Neural Computing and Applications (2021) 33:2973–3009
https://doi.org/10.1007/s00521-020-05151-8

REVIEW

Deep learning application in smart cities: recent development, taxonomy, challenges and research prospects

Amina N. Muhammad[1] · Ali M. Aseere[2] · Haruna Chiroma[3] · Habib Shah[2] · Abdulsalam Y. Gital[4] · Ibrahim Abaker Targio Hashem[3]

Received: 2 July 2019 / Accepted: 18 June 2020 / Published online: 15 July 2020

Abstract

The purpose of smart city is to enhance the optimal utilization of scarce resources and improve the resident's quality of live. The smart cities employed Internet of Things (IoT) to create a sustainable urban life. The IoT devices such as sensors, actuators, and smartphones in the smart cities generate data. The data generated from the smart cities are subjected to analytics to gain insight and discover new knowledge for improving the efficiency and effectiveness of the smart cities. Recently, the application of deep learning in smart cities has gained a tremendous attention from the research community. However, despite raise in popularity and achievements made by deep learning in solving problems in smart cities, no survey has been dedicated mainly on the application of deep learning in smart cities to show recent progress and direction for future development. To bridge this gap, this paper proposes to conduct a dedicated survey on the applications of deep learning in smart cities. In this paper, recent progress, new taxonomies, challenges and opportunities for future research opportunities on the application of deep learning in smart cities have been unveiled. The paper can provide opportunities for experts in the research community to propose a novel approach for developing the research area. On the other hand, new researchers interested in the research area can use the paper as an entry point.

Keywords Deep learning · Convolutional neural network · Smart cities · Deep belief network · Artificial neural networks · Internet of Things

1 Introduction

A new notion of accessing database and software application referred to as Internet of Things (IoT) has emerged. It is the concept in which information is gathered from sensors and software platforms to store on cloud servers. The information on the cloud servers can be accessed by human and automatic systems through software tools. The concept further gives the differences between smart cities (SC), information and communication technology (ICT) and smart grid (SG). The SC renew both the horizontal and vertical service of the systems; this is also referred to as the "system of the systems". While the ICT system is a set of point service for management and controlling applications, the SG is a system that makes all other systems to work in the SC [1].

The IoT applications in SC include smart home that enhanced personal home lifestyle, making it more convenient, easy to monitor and operate home appliances and systems. Industrial automation with minimal human involvement, the machines operations, functionalities, and productivity rates are automatically controlled and monitored. Smart healthcare with the applications of IoT is improved by embedding sensors and actuators in patients and their medicine for monitoring and tracking patients. The SG improves the energy consumption of houses and buildings which the SC makes it more convenient and easy

✉ Haruna Chiroma
chiromaharun@fcetgombe.edu.ng

[1] Department of Mathematics, Gombe State University, Gombe, Nigeria

[2] Department of Computer Science, King Khalid University, Abha, Kingdom of Saudi Arabia

[3] Future Technology Research Center, National Yunlin University of Science and Technology, Douliu, Taiwan

[4] Department of Mathematical Sciences, Abubakat Tafawa Balewa University, Bauchi, Nigeria

for the residents to obtain information of interest [2]. Zhu et al. [3] identified the key elements in IoT as identification, sensing, communication, computation, services and semantics.

The IoT devices such as sensors, actuators, and smartphones in the SC generate data as a result of activities within the SC. The data generated from the SC are subjected to analytics to gain insight and discover new knowledge for improving the efficiency and effectiveness of the SC. As such, it can be used for further development of the SC. The data analytics can be considered as the beginning of the SC. The SC data can be collected directly from variety of sensors, smart phones, citizens and integrated with city data repositories to perform analytical reasoning and to generate needed information for decision making and for better urban governance [4]. The data generated from the SC are from different domains within the SC such as transportation, energy, weather, and air pollution.

To analyze the SC data for discovering new insight, different researchers from the machine learning community applied artificial neural networks (ANN). It was found in the literature that both shallow ANN and deep learning where been applied for data analytics in SC. However, recently, deep learning has gained a serious attention from the research community because of its ability to handle large volume of dataset, less human intervention in data engineering, learn complex structure in large-scale datasets and performs better than the shallow ANN. The deep learning gains prominence as a result of its unique features in dealing with large amount of data without requiring different algorithms for dimensionality reduction, effective work on natural data, and absence of tedious human effort required in data engineering when dealing with conventional techniques in deep learning. The penetration of deep learning in SC has made remarkable achievements in different domains of the SC as listed in the preceding paragraph.

However, despite the popularity and achievements made by deep learning in SC, no survey has been dedicated mainly on deep learning for ANN in SC to show recent progress and direction for future development of the research area. In this paper, we propose to present recent development, challenges and future research direction on the applications of deep learning for ANN in SC. The shallow ANN will be considered in the paper to show its relevance in the SC but emphasis is on the deep learning.

