image sensors are the two key groups of sensor technology in ML-enabled smart sensors system; each group of sensors fundamentally has different sensing principles. The novelty of physical and chemical sensors in ML-enabled smart sensor systems are discussed, and target application areas such as early disease prevention, real-time medical diagnosis, smart farming, and environmental pollution prevention are listed in this article. Subsequently, image-based smart sensor systems are reviewed, and the novelty of the application is reported under smart visual imaging sensing systems. The section summarizes application in real-time objection, facial recognition, remote precision farming, medical disease diagnosis, and biomedical disease and virus analysis.

Although, the most suitable ML algorithm for a smart model is based on the sensor technology and the smart application requirement. Conversely, this is not entirely the case; there is a strong correlation between the image sensor and a CNN-based smart model developed for classification application, it is evident in the reported cases in which a significant number of smart sensor systems perform exceptionally well under this configuration. On the contrary, physical and chemical sensing in smart sensor systems shares fewer similarities of ML choice between each system. However, from the review systems under this section, there is an indicative trend that non-NN smart models are being replaced with an NN-based model for similar types of application.

The future of ML-enabled smart sensor system is the effectiveness of processing and analyzing high-throughput sensory data. These types of datasets continue to grow exponentially and store a tremendous amount of information and data, which are no longer feasible to process and analyze with traditional methods.[246] Emerging smart applications in the future is anticipated to integrate cloud-based solutions, in which the smart system takes full advantage of virtually infinite data storage and almost limitlessly scalable parallel computational power.[247]

One of the many challenges a smart sensor system faces is the ability of the sensor system to adaptively learn from its environment. Once a trained smart model is deployed in the real world, it is referred to as an inference model, in which the trained model stays fixed and cannot be further optimized. As a consequence, any sensor measurements outside the scope of the trained inference model will result in catastrophic errors, false outcomes, and unpredictable smart applications. Moreover, it is practically impossible to capture and consider all input sensor measurement variables and scenarios to develop a smart model in one instance.

Currently, there are software methods that mitigate these types of risks, for example, to combat unknown driving scenarios in self-driving vehicles manufactured by Tesla, Inc. The company periodically provides over-the-air firmware updates of newer and improved inference models that have been trained with newly acquired sensor data obtained from consumers who wish to participate in their improvement program (cloud server). Furthermore, recent advancements in neuromorphic computing enabled by emerging material science have shown a potential solution for “adaptive sensor learning.” The electrical synapses generated from dedicated hardware such as memristors[248] can be coupled with the time-dependent neurons nodes in a spiking neural network (SNN), which mimic the adaptive learning process found in the brain.[249]

Smart sensors systems in their current state have demonstrated the potential of completely replacing conventional sensing systems of analyzing and solving problems, and perhaps one day smart sensor systems could adaptively learn outside its inference model and intelligently self-integrate the features into its system. The natural forward progression of technology will eventually see techniques improves, better frameworks, and future breakthroughs in Artificial Intelligence[250] in which classic non-NN algorithms in smart models will be eventually being replaced by their NN counterparts. As modern society ramps up for the inevitable transitions into industrial revolution 4.0,[247] there will an ever-increasing demand for highly efficient, intelligent, accurate, and autonomous smart sensor systems that processes real-time risk and makes a split-second decision which may have life and death consequences. This is already evident in the context of autonomous vehicles and can extend to manufacturing industries where conventional systems are now being replaced by NN smart machines and robots that are connected to the cloud.[251]

Acknowledgements

N.H. would like to acknowledge and is grateful for the financial support from The Australian Government Research Training Program (RTP) and RMIT University School of Engineering (SoE). J.O. and A.M. would like to acknowledge the financial support from the RMIT University Enable Capability Platform (ECP) Capability Development Fund (CDF) Scheme (CDF19092).
