Digital Communications and Networks 7 (2021) 526–542

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

Deep learning for the internet of things: Potential benefits and use-cases

Tausifa Jan Saleem *, Mohammad Ahsan Chishti

Department of Computer Science and Engineering, National Institute of Technology, Srinagar, J&K, India

ARTICLE INFO

Keywords:
Internet of things (IoT)
Deep learning
Convolutional neural network
Recurrent neural network
Long short term memory

ABSTRACT

The massive number of sensors deployed in the Internet of Things (IoT) produce gigantic amounts of data for facilitating a wide range of applications. Deep Learning (DL) would undoubtedly play a role in generating valuable inferences from this massive volume of data and hence will assist in creating smarter IoT. In this regard, exploring the potential of DL for IoT data analytics becomes highly crucial. This paper begins with a concise discussion on the Deep Neural Network (DNN) and its different architectures. The potential benefits that DL will bring to the IoT are also discussed. Then, a detailed review of DL-driven IoT use-cases is presented. Moreover, this paper formulates a DL-based model for Human Activity Recognition (HAR). It carries out a performance comparison of the proposed model with other machine learning techniques to delineate the superiority of the DL model over other techniques. Apart from enlightening the potential of DL in IoT applications, this paper will serve as an impetus to encourage advanced research in the realm of DL-driven IoT applications.

1. Introduction

The modern advancements in technology and the rapid merger of areas, such as sensing and actuating technologies, embedded systems, wireless communication, and data analytics, are speeding up the development of the Internet of Things (IoT) [1]. The IoT is used for garnering huge business profits [2,3]. The impulsive expansion in the number of devices linked to the IoT and the exponential surge in data utilization demonstrates how the growth of big data and IoT coincide with each other. The amalgamation of these fields is touching almost all areas of technologies and businesses [4,5].

Data analytics means processing every segment of data to discover the patterns in the data, dig out the hidden knowledge, and extract precious information. The key concern of IoT data analytic systems is to attain enhanced elucidation of data so that efficient control decisions can be devised. Those decisions can be utilized to manage and facilitate diverse applications. A major component of the IoT system is a proficient data analytical mechanism that can accomplish tasks like classification, prediction, clustering, regression, association rule mining, etc. Deep Learning (DL) has been used extensively for analyzing the data generated by IoT systems. Both IoT and DL have been stated to be amongst the three vital prudent technological trends for the year 2017 [6].

The rationale behind this meticulous hype of DL is the incompetence of traditional machine learning techniques to meet the mounting analytical requirements of IoT systems. DL models are superior to traditional machine learning techniques in a number of ways: first, they ease the prerequisite that the labeled data should be utilized for training. Therefore, features that might not be identifiable to a human can be mined efficiently by DL approaches. Moreover, these approaches outperform traditional techniques in terms of accuracy. Furthermore, DL architectures are suitable for modelling complex behaviors of multimodal datasets [7]. Table 1 presents a comparison of DL and the conventional machine learning.

The rationale for this paper stems from the fact that DL architectures have shown phenomenal performance in diverse domains like image analysis, speech analysis, recommendation systems, medical data analysis, etc. Hence, exploring the benefits of DL in IoT applications is crucial. For this purpose, this paper explores the benefits of DL models in IoT scenarios and presents a detailed review of research works devoted to DL-driven IoT applications. The paper also formulates a DL-based model for Human Activity Recognition (HAR) and carries out a performance comparison of the proposed model with other machine learning techniques in order to delineate the superiority of DL over other techniques. Moreover, challenges and future aspects are also presented to foster advanced research in the domain of DL-driven IoT data analytics.

The paper is structured as follows: Section 2 contains the description of Deep Neural Network (DNN) and its different architectures. Section 3 presents the benefits of DL in IoT applications. Section 4 presents the review of DL-driven IoT use-cases. Section 5 proposes a model based on DL for HAR. The performance comparison of the proposed model with

* Corresponding author.
E-mail address: tausifa_30phd17@nitsri.net (T.J. Saleem).

https://doi.org/10.1016/j.dcan.2020.12.002
Received 15 June 2020; Received in revised form 31 October 2020; Accepted 1 December 2020
Available online 8 December 2020

Table 1
DL versus conventional machine learning.

Parameter	Deep learning techniques	Conventional machine learning techniques
Data requirement	Require massive data	Require lesser data
Accuracy	High	Lesser
Training time	Long	Relatively lesser
Hardware requirement	Computationally rich devices (GPU)	CPU
Hyper-parameter tuning	Multiple ways	Limited capability
Data pre-processing	Minimal requirement	Highly required
Network design	Complex	Usually simple
Feature engineering	Minimal requirement	Highly required
Ability to model complex problems	Strong	Weak

other machine learning techniques is also provided. Section 6 discusses the various challenges and section 7 presents the future research directions. Section 8 presents the concluding remarks.

2. Deep Neural Network (DNN)

The DNN consists of multiple processing layers that are capable of extracting hierarchical characteristics from the input data [8]. The functioning of DNN is emulated from the working of the human brain. Each layer in the DNN consists of multiple processing units called neurons. A neuron performs the weighted sum of the inputs $[x_1, x_2, x_n]$ and feeds the resulting sum to an activation function that generates the desired output. Each neuron consists of a set of weights $[w_1, w_2, w_n]$ and a bias (b) that is optimized during the training process. The functioning of the artificial neuron is depicted in Fig. 1. Table 2 presents the list of preliminaries.

DL encompasses diverse supervised and unsupervised architectures, including Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Auto-Encoder (AE), Generative Adversarial Network (GAN), Restricted Boltzmann Machine (RBM), and Deep Belief Network (DBN). The summary of these architectures is presented in Table 3. The following section provides a brief description of these architectures.

2.1. Convolutional Neural Network (CNN)

The input to a CNN is a two-dimensional image or a voice signal. The CNN digs out the hierarchical characteristics from the input data with the help of a chain of hidden layers, which comprises convolutional layers and pooling layers [9]. The convolutional layers consist of kernels having the structure of the input data. It carries out the product of input and the kernel and produces outputs called filter maps. The pooling layers reduce the dimensions of these feature maps to slash the processing time and lower down the chances of over-fitting. The output from the final pooling layer is passed through the fully-connected layer to generate the desired output. The CNN has shown promising results in image recognition tasks.

2.2. Recurrent Neural Network

For simulating time-series problems, an extension of a feed-forward neural network called RNN was designed as the conventional feed-forward neural network did not have the capability of modelling such tasks. The input to an RNN is output at time 't-1' and input at time 't' [10]. The neurons in the RNN contain an internal memory for tracking previous calculations. Instead of backpropagation, its variant, known as Back Propagation Through Time (BPTT), is utilized for training the RNN. RNNs cannot be used for modelling time-series data having long-term dependencies, because they suffer from the vanishing gradient problem in those scenarios. To avoid such a phenomenon, an extension of the RNN called Long Short Term Memory (LSTM) was introduced [10]. Every processing unit in an LSTM is equipped with memory and gates. The memory cell stores the values for an arbitrary time, and the gates make a decision about what information is to be thrown away from the memory cells, and what information is to be retained. This results in a network with a controlled flow of data and hence can be used for modelling time series tasks with long time lags. RNNs and LSTMs have been employed in various IoT use-cases with time-series data.

2.3. Auto-Encoder (AE)

The AE falls under the generative category of DNN models. They comprise of one or more hidden layers in addition to the input and output layers [11]. The output layer consists of the same number of neurons as that of the input layer. The goal of AEs is to rebuild the input in order to learn its condensed form. They consist of two main parts: one is an encoder that transforms the input data into a new delineation called a code, and the other is a decoder that regenerates the input back from the code. The major goal of training in AEs is to reduce the error between the input data and the generated output. AEs are utilized for feature extraction and dimensionality-reduction tasks.

2.4. Generative Adversarial Network (GAN)

GANs are the hybrid DNN models that consist of two neural networks: generator and discriminator [12]. These networks function jointly to produce high-quality data. The generative network is trained to capture the data distribution and produce new data based on the patterns learned from the data, whereas the discriminative network is trained to maximize the error rate between the actual data and the data generated by the generative network. GANs are well suited for scenarios with noisy data.

2.5. Restricted Boltzmann Machine (RBM)

The RBM is made of two layers: the visible layer and the hidden layer [13]. The input is fed to the visible layer, and the hidden layer learns the

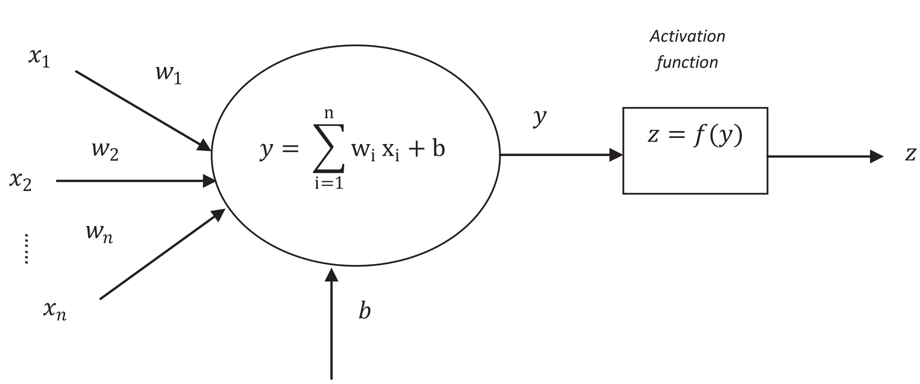

Fig. 1. Artificial neuron.

Table 2
List of preliminaries.

Concept	Description
Training	It is an iterative procedure of learning optimal parameters in the network
Testing	It is a procedure to assess the performance of the model
Dataset	Collection of information consisting of rows and columns where every row represents the data pertaining to an entity and column represents a feature vector
Hyper-parameters	Parameters that are not learned by the model but require manual setting for better model performance
Activation Function	Defines the output of the neuron. They have different types: linear, rectified linear unit, hyperbolic tangent, sigmoid, softmax, etc.
Back-propagation	Algorithm for computing the gradients
Optimizer	Algorithm for fine-tuning the neural network parameters based on the gradients computed using backpropagation method. Examples of optimizers are; Stochastic Gradient Descent, Adam, RMSprop, Adadelta, Adagrad, etc.
Loss-function	Crucial for neural network model optimization. Keeps track of the difference between target output and actual output. Examples include: Mean Squared Error, Hinge loss, Cross-entropy loss, etc.
Epoch	One cycle through the entire training dataset
Iteration	Number of steps required to complete one epoch
Batch-size	Number of training examples processed in one iteration
Learning Rate	Manages the rate at which the model is adjusted to the problem
Over-fitting	Case when the model is not able to generalize the data and consequently produces poor test results
Regularization	Modifications incorporated in the model that result in the reduction of the generalization error. It is performed using techniques like Dropout, L1-regularization, L2-regularization, etc.
Supervised learning	Learning procedure that utilizes labeled data to map input to output
Unsupervised learning	Learning procedure that makes use of unlabeled data to extract previously unknown patterns from the data
Semi-supervised learning	Combined approach that makes use of limited labeled data and a large amount of unlabeled data for model learning
Generative model	Models the class distribution
Discriminative model	Models the decision boundary
Accuracy	$Accuracy = (TP+TN)/(TP+FP+TN+FN)$ where TP represents the number of True Positive instances, TN represents the number of True Negative instances, FP represents the number of False Positive instances, and FN represents the number of False Negative instances
Precision	$Precision = TP/(TP + FP)$
Recall	$Recall = TP/(TP + FN)$
Mean Squared Error (MSE)	$MSE = (\sum_{i=0}^{N} (Z_i^{'} - Z_i)^2)/N$ where Z_i is the actual value, $Z_i^{'}$ is the forecasted value, and N is the total number of data instances
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{(\sum_{i=0}^{N} (Z_i^{'} - Z_i)^2)/N}$
Mean Absolute Error (MAE)	$MAE = 1/N(\sum_{i=0}^{N} \lvert Z_i - Z_i^{'} \rvert)$
Mean Absolute Percentage Error (MAPE)	$MAPE = 100\%/N(\sum_{i=0}^{N} \lvert (Z_i - Z_i^{'})/Z_i \rvert)$
Area Under Curve (AUC)	Area under Receiver Operating Characteristics (ROC) curve
F-measure	$F - measure = 2*(Precision*Recall)/(Precision + Recall)$

probability distribution from the input data. The neurons in the visible and hidden layers are connected in a way that they form a bipartite graph. Back-propagation and gradient descent techniques are utilized during training to determine the optimal parameters in the network. The purpose of training in the RBM is to augment the product value of the probabilities of the units present in the visible layer. The RBM can perform tasks like classification, feature extraction, and dimensionality reduction.

2.6. Deep Belief Network (DBN)

DBNs are a generative type of DNN. They consist of a visible layer and

Table 3
Characteristics of different DNN models.