The contributions of the paper can be summarized as follows:

- The literature review presents a comprehensive work on deep learning solutions in SC extracted from various projects trending in the literature.
- The survey has organized the different deep learning solutions in SC projects into different architecture of the deep learning for easy identification of the type of architectures already applied for SC project. The projects are organized based on the following deep learning architectures: Convolutional Neural Network; Deep Belief Network; Deep Reinforcement Learning; Recurrent Neural Network.
- The literature review created a new taxonomy based on the deep learning solutions in SC depicting the main deep learning architectures and their variants.
- A taxonomy on the domain of the SC that the deep learning algorithms are found to be applied is created for researchers to see the pictorial view of the SC domains already have deep learning solutions.
- We depicted publications trend on the research conducted on the deep learning solutions in SC. The trend clearly indicated that the research area is attracting unprecedented attention from both the industry and the academia. This is because the number of publications in the last 3 years, 2017, 2018 and 2019 has increased as the number of publications within these 3 years exceeded the publications that pre-date 2017.
- The sources of the datasets, deep learning frameworks and hardware used in different projects were extracted and presented in a tabular form to make the literature review a self-contained for new researchers intending to start work in this research area.
- A discussion on the challenges militating against the future development of the research area is outlined in the literature review. The major challenge identified remains the known open problem of lack of systematic procedure for estimating the proper parameters of the deep learning algorithms. However, efforts made by researchers in automatic optimization of the deep learning parameters through meta-heuristic nature inspired algorithms are not applied in this domain to investigate its efficiency and effectiveness in SC. Secondly, the data generated, acquired from internet of vehicles within a SC and analyze based on deep learning architecture is scarce.
- Directions for future research opportunities for solving the identified challenges are outlined for easy identification by researchers in proposing a novel and robust deep learning solutions in SC.

Other sections of the manuscript are as follows: The methodology of the literature review is presented in Sect. 2. Section 3 presents the background of deep learning for ANN and the taxonomy. Section 4 presents the concept of the SC including its layers and architecture. Section 5 highlights the applications of the deep learning for ANN in SC. Section 6 presents the deep learning platforms.

Section 7 highlights dataset from different projects. Section 8 presents general discussion, domains of application and trend of publications. Section 9 discusses the challenges and future research work while the paper is concluded in Sect. 10.

2 Methodology

The methodology explains how the literature review was conducted. The main stages involve in the methodology are as follows: formulation of keywords used, article identification from the academic databases, screening, eligibility, inclusion and exclusion criteria. The keywords used for the search were formulated and refined before using those keywords to search the databases. Example of the keywords includes *smart city, deep reinforcement learning*, *passenger, convolutional neural network*, *air pollution, deep belief network*, *parking, convolutional neural network*, *traffic, neural network*, *vehicle detection, convolutional neural network *, *parking, deep recurrent neural network*, etc. The keywords were used to search the following relevant academic databases ScienceDirect, ACM Digital Library, Springerlink, IEEE Explore, Scopus, DBLP, and Web of Science. Articles were retrieved from the databases, we excluded non-English-language articles, non-peer-reviewed articles such as short communication, editorial, keynotes, book, descriptions and technical reports. The articles considered for inclusion were peer-reviewed from reputable journals, conferences or edited book written in English and report empirical works on deep learning solutions to solve problem in SC. The articles were screened based on titles, abstract, duplicate, conclusion and some of the articles by comprehensively going through the full content of the article. The full texts of the articles were assessed for eligibility to be included for the quantitative analysis.

3 Deep learning background

This section presents background information about deep learning including taxonomy, motivation, benefits, different architectures of deep learning algorithm, discussion on hyperparameters and lastly the benefits of deep learning and its advantage over traditional approaches in the context of SC are discussed.

3.1 Taxonomy, motivation and benefits of deep learning

Figure 1 presents a comprehensive taxonomy of the deep learning architecture and the shallow ANN. The taxonomy is created based on the deep learning and shallow ANN found to be applied for data analytics in the context of SC. The basics of the deep learning for ANN basic concept on how they operate to achieve their goals are provided in the section for interested readers to understand the algorithms operates. The deep learning architecture and other shallow ANN discussed in this section are found to be applied for data analytics in SC environment. The differences between the shallow ANN and the deep learning including advantages and limitations are presented in the section.

The shallow machine learning algorithms have been adjudged to work successfully in solving variety of problems. The shallow ANN architectures prove to perform good in many of the common machine learning problems. The shallow ANN are still active in the majority of the machine learning applications today [5]. However, the shallow machine learning algorithms have not recorded significant success in solving core artificial intelligence problems such as the speech and object recognition [6]. The shallow ANN process raw data in a limited way because certain level of domain expertise is required when constructing pattern recognition system or machine learning system. In addition, the shallow ANN require hand engineering for feature extraction in transforming the raw data to feature vectors before feeding the data into the shallow ANN algorithm for solving classification or pattern recognition problems. On the other hand, deep learning architecture does not require extra feature extraction step before feeding the data to the deep learning architecture, thereby eliminating the extra feature extraction stage required by the shallow ANN. The features are automatically extracted by the deep learning algorithms, and it is done in much better way than the hand-coding feature extraction [5].

The shallow machine learning algorithms relied heavily on the prior beliefs such as the smoothness prior to generalize well. The shallow algorithm typically fails in mitigating statistical challenges that is involved in solving artificial intelligence task. On the other hand, the deep learning introduces explicit and implicit priors to improve the generalization ability when dealing with complex artificial intelligence task. Curse of dimensionality: In terms of dataset, the shallow ANN works well on small dataset, as the amount of the data increases the performance of the shallow ANN decreases. However, the deep learning algorithms perform very well when the dataset is large, as the amount of the data increases the performance of the deep learning algorithm increases [6]. The deep learning algorithms do not perform well on small amount of datasets. Table 1 presents the summary of the strength and weaknesses of the shallow ANN and the deep learning.

It is well known in the research community that the parameters of the ANN remain an open issue as it lacks systematic procedure to get the optimum parameters. Typically,

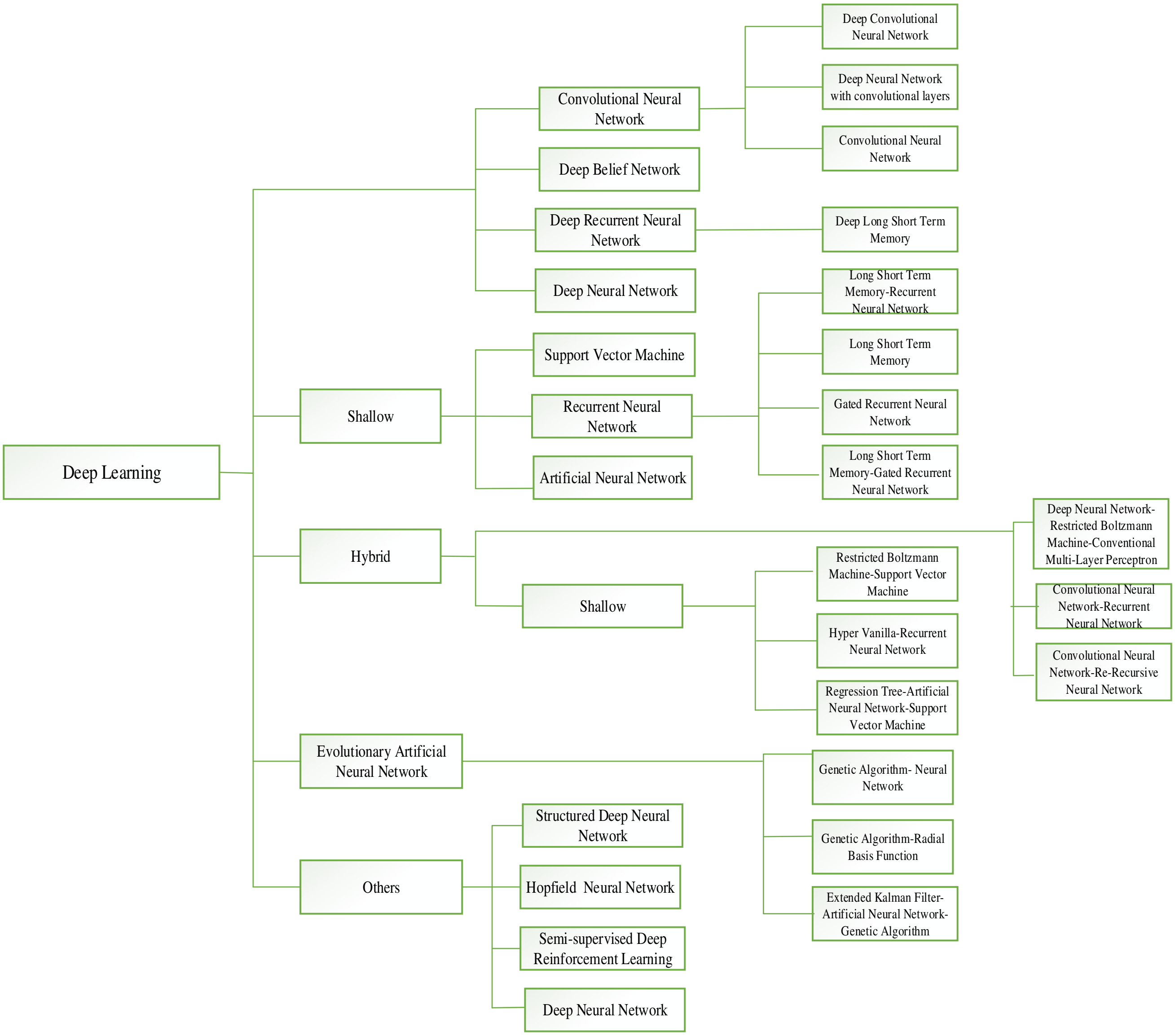
Deep Learning
Convolutional Neural Network
Deep Belief Network
Deep Recurrent Neural Network
Deep Neural Network
Deep Convolutional Neural Network
Deep Neural Network with convolutional layers
Convolutional Neural Network
Deep Long Short Term Memory
Shallow
Support Vector Machine
Recurrent Neural Network
Artificial Neural Network
Long Short Term Memory-Recurrent Neural Network
Long Short Term Memory
Gated Recurrent Neural Network
Long Short Term Memory-Gated Recurrent Neural Network
Hybrid
Shallow
Deep Neural Network-Restricted Boltzmann Machine-Conventional Multi-Layer Perceptron
Convolutional Neural Network-Recurrent Neural Network
Convolutional Neural Network-Re-Recursive Neural Network
Restricted Boltzmann Machine-Support Vector Machine
Hyper Vanilla-Recurrent Neural Network
Regression Tree-Artificial Neural Network-Support Vector Machine
Evolutionary Artificial Neural Network
Genetic Algorithm- Neural Network
Genetic Algorithm-Radial Basis Function
Extended Kalman Filter-Artificial Neural Network-Genetic Algorithm
Others
Structured Deep Neural Network
Hopfield Neural Network
Semi-supervised Deep Reinforcement Learning
Deep Neural Network