DNN	Category	Learning Type	Characteristics	Major Applications
CNN	Discriminative	Supervised	- Does not require hand-crafted features - Show property of translational invariance and local connectivity - Possess Shared weights - Require large training datasets - Prone to model over-fitting	- Image analysis - Image recognition - Video analysis - Gaming - Medical data analysis - Drug discovery - Recommendation systems
RNN	Discriminative	Supervised	- Suitable for time-series modelling tasks - Consists of memory units for keeping track of previous inputs - Difficult training procedure	- Speech recognition - Image classification - Image captioning - Natural language processing - Language translation
AE	Generative	Unsupervised	- Consists of the same number of units in the input and output layer - Performs automatic feature extraction - Reduces the dimensionality of the datasets - Can deal with noisy data	- Image compression - Image de-noising - Image generation - Recommendation systems
RBM	Generative	Unsupervised, Supervised	- Performs tasks like classification, feature extraction and dimensionality reduction - Extracts complex features from the data - Complex training procedure - Can be trained in a supervised manner as well to perform classification tasks	- Collaborative filtering - Intrusion detection - Topic modelling - Medical data analysis
DBN	Generative	Unsupervised, Supervised	- Consists of stacked RBMs - Is trained in a greedy fashion - Extracts high-level features from the data - Usually trained in an unsupervised manner but can be trained in a supervised way as well to	- Image generation - Image recognition - Drug discovery - Medical image analysis - Text classification - Natural language processing - Speech recognition

(*continued on next page*)

Table 3 (*continued*)

DNN	Category	Learning Type	Characteristics	Major Applications
			perform classification tasks	- Intrusion detection
GAN	Hybrid	Semi-supervised	- Consists of two parts: generator and discriminator - Mostly used with noisy data - Can generate new data based on the correlations in the original data	- Image generation - Image quality improvement - Image de-noising - Speech synthesis - Speech enhancement - Medical data analysis

multiple hidden layers [14]. They are competent in mining high-level abstractions from the input data. A DBN consists of multiple stacked RBMs. The training of DBNs is carried out in an unsupervised way where the model learns to rebuild the inputs probabilistically from the features extracted at each layer. DBNs can also be used for discriminative prediction tasks apart from generative feature discovery.

3. Potential benefits of DL for IoT

This section describes the characteristics of IoT data along with the potential of DL models in handling those characteristics. The following is a point-to-point description:

- *Time series data:* unlike conventional data, the data from the IoT has a timestamp for all data points [15–17]. And the conventional data analytical paradigms are not suitable for analyzing such a kind of data.

Solution: DL models are very good at analyzing time-series data. The RNN and the LSTM can process time-series data very well. These two architectures have been specifically designed for this purpose.

- *High noise:* IoT data is affected by noise in the process of data acquisition and data transmission as IoT data is generated by highly constrained devices. Experiments demonstrate that Radio Frequency Identification (RFID) devices only generate 60%–70% of the accurate data [18,19].

Solution: DL models have the ability to tolerate noise. The AEs and GANs can reconstruct the original data from the corrupted data.

- *Weak semantics:* data produced by IoT devices has weak semantics. Hence, intricate semantics ought to be extracted from the collection of weak semantic data so as to facilitate diverse IoT applications [18,19].

Solution: DL models have the capability to dig out high-level characteristics from the weak-semantics data.

- *Massive real-time data:* a large number of sensors deployed in highly dynamic IoT environments continuously generate huge real-time streaming data that need to be analyzed within a specified time-lapse [18,19]. For example, an aeroplane with an average 12-h flight-time generates up to 844 terabytes of real-time streaming data.

Solution: DL, along with the recent computing paradigms, including fog computing and edge computing, has made real-time data analytics possible in time-stringent applications. Training DL models on computationally capable devices and then deploying the model either on fog or edge devices to generate fast inferences, making accurate real-time data analysis possible in such applications.

- *Heterogeneity:* data generated from heterogeneous and distributed sensing systems in the IoT are highly diverse and may be of structured, unstructured or semi-structured nature [16–19]. And analyzing such diverse data with different formats is a tedious job that goes beyond the capability of conventional data analytical approaches.

Solution: DL models have the capability of working on any data, be it structured, unstructured or semi-structured. Moreover, they have shown remarkable results in multi-modal data analysis. The impulse is that DL models can recognize imperceptible features from cross-sensor correlations that result in highly accurate inferences.

- *Large-scale data:* the IoT data is characterized by a huge amount of data, as numerous sensors deployed in IoT environments continuously emanate a large amount of data, which demands to be processed and analyzed within a certain lapse of time [16,17].

Solution: considering that DL models have shown impressive results in IoT applications, a large amount of data should not endanger its use in these applications. Implementing DL models on a large number of computing devices in a distributed manner is the potential solution to this challenge.

4. DL in IoT applications

This section discusses the IoT applications and the role of DL architectures in those applications, including smart homes, smart healthcare, smart grid, smart manufacturing, smart agriculture, and smart transportation. Fig. 2 depicts the applications of DL in IoT systems. The following subsections provide a detailed description of the role of DL in IoT applications.

4.1. Smart home

A smart home is an imperative application of the IoT in which the dwellings are examined by ambient intelligence to offer context-aware services and facilitate remote monitoring and control of home appliances [20]. The following research works prove the fact that DL has been utilized in numerous smart home use-cases for facilitating services like energy demand prediction, behavior monitoring, human activity recognition, posture detection, etc.

4.1.1. Energy demand prediction

Energy demand prediction in a smart home is regarded as a crucial mechanism for managing power consumption. In recent years, due to the superfluous consumption of residential electricity, the requirements of energy prediction and optimization have achieved an urgent significance. DL has become a promising solution for energy demand prediction. Heng et al. [21] predicted the load in a household using the RNN. The authors proposed a novel RNN based on pooling to forecast the load. The pooling strategy was utilized to overcome the over-fitting problem faced by a simple RNN. Performance comparison of the proposed technique was carried out with techniques like Auto-Regressive Integrated Moving Average (ARIMA), Support Vector Regression (SVR), and RNN, and it was concluded that the proposed technique outperforms ARIMA by 19.5%, SVR by 13.1% and simple RNN by 6.5% in terms of Root Mean Squared Error (RMSE). In another study, Shirajum et al. [22] designed an intelligent predictor model based on the RNN to forecast energy demand in smart homes. They compared the performance of RNN with that of ARIMA and demonstrated that the RNN performs better than the ARIMA.

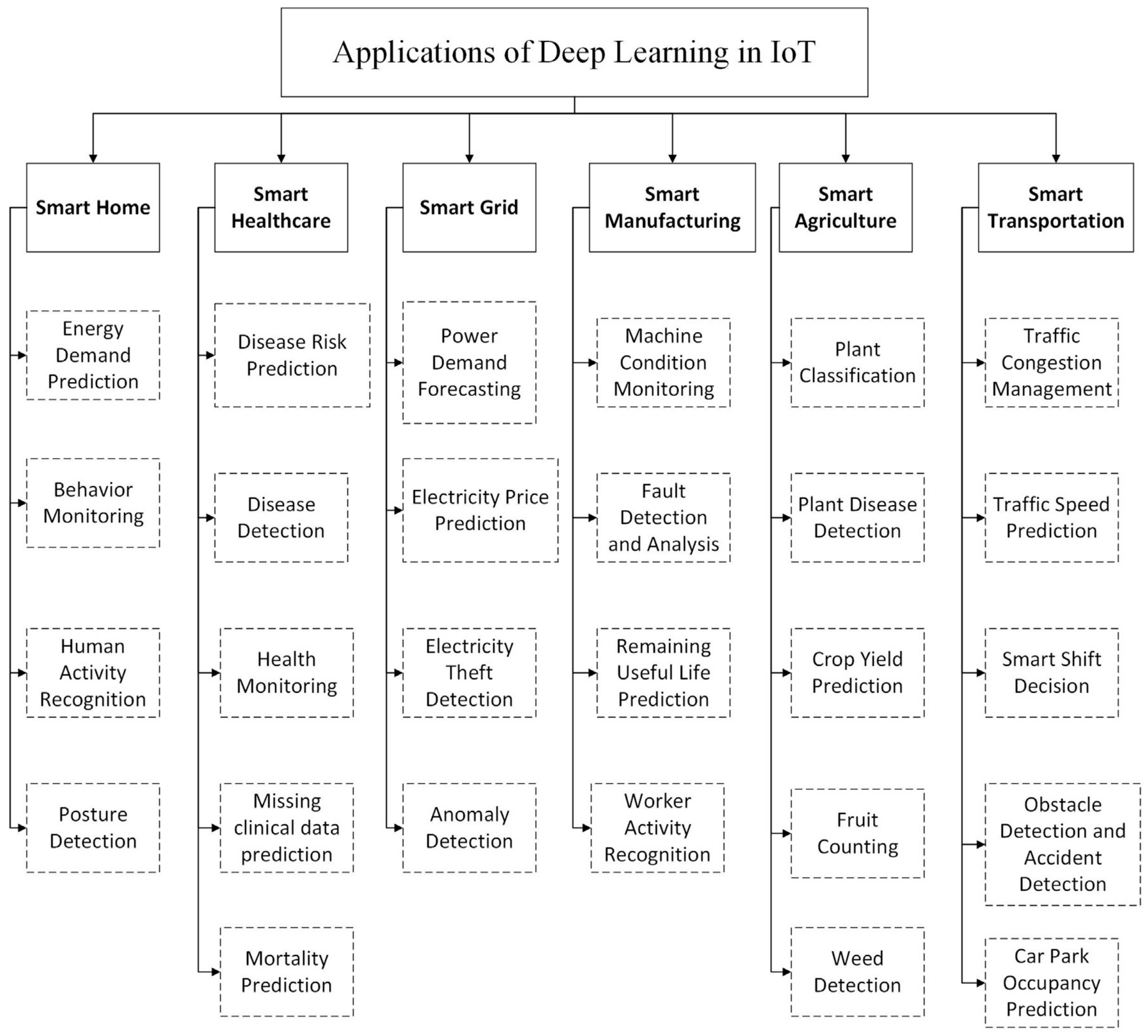

Fig. 2. Applications of DL in IoT.

In Ref. [23], the authors framed a method based on the LSTM to model data management and forecast the power in home solar systems. They demonstrated that the LSTM has the capability to dig out intricate features from the high-dimensional data and gives better results than the SVM. Felan et al. [24] utilized Stacked Denoising AE (SDAE) to present device level feedback to inhabitants in a smart home. The model achieved an accuracy of 90%.

4.1.2. Behavior monitoring

In smart home environments, the IoT data can be analyzed to understand the behavior of individuals. Due to the remarkable success in image recognition, DL has been employed for behavior monitoring. Yin et al. [25] formulated a model based on the amalgam of CNN and RNN to recognize the emotions of subjects from a video. The experiments conducted in the study demonstrate that the proposed model shows impressive results. In another study, Eiman et al. [26] performed emotion detection using the amalgam of CNN and LSTM. The authors collected data pertaining to the human body, location and environment modalities using smartphones and wearable devices. They compared the performance of the hybrid CNN-LSTM model with the Multi-Layer Perceptron (MLP) and CNN and demonstrated that the hybrid model performs better than the CNN and the MLP by 16% and 21%, respectively. The study concludes that DL models are quite effective in classifying human emotions and have the potential of understanding the mental well-being of humans.

4.1.3. Human activity recognition (HAR)

Due to the brisk growth of sensors technology and ubiquitous computing technology, HAR based on sensory data has become more popular. Aiguo et al. [27] formulated a Stacked Auto-encoder (SAE) based model to recognize the activities of individuals. They conducted experiments on three publicly available datasets that contain sensory data collected from three smart homes. The proposed model was compared with Naïve Baye's (NB), Hidden Markov Model (HMM), K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) in proficiency. And it was demonstrated that SAE performed better than all other shallow architectures. The authors concluded that the proposed model could be further optimized by changing the hyper-parameters like the number of layers and the number of hidden units. In another study, Park et al. [28] performed HAR using the RNN. The authors conducted experiments on the MSRC-12 gesture dataset in which human activities are represented as temporal changes in different joint angles. The study was able to detect 12 human activities with an accuracy of 99.55%. Moreover, the study compared the performance of RNN with those of HMM and DBN and demonstrated that RNN outperforms HMM by 7.06% and the DBN by 2.01%. In Ref. [29], Xile et al. performed HAR based on SDAE and LightGBM (LGB) on four different datasets. They utilized SDAE to remove the noise from the raw sensory data and extract the most crucial features in an unsupervised manner. The LGB was used to extract the dependencies between features for accurate classification. The study attained good accuracy on all the datasets. Francisco et al. [30] proposed an amalgam of CNN and LSTM (DeepConvLSTM) for multimodal HAR.