Fig. 1 Taxonomy of the deep learning for ANN in smart cities

Table 1 The summary of the strength and limitations of the deep learning and shallow ANN

	Advantages	Limitations
Deep learning architecture	1. Automatic feature extraction [7]	1. Perform poorly on small dataset [6]
	2. Process massive amount of data efficiently and effectively [8]	2. Require high-performance computing system [8]
	3. Very good in speech [9] and object recognition [10, 11]	3. The training takes long time and require many parameter settings [12]
Shallow ANN	1. Perform well on small amount of data set [6]	1. Its performance decreases as the amount of data increases [6]
	2. Ability to learn complex relationship [13]	2. Require hand engineering feature extraction [14]
	3. Very fast in processing small amount of data [15]	3. Weak in speech recognition and object recognition [6]

the ANN parameters are optimised at the training phase. The performance of the ANN heavily depends on the parameter settings, architecture of the ANN, choice of training algorithm and training features. However, several ways are available to get the optimum estimated parameters of the ANN. First, preliminary experimentation: in this case, preliminary experiment is conducted with different ANN architecture, training algorithm, parameter setting and training features to find out the optimum ANN. After the training, the performance of the ANNs models is validated by comparing the error function of the ANNs architecture and the architecture with the lowest error function is selected showing that the selected ANN has the best estimated parameters. Subsequently, the ANN with the best estimated ANN parameters is deployed in the large-scale experiment. Secondly, application of nature inspired meta-heuristic algorithms, such as the genetic algorithm, differential evolution, artificial bee colony, and particle swarm optimization, can be applied to train the ANN in search of the best estimated ANN parameters, such as weights, number of hidden neurons, and bias. Such type of algorithms starts by generating population of ANN as the initial solutions; the ANNs models generated compete among themselves. The ANN with the optimum objective function survive to the end and are selected as the ANN with the best estimated parameters.

3.2 Convolutional neural network

The convolutional neural network (ConvNet) as shown in Fig. 2 involves a mathematical operation named 'convolution'. The convolution refers to operation that consists of many applications of convolution in parallel. It is a special type of linear operation that uses convolution instead of general matrix multiplication in its layers. The ConvNet describes the first variable as the input, and the second variable as the kernel and the output is termed as the feature map.

The input is the multidimensional array of data and the kernel as the multidimensional array of parameters. These two multidimensional arrays are referred to as tensors; this is because both the elements of input and kernel must be kept differently. The key motivation of ConvNet has three essential elements: sparse interactions, parameter sharing and equivariant representation. The sparse interaction makes the kernel smaller than the input, parameter sharing utilized in a model. The equivariant representations mean if an input changes, an output will also change in the same way [6].

The ConvNet components have two layers, the left and the right. The left layer is seen as the small number of complex layers in which each layer is having several stages. These layers can also be considered as the input layer, output layer and several hidden layers in which the hidden layers are further categorized into convolutional, pooling, fully connected and normalization. A pooling layer contains three steps: The first step executes many convolutions one after the other to give out a set of linear activations. In the second stage, each linear activation is executed via a nonlinear activation function (also refer to detector layer). The third step uses the pooling function to change the output of the next layer. The pooling improves the computational and statistical efficiency of the network and reduces the amount of storage to keep the parameters because of its ability to use the fewer pooling units than the detector units. The ConvNet has "receptive unit" for the insertion space in a neuron. Regularization changes the algorithm to reduce only the generalization error without the training error to solve the problem of underfitting and overfitting [6]. Each of the neurons in the feature maps of the ConvNet is connected to the receptive unit in the preceding layer. The subsequent new feature map is produce by the convolution of the input vectors with the learning kernel. The convolved result obtained is mixed with the activation function—nonlinear. The kernel is shared by all the spatial location of the input vectors to generate each of the feature maps. Different kernels are applied to get the complete feature maps. The feature value of each location in the feature map is computed as follows:

$$z^l_{i,j,k} = w^{l\,T}_k x^l_{i,j} + b^l_k$$

where (i,j) is the feature value at location i and j in the kth feature map of the ith layer. The feature map value is $z^l_{i,j,k}$. The w^l_k is the weight vector and b^l_k represent the bias term of the filter on the layer. The input patch center location is $x^l_{i,j}$ at the locations i and j. The w^l_k generate the feature map to be shared. The mechanism that shares the weights typically reduces the complexity of the ConvNet and fasten the training speed of the ConvNet. The nonlinearity of the ConvNet is introduced by the activation function to detect nonlinear features. The activation of the convolutional feature can be expressed as:

$$a^l_{i,j,k} = a\left(z^l_{i,j,k}\right)$$

$$a^l_{i,j,k} = a\left(w^{l\,T}_k x^l_{i,j} + b^l_k\right)$$

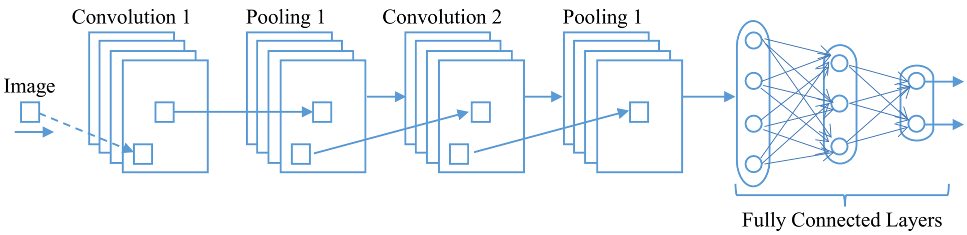

Fig. 2 A typical convolutional neural network

where $a^l_{i,j,k}$ is the activation value of the convolution feature. The activation function is typically sigmoid, tanh and ReLu. The softmax is mostly used at the output layer to solve classification problem. The pooling achieves the shift variance and is always located between two convolution layers.

The pooling function for each of the feature map is computed as:

$$y^l_{i,j,k} = \text{pool}\left(a^l_{m,n.k}\right), \forall(m,n) \in \Re_{ij}$$

The local neighborhood is $\Re_{ij}$ around the location of (i,j). Several pooling exists but the commonly used pooling in the literature is max pooling and average pooling. The kernel in the first level of the convolution detects the edges and curves which are low features. The higher-level features are extracted by the stack of convolution layers and pooling layers.

The loss function of the ConvNet can be computed as follows:

$$\tau = \frac{1}{N}\sum^{N} t\left(\theta; y^{(n)}, o^{(n)}\right)$$

where N is the input and output desired relationship $\left\{\left(x^{(n)}, y^{(n)}\right); n \in [1,\ldots,N]\right\}$, $x^{(n)}$ represent the input sequence and the corresponding target label is $y^{(n)}$, θ is the parameters of the network and lastly, $o^{(n)}$ is the output of the ConvNet. The best of the θ is obtained by the minimizing the lost function through training [16].

3.3 Recurrent neural network

The recurrent neural network (RNN) is the most general and efficient network that presents sequence of learning method [17]. The RNNs are dynamical systems with temporal state structures. They are relatively strong and play a vital role in various temporal processing scenarios and digital applications. The popular RNN model is the Hopfield model which stores information in a dynamical standard feature and the Cohen–Grossberg model used as an associative memory for keeping information and finding way out for optimization problems. The RNN is categorized into global and local. The global RNN agreed feed forward connection between each neuron, while local RNN strict feed forward connection which structured with dynamic neuron models. Two static time models exist for RNN such as the time-delayed and the simultaneous RNNs, the first one is trained to reduce prediction error and the second is learnt to produce general function approximation abilities [4]. The deep RNN comprises of multiple hidden layers as shown in Fig. 3 unlike the shallow RNN.

The RNN is a dynamic system that has input and output; the dynamic system can be expressed as:

$$h_t = f_h(x_t, h_{t-1})$$

$$y_t = f_o(h_t)$$

where $\mathbf{x}_t$ is the input, $\mathbf{y}_t$ is the output and $\mathbf{h}_t$ represent the hidden state. The f_h is the output function parameterized by $\boldsymbol{\theta}_h$ and lastly f_o is the state transition function parameterized by the parameter $\boldsymbol{\theta}_o$.

The data set N for the training

$$D = \left\{\left(X_1^{(n)}, y_1^{(n)}\right), \ldots, \left(X_{Tn}^{(n)}, y_{Tn}^{(n)}\right)\right\}_{n=1}^{N}.$$

Optimizing the RNN parameters can be performed by minimizing the objective function expressed as:

$$J(\theta)\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{Tn} d\left(y_t^{(n)}, f_0\left(h_t^{(n)}\right)\right)$$

where the predefined divergence between $\boldsymbol{a}$ and $\boldsymbol{b}$ is given as follows:

$$h_t^{(n)} = f_h\left(x_t^{(n)}, h_{t-1}^{(n)}\right) \text{ and } h_o^{(n)} = 0.\ d(a,b)$$

The construction of the RNN is accomplished by the following transition and output functions:

$$h_t = f_h(x_t, h_{t-1}) = \phi_h\left(W^T h_{t-1} + U^T x_t\right)$$

$$y_t = f_o(h_t, x_t) = \phi_o\left(V^T h_t\right)$$

where $\mathbf{W}$ is the transition, $\mathbf{U}$ is the input and $\mathbf{V}$ represents the output matrices. The ϕ_h and ϕ_o represent the nonlinear functions called the element-wise. The hidden layer function is typically sigmoid [18].

3.4 Deep belief network

A deep belief network (DBN) as shown in Fig. 4 can be built as a stack of Boltzmann machine (RBM) in which the hidden layer of the RBM in the stack is the input to the

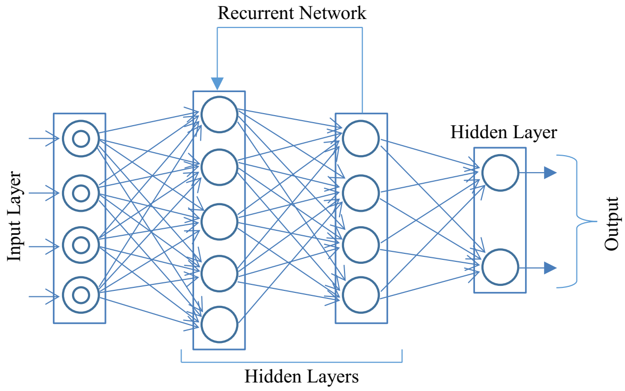

Fig. 3 Deep recurrent neural network

visible layer of the subsequent RBM in the stack. Training of these RBMs occurs one after the other beginning with the initial RBM in the stack. When the training is done for the initial RBM in the stack, features of the input are produced at the hidden layer. These features send on to the next RBM in the stack and the RBM is trained. This process continues until it reaches the last RBM in the stack. The training of each RBM is achieved with the aid of positive and negative phases; the DBN picks up the features of the original data in its layers of the network. A DBN is a directed acyclic graph comprises of stochastic variables. Features of the DBN include simplicity, biologically plausible form, and it can find solutions to various problems [19].

The RBM is applied for the training of the DBN. The RBM is a pre-training technique for training each layer of the DBN. The visible and hidden layers of the RBM carry only two values which is 0’s and 1’s. For example, if the visible unit is visible while the hidden unit is hidden, then the visible = [20] and the hidden = [20]. The computational energy in the visible and hidden nodes is expressed as follows:

$$E(vis, hid; \theta) = -\sum_{i=1}^{D} b_i vis_i - \sum_{j=1}^{F} a_j hid_j - \sum_{i=1}^{D}\sum_{i=1}^{F} w_i vis_i hid_j$$

From the equation we can deduce that the w_{ij} is the weight that connected the visible node to the hidden node for the vis_i and hid_j. The bias for the visible nodes and hidden nodes are represented by b_i and a_j, respectively. The join distribution of the network is expressed as:

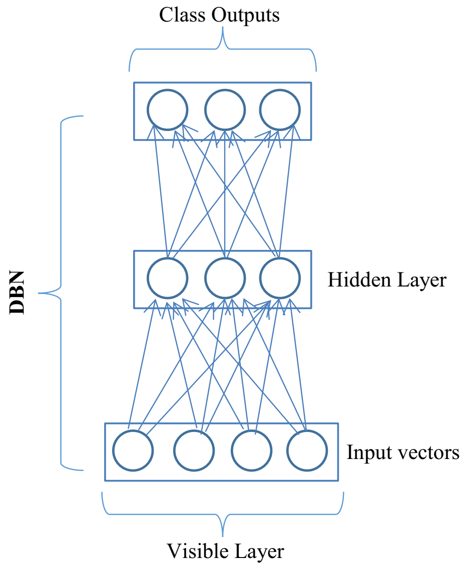

Fig. 4 Structure of deep belief network

$$p(vis, hid; \theta) = \frac{1}{Z(\theta)} \exp(-E(vis, hid; \theta))$$

$$Z(\theta) = \sum_{v}\sum_{h} E(vis, hid; \theta)$$

$$p(vis, hid; \theta) = \frac{1}{\sum_v \sum_h E(vis, hid; \theta)} \exp(-E(vis, hid; \theta))$$

The normalized constant is $Z(\theta)$. The function of the energy can be used to get probability on the input vector. The energy is minimize by improving the probability and θ. The hidden and visible nodes conditional distribution can be expressed by the logistic function as follows:

$$p\left(hid_j = 1 | vis\right) = g\left(\sum_{i=1}^{D} W_{ij} vis_i + a_j\right)$$

$$p\left(vis_j = 1 | hid\right) = g\left(\sum_{i=1}^{D} W_{ij} hid_i + b_j\right)$$

$$g(x) = \frac{1}{1 + \exp}$$

Immediately the hidden nodes have been chosen; the input vector is reconstructed by setting each of the vis_i to 1. Subsequently, the nodes in the hidden states are updated which represent the reconstruction of the network features. The network is trained using contrastive divergence techniques and the weight also changed [21].

3.5 Deep reinforcement learning

The reinforcement learning operates in environment to take action with the goal of maximizing rewards and enhance the efficiency of the learning algorithm [22]. Figure 5 is the block diagram of the reinforcement learning.

In the reinforcement learning environment, agent interacts with the environment ε based on some number of discrete time stages. In each of the time stage t, state s_t is received by the agent. Subsequently, action a_t is selected from the available possible actions A based on the policy π of the A. The π is maps from the s_t to a_t. In the next stage, the agent received the subsequent s_{t+1} and scalar reward r_t. The agent continues the process to the point of termination state before the process restarts again. The total accumulated return from t with $\gamma \in (0,\ 1]$ as the discount factor has a return $R_t = \sum_{k=0}^{\infty} \gamma^k r\, t + k$. The agent goal is to maximize the expected returns from each of the s_t.