The proposed model is capable of mining high-level features from the data and has the ability to model temporal dependencies in the data. DeepConvLSTM consists of eight layers: an input layer, four convolutional layers, two dense layers and a softmax layer. The data captured by wearable devices is fed to the input layer and is processed by four convolutional layers, then the output from convolutional layers is fed to dense layers that perform a non-linear transformation on the data and finally the softmax logistic regression is performed in the output layer. The authors evaluated the performance of the proposed model on two datasets: the opportunity dataset and the Skoda dataset. It was demonstrated that the proposed model outperforms the state-of-the-art techniques by 4% in the case of the opportunity dataset and by 6% in the case of the Skoda dataset. The study concludes that the proposed model has the ability to fuse data from different modalities. In Ref. [31], the authors assessed the proficiency of several approaches for predicting the activities of humans. The experiments conducted in the study demonstrate that the LSTM outperforms other prediction techniques. Han et al. [32] proposed a novel framework named AE-LRCN based on the combination of AE, CNN, and LSTM for HAR. The purpose of the AE is to remove the noise from the input data, the CNN digs out high-level characteristics from the data, and the LSTM models the temporal dependencies in the data. The study demonstrated that the proposed model is able to classify a number of daily human activities with an accuracy of 97.4%.

4.1.4. Posture detection

Another important application of DL in smart home environments is body posture detection. Soham et al. [33] formulated a model based on RBM to detect body postures from the data captured by wearables. The proposed model was evaluated on the Weizmann human silhouette based action database. The dataset consists of videos from nine different subjects signifying ten different actions. The proposed model detected body postures with an accuracy of 80%.

Table 4 provides an outline of the research works discussed in this sub-section.

4.2. Smart healthcare

The IoT and DL have been extensively recognized as impending solutions to lighten the stress on healthcare systems, and thus have become the focus of recent research in such systems. DL has been employed in numerous healthcare use-cases, including disease risk prediction, disease detection, health monitoring, missing clinical data prediction, mortality prediction, etc.

4.2.1. Disease risk prediction

DL has been found to be quite effective in predicting the risk of future diseases. Min et al. [34] devised a CNN-based model to estimate a patient’s risk of cerebral infarction. They combined features from structured data as well as unstructured data to forecast the risk of the disease. The study carried out a performance comparison of CNN-based Unimodal Disease Risk Prediction (CNN-UDRP) and CNN-based Multimodal Disease Risk Prediction (CNN-MDRP) and concluded that CNN-MDRP converges faster than CNN-UDRP. The proposed model attained an accuracy of 94.8%. In another study, Xing et al. [35] formulated a framework based on the RBM to estimate the multiple types of associations between diseases and micro-RNA. The proposed framework consists of three steps: building RBMs from a disease-microRNA interaction network, training RBM by contrastive divergence, performing predictions by calculating conditional probabilities. The authors demonstrated that the proposed model is reliable and yields accurate results. In another study, Min et al.

Table 4
DL in smart home use-cases.

Work	IoT use-case	Dataset	DNN Used	Purpose of DNN	Performance Results
[21]	Household load prediction	Data obtained from Smart Metering Electricity Customer Behavior Trials provided by the Commission for Energy Regulation Ireland	RNN	To predict the load in a household	RMSE: 0.4505
[22]	Energy demand forecasting	Sensory data	RNN	To forecast the energy demand	–
[23]	Solar power system forecasting	National Solar Radiation Database (NSRDB)	LSTM	To model the data management in home solar power systems	MAE: 2.566
[24]	Energy disaggregation	Sensory data	AE	To present device level feedback to home inhabitants	Accuracy: 90%
[25]	Emotion recognition based on video	AFEW 6.0 dataset, HAPPEI dataset	CNN + RNN	To model the movements in a video	Accuracy: 59.02%
[26]	Emotion Detection	Data gathered from wearables and Smartphone	CNN + LSTM	To identify the emotions based on multiple modalities like physiological, location, and environmental	Accuracy: 95%
[27]	Human Activity Recognition	Data gathered from sensors deployed in smart homes	AE	To capture the most effective features and construct a model for human activity recognition	Accuracy: 85.32%
[28]	Human Activity Recognition	MSRC-12 dataset	RNN	To classify the input data into 12 human activities	Accuracy: 99.55%
[29]	Human activity recognition	the Human Moving Modes (HMM), the Human Static Behavior Dataset (HSBD), the Human Dynamic Behavior Dataset (HDBD)	AE	To learn the most crucial features required for human activity recognition	Accuracy (HMM): 95.73% Accuracy (HSBD): 98.22% Accuracy (HDBD): 96.31%
[30]	Multimodal Activity Recognition	Opportunity dataset, Skoda dataset	CNN + LSTM	To mine the spatial and temporal variability of the input	F1 score (Opportunity): 86.4% F1 score (Skoda): 95.8%
[31]	Human Activity Prediction	MIT B, CASAS hh104, van Kasteren	LSTM	To forecast the timestamp of human activity apart from predicting the next event	MAE in seconds (MIT B): 660.54 MAE in seconds (CASAS hh104): 15.84 MAE in seconds (van Kasteren): 1130.59
[32]	Device-free human activity recognition	CSI data gathered by IoT devices	AE + CNN + LSTM	To capture high-level features	Accuracy: 97.4%
[33]	Posture detection	Videos of 9 subjects signifying 10 actions	RBM	To detect the body posture from the data captured by wearables	Accuracy: 80%

[36] devised a 5G-smart diabetes system based on the CNN to predict an individual's risk of disease. The experiments demonstrated that the designed system has the capability of diagnosing whether a patient has diabetes or not and provides treatment recommendations to the patient. Peng et al. [37] reported the utilization of DBN to design a model for predicting cardiovascular diseases. They utilized the functionalities of unsupervised training and supervised optimization to develop an improved DBN that automatically determines the optimal network structure based on the reconstruction error. The authors performed experiments on two datasets: the Statlog (heart) dataset and UCIs heart disease dataset. The optimal network depth for the Statlog (heart) dataset was found to be 4 while the optimal depth for UCIs heart disease dataset was 5. The results conducted in the study demonstrate the model's high accuracy on both datasets. Jaekwon et al. [38] utilized the DBN to construct a cardiovascular risk prediction model. The DBN performed the risk prediction based on six input features: age, systolic blood pressure, diastolic blood pressure, high-density lipoprotein, blood sugar, and smoking habits. The model was trained for 200 epochs with a batch size of 12. The study demonstrated that the model performs better than several benchmarks. The authors in Ref. [39] utilized the LSTM to model the risk prediction of heart failure based on the patient's electronic medical information. Silu, tanh and softmax were used as activation functions in the model, Bridgeout was used as a regularization technique to avoid model over-fitting, and the model was trained for 100 epochs. The performance of the model was compared with MLP, Logistic Regression (LR), KNN and SVM, and it was demonstrated that the LSTM outperforms all other techniques.

4.2.2. Disease detection

DL techniques have the potential of identifying the health complications a patient suffers from. The authors in Ref. [40] utilized the CNN to design a voice-pathology system. The devised system classifies voice signals into normal and pathological categories. Two pre-trained CNN architectures, VGG-16 and CaffeNet, were utilized in the study. The input to the CNN models is the voice spectrogram and its first and second-order derivative with respect to time. The output layer in both CNN models is replaced by an output layer consisting of two neurons. Experiments carried out in the study reveal that the CaffeNet architecture outperforms VGG-16 and other baseline techniques. Ludi et al. [41] formulated an LSTM-based inception model to detect congestive heart failure. The model was trained for 100 epochs with a batch size of 128. Adam was used as an optimizer and the binary entropy as the loss function. The study demonstrated that the proposed model outperforms several existing studies. The authors suggest that the proposed model can be deployed in ECG devices as a diagnostic tool to examine the condition of the human heart.

4.2.3. Health monitoring

Health monitoring is crucial for examining and recuperating personal health. This category of applications utilizes DL for providing crucial suggestions to individuals with health ailments and for notifying healthcare professionals in the case of an emergency. Syed et al. [42] formulated a healthcare framework consisting of the IoT and the cloud for monitoring patients in real- time. The framework was evaluated with the help of EEG classification use-case. EEG signals obtained from the patient are transferred to the cloud where a DL module consisting of CNN classifies the signals into two categories: pathological and normal. Two pre-trained CNN architectures, VGG16 and Alexnet, were used in the study. The final classification layer in both architectures is replaced by the SVM classifier. The authors demonstrated that the proposed Alexnet outperforms the proposed VGG-16 and other state-of-the-art techniques. Finally, the classification results are sent to healthcare specialists to monitor the patient's condition. Qinghan et al. [43] designed a model based on the RNN to envisage the status of obesity depending on parameters like blood pressure, demographics, and step count, etc. The proposed model had 25 hidden units with a dropout rate of 0.5. And the model was trained for 150 epochs. The experiments show that the model is quite effective in forecasting the improvement of obesity status. The authors intend to apply the proposed model for envisaging the status of hypertension and diabetes. Moreover, the authors also plan to include additional features like diet in order to improve the prediction accuracy. Minh et al. [44] reported the use of RNN to model the succession of Alzheimers disease in the presence of missing data. The authors proposed an RNN with 256 units with Adam as an optimizer and the categorical cross-entropy as a loss function. In order to prevent model over-fitting, variational dropout and L2 regularization were used. The model was trained for 100 epochs. The experiments reveal that the RNN outperforms the conventional models. In Ref. [45], the authors devised a model based on Stacked Sparse AE (SSAE) to classify different stages of sleep. The model consists of three layers: a two-layered SSAE and a softmax layer. The number of hidden units in the SSAE is 20 in the first layer and 12 in the second layer. The proficiency of the proposed model was compared with those of the KNN and softmax classifier, and it was demonstrated that the proposed model outperforms other classifiers.

4.2.4. Missing clinical data prediction

Another important application of DL in healthcare is the missing clinical data prediction. In Ref. [46], the authors proposed a model based on the discriminative RBM to capture high-level features from the clinical data and predict missing values. The proposed model has 10 hidden layers, and the learning rate and the number of epochs were set to 30 and 0.05, respectively. L1 was used as the regularization method in the model. The authors evaluated the proposed model on two datasets, the chronic kidney disease dataset and the dermatology dataset. It was demonstrated that the proposed model shows phenomenal performance in predicting missing values in both datasets. Hemant et al. [47] devised an approach based on the LSTM to forecast the missing data in healthcare scenarios. The proposed approach consists of one LSTM layer, which is followed by a dense layer. Adam was used as an optimizer, and the mean square was used as the loss function. The authors performed experiments with different numbers of LSTM units and window sizes. It was demonstrated that 10 LSTM units and the window sizes of 40 resulted in the minimum RMSE value. Moreover, the proficiency of the proposed model was evaluated against that of LR and Gaussian Process Regression (GPR), and it was demonstrated that the proposed approach outperforms these techniques.

4.2.5. Mortality prediction

DL has been employed for mortality prediction as well. Ying et al. [48] formulated an approach based on an RNN consisting of a Gated Recurrent Unit (GRU) to predict mortality using an intensive care dataset. The GRU was used in order to overcome the vanishing gradient problem faced by the RNN. Two variants of the Gated Recurrent Neural Network (GRNN) were implemented: GRNN with Hierarchical Attention (GRNN-HA) and GRNN without hierarchical attention. The mini-batch stochastic gradient descent was utilized as the optimization technique to mitigate the loss function. It was demonstrated in the study that both GRNN-HA and GRNN outperform LR and SVM, and GRNN-HA gives better performance results than GRNN.

In addition to the above applications, DL has been utilized for medical image synthesis and patient record generation as well. Dong et al. [49] formulated an approach based on the GAN to synthesize Computed Tomography (CT) images from Magnetic Resonance Images (MRI). Edward et al. [50] proposed an amalgam of AE and GAN to synthesize a patients record.

Table 5 summarizes the research works discussed in this sub-section.

4.3. Smart grid

Incorporating the IoT and DL in power systems will facilitate services like power demand forecasting, electricity price prediction, electricity theft detection, anomaly detection, etc. DL has been utilized in numerous

Table 5
DL in smart healthcare use-cases.