The $Q^{\pi}(s,\ a) = E[R_t | s_t = s,\ a]$ is the action value which is the expected return from the selected action in a state following a π. The $Q^{\pi}(s,\ a) = \max_{\pi}(s,\ a)$ is the value of the action that gives maximum action value for s and a which can be achieved by any of the π. The value of s

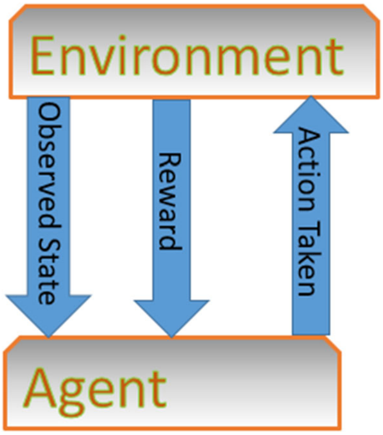

Fig. 5 Reinforcement learning

under π can be defined as $V^{\pi}(s) = E[R_r|s_t = s]$ representing the expected return for following π from s.

In the value-based free reinforcement learning approach, function approximator such as ANN is used to represent the action value function.

The approximation action value function (Q) with the parameter θ is given as $Q(s\,a;\,\theta)$, variety of reinforcement learning algorithms such as the Q-learning is used to update the θ, and the Q-learning target to directly approximate the optimal AVF is expressed as:

$$Q^{*}(s,a) \approx Q(s,a;\theta)$$

In a single-step Q-learning, the θ of the $Q(s,\,a;\,\theta)$ are learned by the iterative minimization of a loss functions expressed as:

$$L_i(\theta_i) = E(r + \gamma \max_{a'} Q(s',\,a';\,\theta_{i-1}) - Q(s,\,a;\,\theta_i))^2,$$

after the s, s' is encountered. This is a one-step Q-learning [23]. The combination of the principles of reinforcement learning and deep neural network creates a powerful algorithm referred to as the deep reinforcement learning (DeepRL) [24].

3.6 Artificial neural network

In an ANN, the processing elements are organized in layers. Neurons in one layer accept signals from neurons in the next layers and send signals to neurons in the next layer, no connection between neurons in the same layer. The total input to each node is the weighted sum of outputs of the nodes in the previous layer. Each node is activated based on the input to the node and the activation of the node [4]. The backpropagation (BP) [25] is an optimization technique commonly used for the training of ANN parameters for better performance of the ANN. The BP is a supervised learning technique that uses a gradient descent in an error variable. The error is initiated by looking at the similarity between output values to a target value [26]. The BP can be used to explain the training of multilayer perceptron with the help of gradient descent evaluating the derivatives of the error function with respect to the weights in the network using good computational technique [19]. The BP is the most popular gradient-descent based algorithm which is derived from the entropy criteria to minimize the training time, and decease the problem of getting local minima by reducing the density of local minima [27]. Figure 6 depicts the typical structure of an ANN.

The BP is a gradient descent algorithm that minimizes the objective function that is the error function. Let's take the following as the training data: $(I_1, T_1), \ldots, (I_n, T_n)$, in which the I_s, $1 \le s \le n$ where I_s is the input and the target is T_s, $1 \le s \le n$, for neurons at the hidden layer, the objective function which is the least square error in the ANN weight is expressed as:

$$E = \frac{1}{nZ_M}\sum_{s=1}^{n}\left[T_s - O_s^M\right]\left[T_s - O_s^M\right]$$

where Z_M and O_s^M represent number of neurons on the output layer and the output vector of the ANN with I_s, respectively. Assuming that w is the vector that is formed by all the weights in the ANN and the gradient $\nabla E(w(k))$ of the E at $w = w(k)$ where $k = 1, 2, 3, \ldots, n$ represents the number of iterations of the ANN weight.

The ANN with a momentum can be expressed as:

$$\nabla w(k) = \alpha(-\nabla E(w(k))) + \beta \nabla w(k-1)$$

The learning rate and the momentum factor are represented by α and β, respectively. The $\nabla w(k) = w(k+1) - w(k)$.

The computation of the ANN with the input vector feed to the input layer is expressed as:

$$o_{s,i}^{m} = f\left(\left[w_i^{m}(k+1)\right]^{\tau} O_s^{m-1}\right)$$

where $o_{s,i}^{m}$ is the ith output of the ANN layer m, $1 \le m \le M$. The activation function is represented as $f(\cdot)$ and the sub-

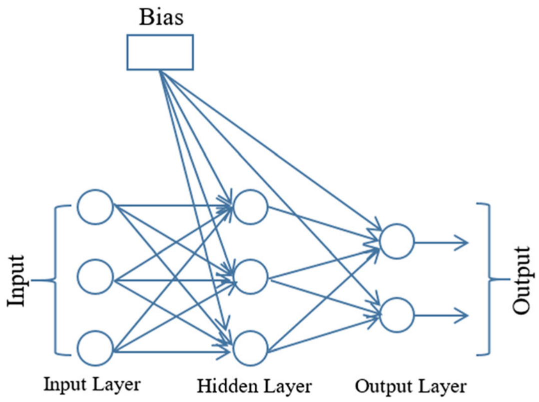

Fig. 6 Structure of artificial neural network

vector of $w(k+1)$ is $w_i^m(k+1)$ comprising all weights from the neurons of the $m-1$ layer to $o_{s,i}^m$ and the vector formed by all the output layer of the layer $m-1$ is $o_{s,i}^{m-1}$ including the unity output that is used as the reference to the ANN bias of the subsequent layer in the ANN and is expressed as [28]:

$$o_{s,i}^{m-1} = \begin{cases} \left[1 \quad O_{s,1}^{m-1} \ldots O_{s,z_m}^{m-1}\right]^{\tau} & \text{for } m > 1, \\ \left[1 \quad I_s^{\tau}\right]^{\tau} & \text{for } m = 1. \end{cases}$$