Work	IoT use-case	Dataset	DNN Used	Purpose of DNN	Performance Results
[34]	Disease prediction	Electronic Health Record, Medical Image data	CNN	To forecast the risk of cerebral infarction in a patient	Accuracy: 94.8%
[35]	Multiple disease-micro-RNA association prediction	Human Micro-RNA Disease Database (HMDD)	RBM	To forecast multiple types of associations between diseases and micro-RNA	AUC: 0.8606
[36]	Diabetes diagnosis	Health dataset collected from a hospital located in Hubei Province, China	CNN	To predict the risk of diabetes in a patient	–
[37]	Cardiovascular disease prediction	Statlog (Heart) and Heart Disease Database datasets	DBN	To design a model for cardiovascular disease prediction	Accuracy (Statlog): 91.26% Accuracy (Heart Disease Database): 89.78%
[38]	Cardiovascular risk prediction	Sixth Korea National Health and Nutrition Examination Survey (KNHANES-VI) 2013 dataset	DBN	To construct a cardiovascular risk prediction model	Accuracy: 83.9%
[39]	Heart failure prediction	Electronic Health Records	LSTM	To model the risk prediction of heart failure	AUC: 0.894
[40]	Voice pathology detection	SVD database, MEEI database	CNN	To classify the voice signal into normal and pathological categories	Accuracy: 97.5%
[41]	Congestive heart failure	Beth Israel Deaconess Medical Center (BIDMC) CHF, Normal Sinus Rhythm (NSR), Fantasia Database (FD),	LSTM	To detect the congestive heart failure	Accuracy (BIDMC):99.22% Accuracy (NSR): 98.85% Accuracy (FD): 98.92%
[42]	Pathology detection and monitoring	Temple University Hospital dataset	CNN	To classify EEG signals into two categories, pathological and normal	Accuracy: 87.32%
[43]	Obesity status prediction	Electronic medical records, Sensory data from wearables	RNN	To forecast the improvement in obesity status based on parameters like blood pressure, demographics, and step count	Accuracy: 86%
[44]	Alzheimers disease recognition	Alzheimers Disease Neuroimaging Initiative (ADNI) dataset	RNN	To model the succession of Alzheimers disease for seven years	MAE: 5.2 $\pm$ 0.45
[45]	Sleep stage classification	ISRUC-Sleep dataset	AE	Dimensionality reduction, feature extraction, and classification	Accuracy: 82.2%
[46]	Clinical Decision and risk prediction	Chronic kidney disease (CKD) and dermatology datasets	RBM	To capture high-level features from the clinical data and predict missing values	Accuracy (CKD): 97.5% Accuracy (Dermatology): 80%
[47]	Missing data prediction in healthcare	MIT-BIH dataset	LSTM	To predict the missing data in healthcare scenarios	RMSE: 0.07
[48]	Clinical outcome prediction	Medical Information Mart for Intensive Care (MIMIC) dataset	RNN	To predict mortality	F1 score: 0.5766 $\pm$ 0.0623
[49]	Medical Image Synthesis	Brain data from ADNI dataset, Pelvic dataset	GAN	To synthesize Computed Tomography image from Magnetic Resonance Image	MAE (Brain dataset): 18.9 MAE (Pelvic dataset): 4.6
[50]	Generating patient records	Sutter PAMF, MIMIC-III, Sutter Heart Failure	AE + GAN	To synthesize a patients record	–

smart grid use-cases. The following provides a description of those use-cases.

4.3.1. Power demand forecasting

Power demand forecasting is a vital procedure for achieving efficiency and reliability in the operation of power grids. DL has been employed in numerous power demand forecasting use-cases because of its accurate prediction capabilities. Salah et al. [51] reported the utilization of LSTM-RNN for the prediction of short-term and medium-term electric load using France metropolitan's electricity consumption data. The selection of optimal parameters was performed using feature selection and a Genetic Algorithm (GA). The optimal values of parameters as determined by GA are: the number of hidden layers is 6, the number of epochs is 50, the batch size is 125, the activation function is ReLu, and the optimizer is Adam. Experiments carried out in the study reveal that LSTM-RNN outperforms the traditional machine learning techniques. In another study, Xiaoyu et al. [52] devised a novel framework (RBM-Elman) based on the RBM and the Elman neural network to envisage the short-term load in power systems. The purpose of the RBM in the study is to initialize parameters of the Elman neural network in order to get rid of problems like slow rate of convergence and local-optima trapping. The DBN and the Elman neural network are used to compare the performance of the hybrid model. The study reveals that the hybrid model outperforms these two techniques. The authors in Ref. [53] proposed a framework based on the DNN for short term load forecasting consisting of three steps: data processing, model training, and forecasting. They performed forecasting depending on parameters like weather, date and previous electricity usage. The authors compared the performance of the pre-training RBM and the ReLU without pre-training. It was demonstrated that the DNN with ReLU gives more accurate results as compared to the pre-trained RBM. Moreover, it was revealed that DNN with ReLU is easy to train. The authors also presented a comparative analysis of the proposed framework with Shallow Neural Network (SNN), Double Seasonal Holtâ€"Winters (DSHW) model, and ARIMA for predicting the short-term load. The study concludes that the proposed framework outperforms the other techniques. MAPE and RRMSE were used as measures to assess the performance of the aforementioned techniques. Myoungsoo et al. [54] proposed a framework consisting of a number of LSTM networks and a CNN for power demand forecasting. The number of LSTMs depends on the number of features in the dataset. These features include information about the temperature, humidity, and season. Each LSTM network consists of two hidden layers. The output from the LSTMs is fed to a CNN, which comprises of two convolutional layers followed by a max-pooling layer and a fully-connected layer. The CNN outputs the power demand value. The study demonstrates that the hybrid model outperforms the baseline approaches. In Ref. [55], Elena et al. utilized the DBN to learn features from the data in an unsupervised way in order to forecast the energy consumption in the smart grid. The study attained an accuracy of 96.5%.

4.3.2. Electricity price prediction

DL has the ability to predict the electricity prices. This is highly crucial for electricity users as they can adjust their electricity usage based on the predicted values. Waleed et al. [56] formulated a framework for forecasting the price of electricity. The framework consists of three steps: redundancy reduction, dimensionality reduction and price forecasting. The redundancy reduction is performed using the grey correlation analysis, the dimensionality reduction is made using the recursive feature elimination method, and finally, price forecasting is carried out using the CNN. The framework was evaluated on the ISO New England Control Area (ISO NE-CA) 2016 dataset. The experiments carried out in the study reveal that the model outperforms the conventional approaches.

4.3.3. Electricity theft detection

Among the losses of electricity providers, electricity theft has the sternest hazardous consequences. Zibin et al. [57] formulated a novel electricity theft-detection technique based on the CNN. The proposed technique has two CNN components: the deep CNN component and the wide CNN component. The deep CNN component assists in identifying the non-periodic behavior of theft and periodic behavior of normal electric consumption. The wide CNN component extracts the global characteristics from the data. The study demonstrates that the proposed framework outperforms existing benchmark techniques like LR, SVM, Random Forest (RF), and CNN.

4.3.4. Anomaly detection

DL has the ability to detect anomalies in power grid systems. In Ref. [58], the authors devised an RNN-based model to detect anomalies in power systems. The model detects anomalies by keeping track of the reconstruction error. When the value of the reconstruction error becomes unexpectedly high, an anomaly is said to occur. The proposed model attained a precision value of 95%. In another study, Ye et al. [59] presented a framework based on the SSAE and a softmax classifier for anomaly detection in smart grids. The SSAE digs out high-level features from the data collected from smart meters, and the softmax classifier detects anomalies. The study achieved an accuracy of 95.96%.

Apart from the applications discussed above, DL has been applied in the following use-cases as well. In Ref. [60], the authors devised a GAN-based model to integrate data from heterogeneous power sources to facilitate an intelligent distribution system. In another study, Chi et al. [61] utilized the GAN for data generation to understand crucial features in the real dataset and create new data instances based on the understanding of the dataset. Long et al. [62] formulated an AE-based model to monitor the blade breakage of wind turbines. They used the RBM to initialize the parameters of the AE model. The performance of the model was evaluated on blade breakage cases obtained from four Chinese wind farms. The experiments conducted in the study demonstrate that the model is able to identify the blade breakage effectively.

Table 6 provides a summary of the research works described in this sub-section.

4.4. Smart manufacturing

The introduction of the IoT and DL in manufacturing will enable services like machine condition monitoring, fault detection and analysis, remaining useful life prediction, etc. The rationale for this statement stems from the research works presented below.

4.4.1. Machine condition monitoring

Monitoring the condition of machines is highly crucial for mitigating maintenance costs and maximizing production. DL has been identified as a powerful tool for the said purpose. Weishan et al. [63] formulated an LSTM-based approach to forecast the working conditions of the equipment in order to enhance its operation quality. They also developed a feature-engineering workflow to remove outliers from the sensed data and carry out its correlation analysis. The study finds the optimal values of hyper-parameters for the LSTM by applying the orthogonal experimental design strategy. The proficiency of the model was assessed based on the RMSE, and the approach was evaluated against the ARIMA. The experiments carried out in the study demonstrate that the proposed model outperforms the ARIMA. Moreover, the authors stated that the proposed approach could be used for time-series analysis in other industrial IoT applications. Jinjiang et al. [64] formulated a DBN-based approach to forecast the condition of machines in manufacturing systems. The proposed approach consists of three steps: in the first step, the sensor data is normalized with the help of Gaussian neurons present in the visible layer of the DBN. The second step involves the optimization of hyper-parameters using the extremum disturbed and simple particle swarm optimization. The final step involves the model acceleration using the hybrid Modified Liuâ€“Storey Conjugate Gradient (MLSCG) algorithm. The experiments demonstrate that the proposed approach outperforms the traditional benchmark techniques in terms of accuracy as

Table 6
DL in smart grid use-cases.

Work	IoT use-case	Dataset	DNN Used	Purpose of DNN	Performance Results
[51]	Electric load forecasting	France metropolitan's electricity consumption data	LSTM	To predict the electrical load	MAE: 270.4
[52]	Load forecasting	Power load data collected from a city in UK	RBM	To predict the load in power systems	MAPE: 0.0346
[53]	Short-term load forecasting	Dataset provided by Korean Electric Power Corporation	DNN with ReLu	To forecast short-term load	MAPE: 2.19
[54]	Power Demand Forecasting	Korea's daily power demand dataset Provider: Korea Power Exchange	CNN + LSTM	To extract the feature sets from the input	MAPE:0.82%
[55]	Energy prediction	Load data provided by Baltimore Gas and Electric Company	DBN	To learn features from the data in an unsupervised manner	Accuracy: 96.5%
[56]	Electricity price forecasting	Electricity Price data	CNN	To reduce the inconsistency and redundancy between the features	Accuracy: 97.03%
[57]	Electricity theft detection	Electricity Consumption data offered by State Grid Corporation of China (SGCC)	CNN	To identify the abnormal usage of electricity	Mean Average Precision: 0.9565
[58]	Anomaly detection	Power Demand dataset and the Voltage Amplitude and Phase Angle in IEEE 39 bus system	RNN	To detect anomalies in power systems	Precision: 95%
[59]	Anomaly detection	Data obtained from the Operational Energy Management System (OEMS), China	AE	High-level feature extraction	Accuracy: 95.96%
[60]	Intelligent power distribution	Voltage data of 171 towns from 2015 to 16	GAN	To integrate data from heterogeneous power sources	RMSE: 0.1460
[61]	Data generation	Pecan Street Dataset	GAN	To understand crucial features in the real dataset and create new data instances based on the understanding of the dataset	F1 Score: 0.96
[62]	Wind Turbine Blade Breakage Monitoring	Data gathered from wind farms in China	AE	Feature extraction	–

well as speed.

4.4.2. Fault detection and analysis

Fault detection and analysis is the basic requirement of smart manufacturing. DL techniques have been employed to identify faults and diagnose the causes responsible for the faults. In Ref. [65], the authors designed a manufacturing inspection system based on the CNN to classify the production items into defected and non-defected categories and to discover the category and degree of the defect in a product. The designed system implemented the concept of fog computing in order to perform an analysis in real-time. The sensed data is sent to fog nodes for extracting low-level features, and if possible, the inferences are generated at the fog nodes, which is time-saving and avoids traffic congestion. However, if the fog nodes cannot provide accurate inferences, then the output of the fog nodes is forwarded to the cloud for further analysis. Experiments carried out in the study reveal that the designed system is efficient. In Ref. [66], Ye et al. reported the utilization of a CNN to identify and forecast faults and wearing conditions in machines. The experiments conducted in the study demonstrate that the CNN works well and can be used in diverse scenarios, including electrical and mechanical systems. Li et al. [67] devised an approach based on the Dynamic Sparse Stacked Auto-Encoder (DSSAE) to classify faults in dynamic processes. Unlabeled data is used to initialize the weights of the neural network. The proposed approach takes the temporal relationship between data items into consideration and classifies faults in the context of time sequence. The proposed approach was evaluated against SAE and SSAE, and it was demonstrated that the proposed approach outperforms these techniques.

4.4.3. Remaining useful life prediction

Remaining useful life is a very crucial parameter for machine health management. In Ref. [68], the authors proposed a methodology based on regression and Integrated Deep De-noising Auto-encoders (IDDA) for the predicting the remaining useful life of machines. The proposed methodology consists of two De-noising AEs (DDA) and a component for performing linear regression. It predicts the remaining useful life of equipment based on the current state data of the equipment. The data is divided into two frames: distant and current. These two data frames are fed as input to two different DDAs. The outputs from these DDAs are fused, and the fused output is passed through the regression component that outputs the remaining useful life of the equipment. In another study, Lei et al. [69] proposed an approach based on the AE and the DNN for envisaging the remaining useful life of bearings. The proposed method firstly extracts the time-domain characteristics, frequency-domain characteristics, and time-frequency domain characteristics from the vibration signals captured from bearings. Then the time-domain characteristics are fed as input to the AE, which compresses these features. And finally, the frequency-domain characteristics, time-frequency domain characteristics and compressed time-domain characteristics are fed as input to the DNN for obtaining the prediction results. The DNN model consists of 9 layers, and the model is trained for 50 epochs. The study concludes that the proposed approach is quite effective in remaining useful life predictions and outperforms the conventional methods.

4.4.4. Worker activity recognition

DL has been utilized to recognize the activities of workers in manufacturing units. Wenjin et al. [70] formulated a CNN-based approach to classify workers activities into six groups, namely, grab tool, use-power screwdriver, hammer nail, turn a screwdriver, rest arm, and use wrench. Two types of signals were captured from the worker's armband: Inertial Measurement Unit (IMU) and surface Electromyography (sEMG). These two types of signals were fed as input to the CNN, which outputs the probability distribution of activities. The study achieved an accuracy of 98%.

In addition to the applications discussed above, DL has been utilized for intrusion detection as well. Seong-Taek et al. [71] designed a system for detecting intrusions based on the AE in a smart factory. The analysis carried out in the study demonstrates that the designed system outperforms the traditional systems.

Table 7 sums up the research works discussed in this sub-section.

4.5. Smart agriculture

Integrating the IoT and DL in agriculture will result in services like plant classification, plant disease detection, crop yield prediction, fruit counting, weed detection, and so on, thereby transforming traditional agriculture into smart agriculture. The following subsection provides a description of research works that have employed DL in agricultural use-cases.

4.5.1. Plant classification

After great success in image recognition, DL is employed for the classification of plants. In Ref. [72], the authors framed a model based on

Table 7
DL in Smart manufacturing use-cases.

Work	IoT use-case	Dataset	DNN Used	Purpose of DNN	Performance Results
[63]	Industrial IoT equipment analysis	Sensory data obtained from 33 sensors deployed on a pump in power station	LSTM	To predict the working conditions of the equipment in order to enhance its operation quality	RMSE: 0.004
[64]	Condition prediction	Sensory data gathered from a centrifugal compressor	DBN	To predict the condition of machines in manufacturing systems	MSE: 0.0157
[65]	Manufacture inspection system	Sensory data	CNN	To classify the production items into defected and non-defected and to discover the category of defect	–
[66]	Diagnosis and monitoring in manufacturing	Bearing data offered by Case Western Reserve University (CWRU)	CNN	To identify and forecast faults and wearing conditions in machines	Accuracy: 100%
[67]	Fault classification	Process data samples	AE	To learn features from a variety of faults	Accuracy: 90.2%
[68]	Remaining useful life prediction in machines	Data collected from CNC machining centers	AE	To extract the features that are crucial for forecasting the lasting useful life	–
[69]	Remaining useful life prediction of bearings	IEEE PHM2012 data provided by the FEMTO-ST Institute in France	AE	To extract the features that are crucial for forecasting the lasting useful life of bearings	Mean squared error: 0.200
[70]	Worker activity recognition	Sensory data	CNN	To classify worker's activities into six groups, namely, grab tool, use-power screwdriver, hammer nail, turn a screwdriver, rest arm, and use wrench	Accuracy: 98%
[71]	Intrusion detection system	Sensory data, network traffic data	AE	To design an intrusion detection system	Improvement in detection rate: 29% −1.29

the CNN to classify legumes into three categories: soya-bean, red beans, and white beans, depending on the vein patterns of their leaves. They also investigated the effect of varying the CNN depth (2 layers to 6 layers) on the model accuracy. The results show that the accuracy of the model increases with an increase in the depth of CNN.

4.5.2. Plant disease detection

Early diagnosis of plant diseases is crucial in precision agriculture. In Ref. [73], the authors utilized the CNN to classify the leaf images into healthy and diseased categories and find the category of the disease that a plant suffers from. They demonstrated that the CNN was able to extract the color and texture features of lesions corresponding to specific diseases. Moreover, the authors identified that 75% of parameters in the CNN model were not important for diagnosis. The study achieved an accuracy of 97.14%. In Ref. [74], the authors formulated a CNN-based approach to classify leaf images into healthy and diseased categories. The proposed model attained an accuracy of 99.53%.

4.5.3. Crop yield prediction

Crop yield prediction is of utmost importance in the agricultural industry. In Ref. [75], Zehui et al. devised an LSTM-based model to predict the yield of corn. Experiments conducted in the study demonstrate that the proposed approach yields accurate results for predicting the corn yield.

4.5.4. Fruit counting

Another important application of DL in agriculture is fruit counting. Maryam et al. [76] utilized the CNN to predict the number of tomatoes. They utilized the Inception-ResNet based architecture for the CNN. The results show that the proposed CNN can accurately calculate the number of tomatoes even under the conditions of shadow and overlap, etc. Moreover, the authors stated that the proposed approach could be used to count other fruits as well. However, the model was not able to count un-ripened fruits.

4.5.5. Weed detection

Weed detection is important for improving crop yield and quality. Huasheng et al. [77] utilized the CNN to classify the input images into three categories: rice, weed, and others. They implemented the concept of transfer learning and utilized three pre-trained CNN architectures: VGG16, Alexnet and Googlenet, for enhancing the generalization potential of the neural network. Moreover, the skip architecture was used to enhance the prediction performance. The results show that VGG-16 outperformed the other two pre-trained architectures.

Apart from the applications described above, DL has been utilized in the following use-cases as well. Khurshid et al. [78] designed a system to obtain the data related to the temperature, humidity and soil-moisture content of plants and proposed an approach based on the RNN to estimate the maximum and minimum temperature values for ten days. In Ref. [79], Clement et al. devised a CNN and SVM based model to categorize images into two classes: root and soil. They implemented the concept of transfer learning to extract high-level features from X-ray tomographic images. These features were then fed as input to the SVM that outputs the classification result. Xiaodong et al. [80] utilized the DBN to forecast the moisture content present in the soil. The computational analysis carried out in the study reveals that the DBN outperforms the conventional techniques. Gunjan et al. [81] established an LSTM-based model to forecast the weather and soil attributes of a region for crop planning.

Table 8 summarizes the research works discussed in this sub-section.

Table 8
DL in smart agriculture use-cases.

Work	IoT use-case	Dataset	DNN Used	Purpose of DNN	Performance Results
[72]	Plant Identification	Dataset consisting of vein leaf images of soybean, red beans, and white beans	CNN	To classify legumes into three categories: soybean, red beans, and white beans	Accuracy (soybean): 98.8% Accuracy (red bean): 98.3% Accuracy (white bean): 90.2%
[73]	Plant disease diagnosis	Plant Village dataset	CNN	To classify the leaf images into healthy and diseased categories and find the category of disease	Accuracy: 97.14%
[74]	Plant disease detection	Leaf images of plants	CNN	To classify images into healthy and diseased categories	Accuracy: 99.53%
[75]	Corn yield prediction	Data obtained from the National Agricultural Statistics Service (NASS) Quick Stats	LSTM	To forecast the yield of corn	MSE: 191.05
[76]	Tomato counting	Dataset consisting of 24,000 images	CNN	To forecast the count of tomatoes	RMSE: 1.16
[77]	Weed mapping in smart agriculture	Data collected using multi-rotor UAV	CNN	To classify the input images into three categories; rice, weed, and others	Accuracy: 88.3%
[78]	Internet of plants based system	Sensory data	RNN	To envisage the maximum and minimum temperature values for ten days	MSE: 0.8826
[79]	Root and soil segmentation	X-ray tomographic images of soil	CNN	To categorize images into two classes: root and soil	Quality Measure: 0.57
[80]	Soil moisture content prediction	Data gathered from corn field (irrigated) in China	DBN	To forecast the moisture content present in the soil	RMSE: 6.77
[81]	Weather prediction	Dataset from Syngeta crop challenge (2016)	LSTM	To predict weather according to the conditions of the preceding year	Mean Relative Error: 1–3%

4.6. Smart transportation

Smart transportation incorporates a wide range of technologies in order to facilitate an efficient, reliable, and safer transportation system. DL is a potential solution for facilitating services like traffic congestion management, traffic speed prediction, obstacle detection, accident detection, car park occupancy prediction, etc. The following subsection provides the motivation for introducing DL into transportation systems.

4.6.1. Traffic congestion management

Traffic congestion management is the fundamental requirement for the design of an intelligent transportation system. Shidrokh et al. [82] formulated a DBN and Firefly Algorithm (DFA) based approach for predicting traffic flows. They proposed a DBN consisting of three layers to extract features from time-series data. DFA was utilized to fine-tune the hyper-parameters of DBN like number of layers, number of units per layer, and learning rate. The experiments conducted in the study reveal that the proposed approach outperforms the benchmark traffic flow forecasting techniques. The proposed approach can assist in avoiding the traffic congestion problem and can provide suggestions to traffic managers. In another study, Shengdong et al. [83] devised an amalgam of CNN and Gated Recurrent Units (GRU) with an attention mechanism for predicting traffic flows. The purpose of CNN in the devised approach is to extract the local characteristics, and that of GRU is to model the long-term dependencies. The proposed architecture consists of multiple CNN-GRU-attention modules for processing multi-modal features like speed, journey time, weather, etc. Simulation studies carried out in the study reveal that the approach is quite effective for traffic flow prediction and is better than conventional techniques. Xianglong et al. [84] utilized the KNN and the LSTM to model the spatio-temporal dynamics of traffic flows. The study identified a test station and used the KNN to select the K stations related to the test station. A multi-layer LSTM predicted the traffic in all these stations. The final prediction is carried out using the rank-exponent weighting technique. The study attained an accuracy of 95.75%. Wang et al. [85] formulated a framework to model the traffic flow in urban road networks. The proposed framework consists of two steps. The first step is data processing that involves data padding, removal of outliers and abnormal data points. The techniques utilized for data processing include distance-based outlier detection algorithm, wavelet-based similarity measurement approach and data padding method. The second step involves traffic prediction using the LSTM-RNN. Experiments conducted in the study demonstrate that the proposed framework outperforms conventional techniques. In Ref. [86], Juan et al. utilized the RBM to predict the traffic conditions. The study implements the concept of fog computing for performing predictions in real-time. Yuhan et al. [87] formulated the DBN-based and the LSTM-based models to predict the traffic flow under different scenarios of rainfall. The study demonstrates that the introduction of a rainfall factor enhances the prediction accuracy as rainfall affects the flow of traffic significantly. Moreover, it was revealed that the LSTM performs better than the DBN in extracting features from the traffic-flow data. In Ref. [88], the authors proposed the LSTM-based and the CNN-based models to forecast the traffic patterns around a node in a road network. The proposed LSTM consisted of two layers with 20 hidden units each. And the CNN consisted of two convolutional and two fully-connected layers. Simulation studies demonstrate that both models successfully predicted the traffic congestion in transportation networks, and the CNN outperformed the LSTM. Sen et al. [89] created a dataset based on image analysis for traffic congestion. They also proposed a model, Deep Congestion Prediction Network (DCPN), based on the AE to represent the temporal dynamics of a network and forecast the congestion in the traffic. The proposed approach consists of an AE along with two fully connected layers. The performance of the proposed model was evaluated on the created dataset. The experiments in the study reveal that the proposed model outperforms the benchmark techniques. Xiaoguang et al. [90] utilized the RBM to extract temporal and spatial features from the input data to model traffic flow predictions. Xiaolei et al. [91] proposed a model based on the RBM and RNN to forecast the traffic congestion evolution from GPS data. They utilized a conditional RBM which consisting of a feedback loop between the visible and the hidden layers to analyze temporal data. The proposed model successfully extracts the spatio-temporal characteristics from the input data. The performance of the proposed model was evaluated against the backpropagation neural network and the SVM, and it was demonstrated that the proposed model outperforms these techniques.

4.6.2. Traffic speed prediction

Traffic speed prediction is another important issue in traffic management. In Ref. [92], Xiaolei et al. utilized the CNN to learn the traffic as images and forecast the speed of the traffic network. They converted traffic into images with two dimensions: time and space. The proposed CNN architecture consists of three convolutional layers, three pooling layers and a fully connected layer. Simulation results show that the proposed CNN performs better than other benchmark techniques. In Ref. [93], the authors proposed a hybrid model (LC-RNN) based on the RNN and CNN to forecast the traffic speed. The proposed model utilizes the strengths of both CNN and RNN to perform accurate predictions.

4.6.3. Smart shift decision

The smart shift decision system is a fundamental requirement in self-driving cars. Zengcai et al. [94] established an AE-based model to forecast shifting gear timing based on throttle percentage, vehicle velocity, and acceleration. Simulation results demonstrate that the model can not only produce accurate results, but also is quite economical and efficient as well.

4.6.4. Obstacle detection and accident detection

DL has also been employed for obstacle detection and accident detection in intelligent transportation systems. Abdelkader et al. [95] formulated an amalgam of the Deep Stacked Auto-encoder (DSA) and the KNN for obstacle detection. The DSA extracts high-level features from the data by utilizing characteristics like greedy learning and dimensionality-reduction. The KNN detects the obstacles in an unsupervised manner. The study compares the formulated approach with the DBN, and concludes that the proposed model outperforms the DBN. In Ref. [96], the authors designed an AE-based approach for road accident detection from surveillance video.

4.6.5. Car park occupancy prediction

Another crucial application of DL in intelligent transportation systems is the prediction of the availability of parking space in a specific area. Andres et al. [97] formulated an RNN-based model to estimate the rate of occupancies of car parks. The simulation results show that the proposed system is quite effective and performs better than the conventional methods.

In addition to the applications discussed above, DL has been utilized for path planning and transportation mode detection. Mehdi et al. [98] utilized the GAN for path planning to generate safe and reliable paths. Xuan et al. [99] designed a deep transport framework based on the LSTM to predict human movement and detect the transportation mode. In Ref. [100], Weitao et al. designed an LSTM-based system to identify the mode of transport using a kinetic energy harvester.

Table 9 provides a summary of the research works discussed in this sub-section.

5. Case study: human activity recognition (HAR)

This section proposes a model for HAR based on the LSTM. The experiments were carried on the WISDM activity prediction dataset. The WISDM activity prediction dataset was collected by the Wireless Sensor Data Mining Laboratory. The dataset consists of accelerometer values (time-series data) collected at a sampling frequency of 20 Hz from 36 subjects. It consists of 1,098,207 rows and six columns. These columns include: user ID, timestamp, accelerometer value in X-direction (acc-x), accelerometer value in Y-direction (acc-y), accelerometer value in Z-direction (acc-z), and activity-name. The reason why LSTM is employed in this study for activity prediction is the phenomenal performance shown by the LSTM in modelling problems with time-series data. Fig. 3 shows the proposed framework. The proposed model performs the time-series analysis of the accelerometer values and outputs the activity name.

Table 9
DL in smart transportation use-cases.

Work	IoT use-case	Dataset	DNN Used	Purpose of DNN	Performance Results
[82]	Traffic flow prediction	Data containing historical road traffic flow	DBN	To design a model for predicting traffic flows	RMSE: 6.1
[83]	Traffic flow forecasting	Highways England dataset	CNN + GRU	To extract the features from input data	RMSE: 4.35
[84]	Traffic flow forecasting	Traffic flow data Provider: Transportation Research Data Lab (TDRL), University of Minnesota Duluth (UMD)	LSTM	To model the temporal dynamics of the traffic flow	Accuracy: 95.75%
[85]	Short-term traffic prediction	Floating Car Data	LSTM	To model the traffic flow in urban road networks	Accuracy: 95%
[86]	Real-time traffic forecasting	Floating car data gathered in Barcelona	RBM	To predict traffic in real-time	–
[87]	Traffic flow prediction in rain	Transportation data obtained from Deshengmen to Madian, Beijing, China	DBN + LSTM	To forecast the traffic flow under different scenarios of rainfall	MAPE: 11.69%
[88]	Traffic congestion prediction	Traffic data collected from Traffic Control Stations, Northern California	LSTM	To forecast the traffic patterns around a node in a road network	–
[89]	Traffic congestion prediction	Seattle Area Traffic Congestion Status (SATCS) dataset	AE	To capture temporal dynamics of a network and forecast congestion in the traffic	MAE: 0.0095
[90]	Traffic prediction based route finding	Dataset collected from Wuhan city	RBM	To extract temporal and spatial features from the input data in order to model traffic flow predictions	Precision: 0.85
[91]	Congestion Evolution Prediction in the transportation network	GPS data	RBM + RNN	To forecast the traffic congestion evolution from GPS data	Accuracy: 88%
[92]	Traffic speed prediction	GPS data	CNN	To learn the traffic as images and predict the speed of the traffic network	Accuracy: 94.2%
[93]	Traffic speed prediction	Trajectory data from Beijing and Shanghai	RNN + CNN	To forecast the traffic speed	RMSE: 4.686
[94]	Smart shift decision method	Vehicular data gathered with different road conditions	AE	To forecast shifting gear timing based on throttle percentage, vehicle velocity, and acceleration	Accuracy: 94.38%
[95]	Obstacle detection	Malaga Stereo-Vision Urban Dataset (MSVUD), Daimler Urban Segmentation Dataset (DUSD), and Bahnhof dataset	AE	To alleviate the number of features in the input	AUC (MSVUD, DUSD): 0.91 AUC (Bahnhof Dataset): 0.94
[96]	Road Accident detection	Traffic videos	AE	To extract Spatio-temporal features from the surveillance video	AUC: 0.8106
[97]	Car park occupancy prediction	Data from 29 Parking slots in Birmingham	RNN	To forecast the rate of occupancies of car parks	MAE: 0.028
[98]	Path planning	Localization dataset, Path planning dataset	GAN	To generate safe and reliable paths	Accuracy: 99%
[99]	Human mobility and transportation mode prediction	GPS data and transportation network data	LSTM	To forecast the movements of humans and detect the transportation mode	Precision accuracy: 83.26%
[100]	Mode detection system	38.6 h of transportation data	LSTM	To identify the mode of transport based on kinetic energy harvester	Accuracy: 97%

The methodology espoused is given as

- *Dataset split:* the HAR dataset is split in the ratio of 80:20. Eighty percent of the data is utilized for training the model, and twenty percent is used for testing purposes.
- *Model training:* The training dataset is utilized for training the LSTM model. Fig. 4 presents the proposed LSTM model. The model comprises of an input layer, two hidden layers, and an output layer. The number of units in each hidden layer is 64. Rectified linear unit is used as an activation function in the hidden layers, and the softmax is used in the output layer. Each LSTM unit consists of a memory cell and three control gates: the input gate, the forget gate, and the output gate. These gates perform the following computations:

$$f_t = g\left(w_f.\left[h_{(t-1)}, x_t\right] + b_f\right)$$

where f_t represents the forget gate, w_f is the weight matrix associated with the forget gate, $h_{(t-1)}$ is the previous output, x_t is the input, b_f represents the bias vector for the forget gate and $g()$ represents the ReLU activation function.

$$i_t = g(w_i.[h_{(t-1)}, x_t] + b_i)$$

where i_t represents the input gate, w_i and b_i represent the weight matrix and the bias vector associated with the input gate.

$$o_t = g(w_o.[h_{(t-1)}, x_t] + b_o)$$

where o_t represents the input gate, w_o, b_o represent the weight matrix and the bias vector associated with the input gate.

$$(c_t) = tanh(w_c.[h_{(t-1)}, x_t] + b_c)$$

where (c_t) represents the update in the memory cell, w_c and b_c represent the weight matrix and the bias vector associated with the memory cell.

$$c_t = f_t * c_{(t-1)}) + i_t * (c_t)$$

where c_t represents the updated memory cell.

$$h_t = o_t * tanh(c_t)$$

where h_t represents the output of the LSTM unit.

The experiments were conducted on an Intel (R) Core i3-6006U CPU@ 2.00 GHz system with 4 GB RAM. The proposed model was designed on Python 3.5 using Keras and Tensorflow. Table 10 lists the various parameter values for the proposed model. Training is carried out on the dataset in batches of size 1024 with a learning rate of 0.0025. Sixty epochs are performed on the dataset with Adam as an optimizer and the softmax cross-entropy as the loss function. Fig. 5 presents a plot between the training accuracy and the number of epochs.

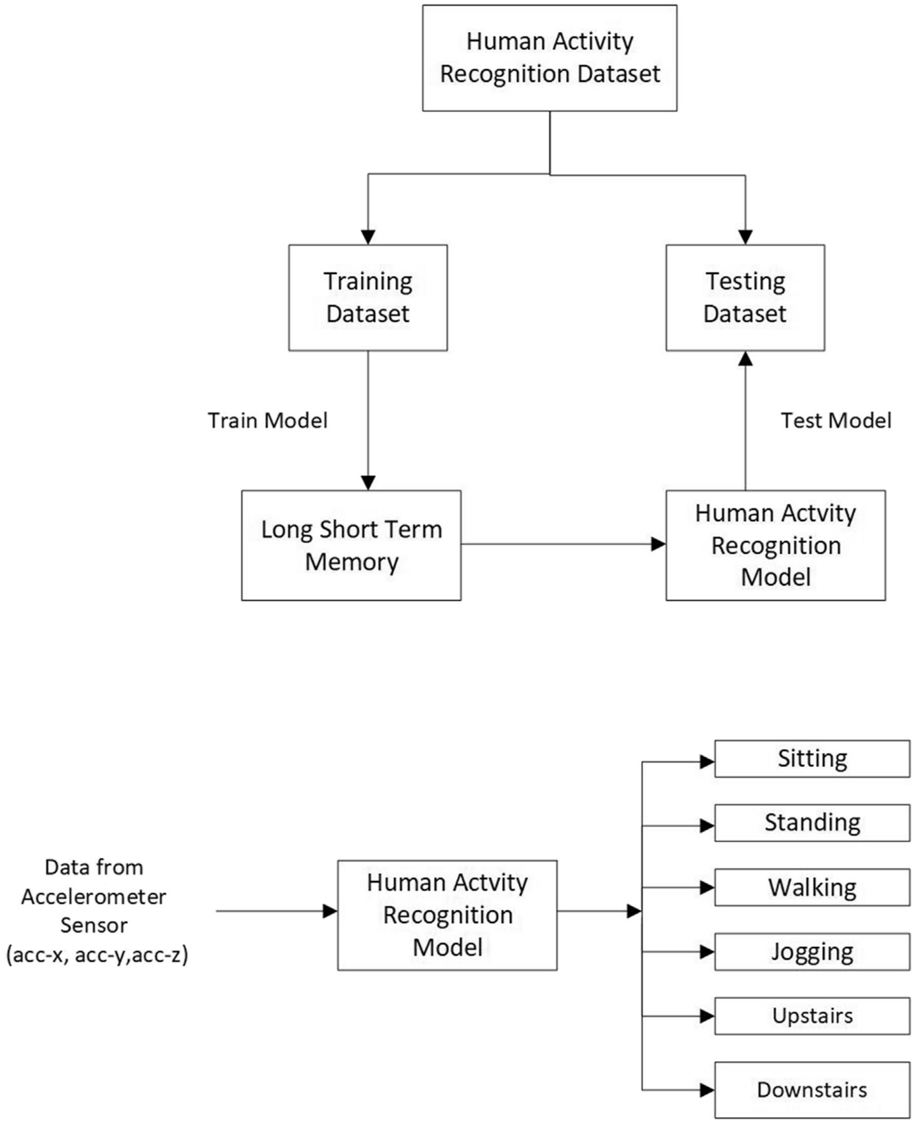

Fig. 3. Proposed framework for HAR.

- *Testing:* the model is tested on the test dataset. The proficiency of the model is assessed based on accuracy percentage. Accelerometer readings (acc-x, acc-y, acc-z) are fed as the input to the model, and the output is the activity-name. Six activities are recognized by the model, including sitting, standing, walking, jogging, upstairs, and downstairs.
- *Performance comparison:* the performance of the proposed model is evaluated against the following machine learning techniques: LR, decision tree, SVM, and RF (Table 11 presents the model parameters of these techniques). This is done to delineate the superiority of DL models (LSTM in this case) in generating accurate results in comparison with the conventional machine learning techniques. Table 12 presents the accuracies of all these classifiers, including the proposed model. Fig. 6 plots the accuracies of these classifiers. It is obvious from the figure that the proposed model outperforms all other classifiers in terms of accuracy. The proposed model yields an accuracy of 98%, which is much better than the accuracies of other classifiers.

6. Challenges

Although DL techniques have been leveraged in various IoT applications, the technology still needs to embrace various challenges to experience the thrill of victory. In this section, we highlight these challenges:

- *Need to repeatedly collect and manage data:* the entire performance and accuracy of DL algorithms depend on their data sources. This data has to be clean to be useful. As such, the implementation of proper data collection and management tools is a significant challenge. Noisy and unlabeled data are other issues that arise because of the humongous amounts of IoT generated data.
- *Training deep models:* as we all know, the deeper the depth of DL network, the higher the ability to extract key features. The training of such a DL network calls for complicated tasks and problems like gradient vanishing, over-fitting, etc.
- *Hardware constraints:* IoT devices are constrained resource-wise. As such, the implementation of deep models on them remains a challenge. This challenge is addressed in literature in two ways: 1) treating end IoT devices only as data collectors; 2) reducing the network complexity to let end IoT devices perform some learning tasks. However, none of the solutions has been exploited properly.
- *High velocity of IoT data:* it brings overwhelming problems and requirements for high-speed analysis and data processing. This, in turn,

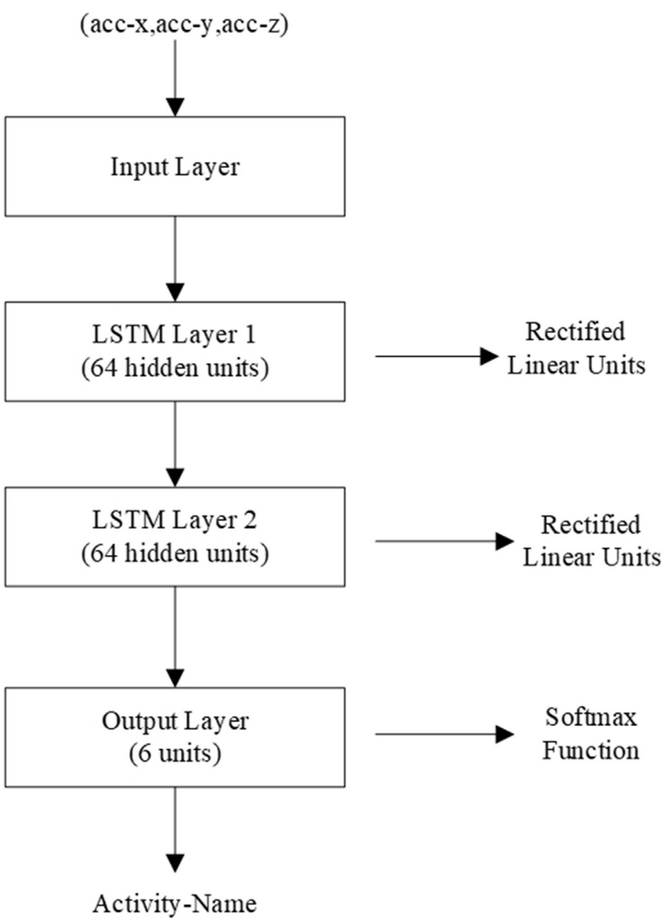

Fig. 4. Proposed model.

Table 10
Hyper-parameter values for LSTM.

Hyper-parameter	Value
Number of training samples	43,921
Number of testing samples	10,980
Number of time steps	200
Number of features	3
Number of output classes	6
Batch size	1024
Number of Epochs	60
Learning Rate	0.0025
Optimizer	Adam
Loss Function	Softmax-cross-entropy

Table 11
Model parameters.

Technique	Parameters
Logistic Regression	Number of iterations: 100, Regularization: L2, Optimizer: Adam
Decision Tree	Maximum depth: 8, min samples leaf: 1, min samples split: 2
Support Vector Machine	Kernel: Radial basis function
Random Forest	Maximum depth: 5, Number of estimators: 60, min samples leaf: 1, min samples split: 2

brings the challenges of the need for huge computation powers, high-performance GPU, high storage requirements, etc.

7. Future aspects

The challenges discussed in the previous section create plenty of opportunities and fertile ground for future research approaches. This section points out the future directions of pivotal DL models:

Table 12
Accuracies of classifiers.

Classifier	Accuracy (%)
Logistic Regression	90.3
Decision Tree	84.75
Support Vector Machine	89.26
Random Forest	87.54
Proposed Model	98

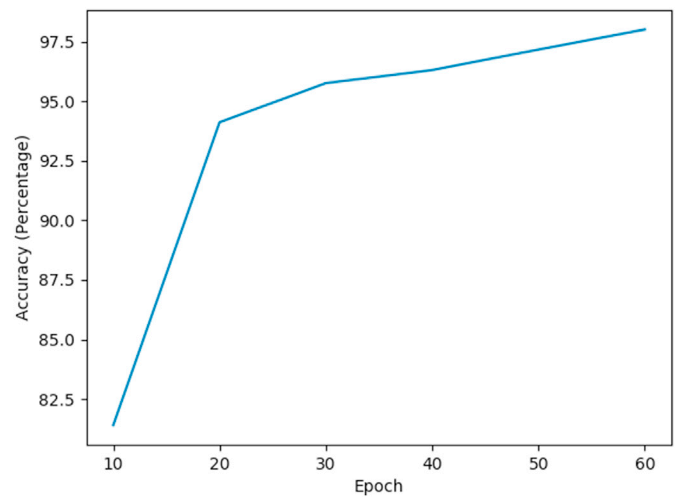

Fig. 5. Accuracy versus epoch.

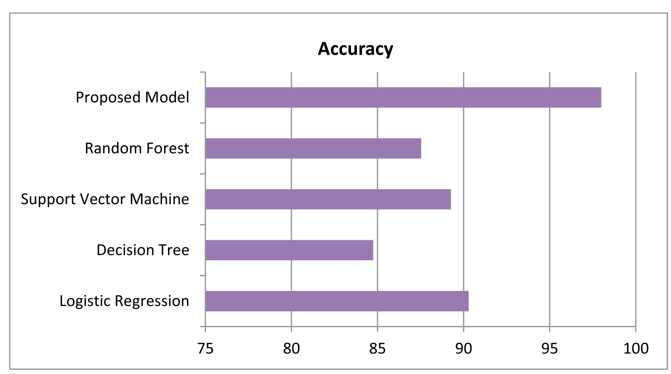

Fig. 6. Accuracies of classifiers.

- The interpretability of DL will receive great attention in the future because its weak reasoning and data understanding ability make it a black box solution.
- It is difficult for DL to model simultaneously various complex data modalities. Yet another trending direction in the research is the multi-modal DL.
- DL requires comprehensive labeled datasets to train the machines and to predict the concealed data. This challenge is even more difficult when the datasets available are limited in size or need to be processed in real time. Therefore, solutions to mitigate this issue could be worked upon.
- A special feature of the IoT device is its mobility. The research on the successful method of integrating these data with DL methods paves the way for improving IoT services. Work can also be done on integrating the context-based information to the sensor data obtained from IoT devices.

- Semi-supervised or unsupervised frameworks are required in DL. At present, most algorithms are supervised, which require high volumes of labeled training data which is always difficult to obtain.
- DL methods could be employed to monitor logs for assuring the security of cyber-physical systems and IoT systems.
- Investing ways to maintain huge data repositories.

8. Conclusion

Extracting actionable insights from raw IoT data is an extremely difficult task that goes beyond the proficiency of traditional data analytical paradigms. DL presents a perfect solution for a variety of classification and prediction tasks in the IoT as it has the potential to learn hierarchical representations from the input data. Moreover, DL is suitable for modelling intricate behaviors of heterogeneous datasets. It consists of multiple architectures with diverse applications. The CNN works superbly well with image and voice data. The RNN and the LSTM are utilized to predict time series. AEs are utilized for dimensionality reduction of high-dimensional data. GANs are suitable for noisy scenarios. The RBM and the DBN capture complex representations of data in an unsupervised manner. These diverse applications of DL models make them very suitable for IoT scenarios. For this purpose, this paper presented a comprehensive review of the research works dedicated to DL-driven IoT use-cases. Moreover, the paper also proposed an LSTM-based model for HAR and carried out a performance comparison of the proposed model with other machine learning techniques in order to determine the advantages of the DL model over these techniques.

Although DL models outperform conventional machine learning procedures, the computational intricacy of these models imperils the utilization of DL models in time-stringent IoT applications. Hence, investigating the ways that will alleviate the computational complexities of these models without the loss of accuracy is highly crucial. Finally, it is worth noting that the introduction of DL into IoT systems will bring a wider range of application-oriented and business-specific services.

Declaration of competing Interest

None.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: a survey on enabling technologies, protocols, and applications, IEEE communications surveys & tutorials 17 (4) (2015) 2347–2376.
[2] S.R. Zahra, M.A. Chishti, Assessing the services, security threats, challenges and solutions in the internet of things, Scalable Computing, Practice and Experience 20 (3) (2019) 457–484.
[3] S.R. Zahra, M.A. Chishti, Ransomware and internet of things: a new security nightmare, in: 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2019, pp. 551–555.
[4] E. Ahmed, I. Yaqoob, I.A.T. Hashem, I. Khan, A.I.A. Ahmed, M. Imran, A.V. Vasilakos, The role of big data analytics in internet of things, Comput. Network. 129 (2017) 459–471.
[5] S. R. Zahra, M. A. Chishti, Fuzzy logic and fog based secure architecture for internet of things (flfsiot), Journal of ambient intelligence and humanized computing. doi: 10.1007/s12652-020-02128-2.
[6] K. Panetta, Gartner's top 10 strategic technology trends for 2017, Smarter with Gartner 18.
[7] X.-W. Chen, X. Lin, Big data deep learning: challenges and perspectives, IEEE access 2 (2014) 514–525.
[8] T.J. Saleem, M.A. Chishti, Data analytics in the internet of things: a survey, Scalable Computing, Practice and Experience 20 (4) (2019) 607–630.
[9] S. Srinivas, R.K. Sarvadevabhatla, K.R. Mopuri, N. Prabhu, S.S. Kruthiventi, R.V. Babu, A taxonomy of deep convolutional neural nets for computer vision, Frontiers in Robotics and AI 2 (2016) 36.
[10] A. Sherstinsky, Fundamentals of Recurrent Neural Network (Rnn) and Long Short-Term Memory (Lstm) Network, arXiv preprint arXiv:1808.03314.
[11] P. Baldi, Autoencoders, unsupervised learning, and deep architectures, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, 2012, pp. 37–49.
[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[13] A. Fischer, C. Igel, An introduction to restricted Boltzmann machines, in: Iberoamerican Congress on Pattern Recognition, Springer, 2012, pp. 14–36.
[14] N. Lopes, B. Ribeiro, Machine Learning for Adaptive Many-Core Machines: A Practical Approach.
[15] M. Chen, S. Mao, Y. Liu, Big Data: A survey, Mobile networks and applications 19 (2) (2014) 171–209.
[16] T. Li, Y. Liu, Y. Tian, S. Shen, W. Mao, A storage solution for massive iot data based on nosql, in: 2012 IEEE International Conference on Green Computing and Communications, IEEE, 2012, pp. 50–57.
[17] S. Xie, Z. Chen, Anomaly Detection and Redundancy Elimination of Big Sensor Data in Internet of Things, arXiv preprint arXiv:1703.03225.
[18] M. Ma, P. Wang, C.-H. Chu, Data management for internet of things: challenges, approaches and opportunities, in: 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, IEEE, 2013, pp. 1144–1151.
[19] H. Cai, B. Xu, L. Jiang, A.V. Vasilakos, Iot-based big data storage systems in cloud computing: perspectives and challenges, IEEE Internet of Things Journal 4 (1) (2016) 75–87.
[20] T.J. Saleem, M.A. Chishti, Deep learning for internet of things data analytics, Procedia Computer Science 163 (2019) 381–390.
[21] H. Shi, M. Xu, R. Li, Deep learning for household load forecasting-a novel pooling deep rnn, IEEE Transactions on Smart Grid 9 (5) (2017) 5271–5280.
[22] M.S. Munir, S.F. Abedin, M.G.R. Alam, C.S. Hong, et al., Rnn based energy demand prediction for smart-home in smart-grid framework, Available at: https://pdfs.semanticscholar.org/4484/29b7aad6c4598c2e0c72ab41cdc1b85f73cc.pdf, 2017, 437-439.
[23] K. Zaouali, R. Rekik, R. Bouallegue, Deep learning forecasting based on auto-lstm model for home solar power systems, in: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2018, pp. 235–242.
[24] F.C.C. Garcia, C.M.C. Creayla, E.Q.B. Macabebe, Development of an intelligent system for smart home energy disaggregation using stacked denoising autoencoders, Procedia Computer Science 105 (C) (2017) 248–255.
[25] Y. Fan, X. Lu, D. Li, Y. Liu, Video-based emotion recognition using cnn-rnn and c3d hybrid networks, in: Proceedings of the 18th ACM International Conference on Multimodal Interaction, 2016, pp. 445–450.
[26] E. Kanjo, E.M. Younis, C.S. Ang, Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection, Inf. Fusion 49 (2019) 46–56.
[27] A. Wang, G. Chen, C. Shang, M. Zhang, L. Liu, Human activity recognition in a smart home environment with stacked denoising autoencoders, in: International Conference on Web-Age Information Management, Springer, 2016, pp. 29–40.
[28] S. Park, J. Park, M. Al-Masni, M. Al-Antari, M.Z. Uddin, T. Kim, A depth camera-based human activity recognition via deep learning recurrent neural network for health and social care services, Procedia Computer Science 100 (100) (2016) 78–84.
[29] X. Gao, H. Luo, Q. Wang, F. Zhao, L. Ye, Y. Zhang, A human activity recognition algorithm based on stacking denoising autoencoder and lightgbm, Sensors 19 (4) (2019) 947.
[30] F.J. Ordóñez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, Sensors 16 (1) (2016) 115.
[31] N. Tax, Human activity prediction in smart home environments with lstm neural networks, in: 2018 14th International Conference on Intelligent Environments (IE), IEEE, 2018, pp. 40–47.
[32] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, C.J. Spanos, Deepsense: device-free human activity recognition via autoencoder long-term recurrent convolutional network, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6.
[33] S.J. Desai, M. Shoaib, A. Raychowdhury, An ultra-low power, "always-on" camera front-end for posture detection in body worn cameras using restricted boltzman machines, IEEE transactions on multi-scale computing systems 1 (4) (2015) 187–194.
[34] M. Chen, Y. Hao, K. Hwang, L. Wang, L. Wang, Disease prediction by machine learning over big data from healthcare communities, Ieee Access 5 (2017) 8869–8879.
[35] X. Chen, C.C. Yan, X. Zhang, Z. Li, L. Deng, Y. Zhang, Q. Dai, Rbmmmda: predicting multiple types of disease-microrna associations, Sci. Rep. 5 (2015) 13877.
[36] M. Chen, J. Yang, J. Zhou, Y. Hao, J. Zhang, C.-H. Youn, 5g-smart diabetes: toward personalized diabetes diagnosis with healthcare big data clouds, IEEE Commun. Mag. 56 (4) (2018) 16–23.
[37] P. Lu, S. Guo, H. Zhang, Q. Li, Y. Wang, Y. Wang, L. Qi, Research on improved depth belief network-based prediction of cardiovascular diseases, Journal of healthcare engineering (2018), https://doi.org/10.1155/2018/8954878.
[38] J. Kim, U. Kang, Y. Lee, Statistics and deep belief network-based cardiovascular risk prediction, Healthcare informatics research 23 (3) (2017) 169–175.
[39] G. Maragatham, S. Devi, Lstm model for prediction of heart failure in big data, J. Med. Syst. 43 (5) (2019) 111.
[40] M. Alhussein, G. Muhammad, Voice pathology detection using deep learning on mobile healthcare framework, IEEE Access 6 (2018) 41034–41041.
[41] L. Wang, X. Zhou, Detection of congestive heart failure based on lstm-based deep network via short-term rr intervals, Sensors 19 (7) (2019) 1502.
[42] S.U. Amin, M.S. Hossain, G. Muhammad, M. Alhussein, M.A. Rahman, Cognitive smart healthcare for pathology detection and monitoring, IEEE Access 7 (2019) 10745–10753.

[43] Q. Xue, X. Wang, S. Meehan, J. Kuang, J.A. Gao, M.C. Chuah, Recurrent neural networks based obesity status prediction using activity data, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 865–870.
[44] M. Nguyen, N. Sun, D.C. Alexander, J. Feng, B.T. Yeo, Modeling alzheimer's disease progression using deep recurrent neural networks, in: 2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI), IEEE, 2018, pp. 1–4.
[45] S. Najdi, A.A. Gharbali, J.M. Fonseca, Feature transformation based on stacked sparse autoencoders for sleep stage classification, in: Doctoral Conference on Computing, Electrical and Industrial Systems, Springer, 2017, pp. 191–200.
[46] M. Sun, T. Min, T. Zang, Y. Wang, Cdl4cdrp: a collaborative deep learning approach for clinical decision and risk prediction, Processes 7 (5) (2019) 265.
[47] H. Verma, S. Kumar, An accurate missing data prediction method using lstm based deep learning for health care, in: Proceedings of the 20th International Conference on Distributed Computing and Networking, 2019, pp. 371–376.
[48] Y. Sha, M.D. Wang, Interpretable predictions of clinical outcomes with an attention-based recurrent neural network, in: Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, 2017, pp. 233–240.
[49] D. Nie, R. Trullo, J. Lian, L. Wang, C. Petitjean, S. Ruan, Q. Wang, D. Shen, Medical image synthesis with deep convolutional adversarial networks, IEEE (Inst. Electr. Electron. Eng.) Trans. Biomed. Eng. 65 (12) (2018) 2720–2730.
[50] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, J. Sun, Generating Multi-Label Discrete Patient Records Using Generative Adversarial Networks, arXiv preprint arXiv:1703.06490.
[51] S. Bouktif, A. Fiaz, A. Ouni, M.A. Serhani, Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: comparison with machine learning approaches, Energies 11 (7) (2018) 1636.
[52] X. Zhang, R. Wang, T. Zhang, Y. Liu, Y. Zha, Short-term load forecasting using a novel deep learning framework, Energies 11 (6) (2018) 1554.
[53] S. Ryu, J. Noh, H. Kim, Deep neural network based demand side short term load forecasting, Energies 10 (1) (2017) 3.
[54] M. Kim, W. Choi, Y. Jeon, L. Liu, A hybrid neural network model for power demand forecasting, Energies 12 (5) (2019) 931.
[55] E. Mocanu, P.H. Nguyen, W.L. Kling, M. Gibescu, Unsupervised energy prediction in a smart grid context using reinforcement cross-building transfer learning, Energy Build. 116 (2016) 646–655.
[56] W. Ahmad, N. Javaid, A. Chand, S.Y.R. Shah, U. Yasin, M. Khan, A. Syeda, Electricity price forecasting in smart grid: a novel e-cnn model, in: Workshops of the International Conference on Advanced Information Networking and Applications, Springer, 2019, pp. 1132–1144.
[57] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, Y. Zhou, Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids, IEEE Transactions on Industrial Informatics 14 (4) (2017) 1606–1615.
[58] Z. Fengming, L. Shufang, G. Zhimin, W. Bo, T. Shiming, P. Mingming, Anomaly detection in smart grid based on encoder-decoder framework with recurrent neural network, J. China Univ. Posts Telecommun. 24 (6) (2017) 67–73.
[59] Y. Yuan, K. Jia, A distributed anomaly detection method of operation energy consumption using smart meter data, in: 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), IEEE, 2015, pp. 310–313.
[60] Y. Tan, W. Liu, J. Su, X. Bai, Generative adversarial networks based heterogeneous data integration and its application for intelligent power distribution and utilization, Appl. Sci. 8 (1) (2018) 93.
[61] C. Zhang, S.R. Kuppannagari, R. Kannan, V.K. Prasanna, Generative adversarial network for synthetic time series data generation in smart grids, in: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), IEEE, 2018, pp. 1–6.
[62] L. Wang, Z. Zhang, J. Xu, R. Liu, Wind turbine blade breakage monitoring with deep autoencoders, IEEE Transactions on Smart Grid 9 (4) (2016) 2824–2833.
[63] W. Zhang, W. Guo, X. Liu, Y. Liu, J. Zhou, B. Li, Q. Lu, S. Yang, Lstm-based analysis of industrial iot equipment, IEEE Access 6 (2018) 23551–23560.
[64] J. Wang, K. Wang, Y. Wang, Z. Huang, R. Xue, Deep Boltzmann machine based condition prediction for smart manufacturing, Journal of Ambient Intelligence and Humanized Computing 10 (3) (2019) 851–861.
[65] L. Li, K. Ota, M. Dong, Deep learning for smart industry: efficient manufacture inspection system with fog computing, IEEE Transactions on Industrial Informatics 14 (10) (2018) 4665–4673.
[66] Y. Yuan, G. Ma, C. Cheng, B. Zhou, H. Zhao, H.-T. Zhang, H. Ding, Artificial Intelligent Diagnosis and Monitoring in Manufacturing, arXiv preprint arXiv: 1901.02057.
[67] L. Jiang, Z. Ge, Z. Song, Semi-supervised fault classification based on dynamic sparse stacked auto-encoders model, Chemometr. Intell. Lab. Syst. 168 (2017) 72–83.
[68] H. Yan, J. Wan, C. Zhang, S. Tang, Q. Hua, Z. Wang, Industrial big data analytics for prediction of remaining useful life based on deep learning, IEEE Access 6 (2018) 17190–17197.
[69] L. Ren, Y. Sun, J. Cui, L. Zhang, Bearing remaining useful life prediction based on deep autoencoder and deep neural networks, J. Manuf. Syst. 48 (2018) 71–77.
[70] W. Tao, Z.-H. Lai, M.C. Leu, Z. Yin, Worker activity recognition in smart manufacturing using imu and semg signals with convolutional neural networks, Procedia Manufacturing 26 (2018) 1159–1166.
[71] S.-T. Park, G. Li, J.-C. Hong, A study on smart factory-based ambient intelligence context-aware intrusion detection system using machine learning, Journal of Ambient Intelligence and Humanized Computing (2018) 1–8.
[72] G.L. Grinblat, L.C. Uzal, M.G. Larese, P.M. Granitto, Deep learning for plant identification using vein morphological patterns, Comput. Electron. Agric. 127 (2016) 418–424.
[73] Y. Toda, F. Okura, et al., How convolutional neural networks diagnose plant disease, Plant Phenomics 2019 (2019) 9237136.
[74] K.P. Ferentinos, Deep learning models for plant disease detection and diagnosis, Comput. Electron. Agric. 145 (2018) 311–318.
[75] Z. Jiang, C. Liu, N. P. Hendricks, B. Ganapathysubramanian, D. J. Hayes, S. Sarkar, Predicting County Level Corn Yields Using Deep Long Short Term Memory Models, arXiv preprint arXiv:1805.12044.
[76] M. Rahnemoonfar, C. Sheppard, Deep count: fruit counting based on deep simulated learning, Sensors 17 (4) (2017) 905.
[77] H. Huang, J. Deng, Y. Lan, A. Yang, X. Deng, L. Zhang, A fully convolutional network for weed mapping of unmanned aerial vehicle (uav) imagery, PloS One 13 (4).
[78] K. Aliev, E. Pasero, M.M. Jawaid, S. Narejo, A. Pulatov, Internet of plants application for smart agriculture, Int. J. Adv. Comput. Sci. Appl. 9 (4) (2018) 421–429.
[79] C. Douarre, R. Schielein, C. Frindel, S. Gerth, D. Rousseau, Deep Learning Based Root-Soil Segmentation from X-Ray Tomography Images, bioRxiv, 2016, 071662.
[80] X. Song, G. Zhang, F. Liu, D. Li, Y. Zhao, J. Yang, Modeling spatio-temporal distribution of soil moisture by deep learning-based cellular automata model, Journal of Arid Land 8 (5) (2016) 734–748.
[81] G. Sehgal, B. Gupta, K. Paneri, K. Singh, G. Sharma, G. Shroff, Crop planning using stochastic visual optimization, in: 2017 IEEE Visualization in Data Science (VDS), IEEE, 2017, pp. 47–51.
[82] S. Goudarzi, M.N. Kama, M.H. Anisi, S.A. Soleymani, F. Doctor, Self-organizing traffic flow prediction with an optimized deep belief network for internet of vehicles, Sensors 18 (10) (2018) 3459.
[83] S. Du, T. Li, X. Gong, Z. Yu, Y. Huang, S.-J. Horng, A Hybrid Method for Traffic Flow Forecasting Using Multimodal Deep Learning, arXiv preprint arXiv: 1803.02099.
[84] X. Luo, D. Li, Y. Yang, S. Zhang, Spatiotemporal traffic flow prediction with knn and lstm, J. Adv. Transport. (2019), https://doi.org/10.1155/2019/4145353.
[85] W. Xiangxue, X. Lunhui, C. Kaixun, Data-driven short-term forecasting for urban road network traffic based on data processing and lstm-rnn, Arabian J. Sci. Eng. 44 (4) (2019) 3043–3060.
[86] J.L. Pérez, A. Gutierrez-Torre, J.L. Berral, D. Carrera, A resilient and distributed near real-time traffic forecasting application for fog computing environments, Future Generat. Comput. Syst. 87 (2018) 198–212.
[87] Y. Jia, J. Wu, M. Xu, Traffic flow prediction with rainfall impact using a deep learning method, J. Adv. Transport. (2017), https://doi.org/10.1155/2017/6575947.
[88] M. Fouladgar, M. Parchami, R. Elmasri, A. Ghaderi, Scalable deep traffic flow neural networks for urban traffic congestion prediction, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2251–2258.
[89] S. Zhang, Y. Yao, J. Hu, Y. Zhao, S. Li, J. Hu, Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks, Sensors 19 (10) (2019) 2229.
[90] X. Niu, Y. Zhu, Q. Cao, X. Zhang, W. Xie, K. Zheng, An online-traffic-prediction based route finding mechanism for smart city, Int. J. Distributed Sens. Netw. 11 (8) (2015) 970256.
[91] X. Ma, H. Yu, Y. Wang, Y. Wang, Large-scale transportation network congestion evolution prediction using deep learning theory, PloS One 10 (3).
[92] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (4) (2017) 818.
[93] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, X. Zhou, Lc-rnn: a deep learning model for traffic speed prediction, in: IJCAI, 2018, pp. 3470–3476.
[94] Z. Wang, Y. Qi, G. Zhang, L. Zhao, Smart shift decision method based on stacked autoencoders, J. Contr. Sci. Eng. (2018), https://doi.org/10.1155/2018/1098753.
[95] A. Dairi, F. Harrou, Y. Sun, M. Senouci, Obstacle detection for intelligent transportation systems using deep stacked autoencoder and *k* -nearest neighbor scheme, IEEE Sensor. J. 18 (12) (2018) 5122–5132.
[96] D. Singh, C.K. Mohan, Deep spatio-temporal representation for detection of road accidents using stacked autoencoder, IEEE Trans. Intell. Transport. Syst. 20 (3) (2018) 879–887.
[97] A. Camero, J. Toutouh, D.H. Stolfi, E. Alba, Evolutionary deep learning for car park occupancy prediction in smart cities, in: International Conference on Learning and Intelligent Optimization, Springer, 2018, pp. 386–401.
[98] M. Mohammadi, A. Al-Fuqaha, J.-S. Oh, Path planning in support of smart mobility applications using generative adversarial networks, in: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2018, pp. 878–885.
[99] X. Song, H. Kanasugi, R. Shibasaki, Deeptransport: prediction and simulation of human mobility and transportation mode at a citywide level, in: IJCAI, vol. 16, 2016, pp. 2618–2624.
[100] W. Xu, X. Feng, J. Wang, C. Luo, J. Li, Z. Ming, Energy harvesting-based smart transportation mode detection system via attention-based lstm, IEEE Access 7 (2019) 66423–66434.