The deep ANN (D-ANN) architecture is unlike the ANN architecture in Fig. 6 with a single hidden layer. The deep neural network has multiple hidden layers, at least 3 or more than 2.

3.7 Support vector machine

The support vector machine (SVM) is shown in Fig. 7; it implements the structural risk minimization procedure which reduces the upper bound of the generalization error. The learning process for building the SVM is fast; its parameters can be handled in different case with the unique solution that reduces the empirical risk and is immune to local minima. The major concept of SVM is to convert the training set in a high-dimensional space into a lower-dimensional feature space by means of a set of nonlinear kernel functions, where the conversions of the training sets are always in the feature space [4]. The SVM is basically a two classifier which classifies the data from the class as positive and groups the other data from the class as negative which is called one-versus-the-rest approach [29].

The SVM is a decision machine which solves the problems of classification, clustering, regression using sinusoidal data set, and finds the solution for an unsupervised learning problem related to probability density estimation. The amount of root functions in the models is relatively smaller compared to the training points, as well as adding the size of the training set which yields probabilistic results [30]. The description of the SVM starts with the two linear separable classes. To illustrate the operation of the SVM, let us assume that we have a data set $D = \{(X_i, y_i)\}_{i=1}^{\ell}$ where $y_i \in$ [31]. Suppose that we are to determine the number of the classifiers that separate the data set to determine the one that will produce the lowest generalization error. The hyperplane is a good choice intuitively; it can be able to leave a maximum margin between the classes. In the case the two classes are none separable, the hyperplane that maximizes and minimizes margin and misclassification error, respectively, can be considered. The positive constant C controls the balance between the margin and the misclassification error. The value of the positive constant C is chosen before the start of running the SVM. Therefore, it can be concluded that solution to this problem is a linear classifier expressed as:

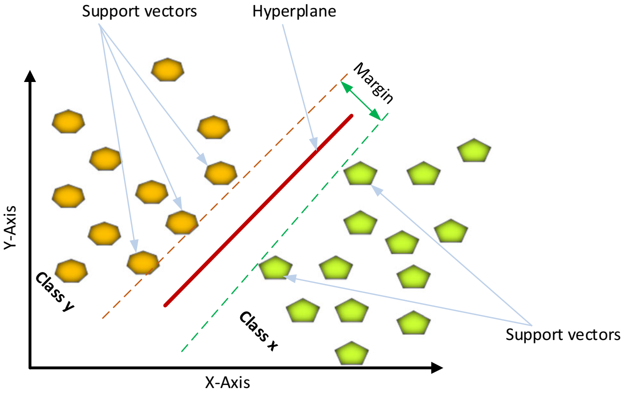

Fig. 7 A sample of support vector machine

$$f(x) = sig\left(\sum_{\ell=1}^{\ell} \lambda_i y_i X^T X_i + b\right)$$

where λ_i is the coefficients which are the solutions to the quadratic programming problems:

$$\text{Minimize} \quad W(\Lambda) = -\Lambda^T 1 + \frac{1}{2}\Lambda^T D \Lambda$$

$$\text{Subject to} \quad \begin{aligned} \Lambda^T y &= 0 \\ \Lambda - C1 &\le 0 \\ -\Lambda &\le 0 \end{aligned}$$

where $(\Lambda)_i = \lambda_i, (1) = 1$ and $D_{ij} = y_i y_j X_i^T X_j$.

Only a small fraction of the coefficients λ_i are different from 0. The solution to the problem is determined by the data point associated with the coefficients that are none 0 in view of the fact that every coefficient corresponds to a certain data point. The data points in questions are referred to as the support vectors. These support vectors are the only once relevant to the problem solutions while other data point can be discarded from the set of the data. The number of the support vectors is typically small in number and those support vectors are normally at the border between the classes. The kernel functions play role on the performance of the SVM and its choice is not straight forward. The famous among the kernel functions includes Gaussian RBF, polynomials, and multi-layer perceptron [32].

3.8 Estimation of deep learning architecture hyperparameters

Hyperparameters of the deep learning architecture, regularization and optimization are sacrosanct in determining the performance of the deep learning algorithms, meaning that the performance of the deep learning algorithm is sensitive to hyperparameters setting. The performance of deep learning hyperparameters varies across different datasets. In practice, hyperparameters optimization is a challenging task which makes it a difficult problem as follows [33]:

1. Large models such as deep learning make function evaluation to be extremely expensive.
2. It frequently has a complex configuration space.
3. With limited size of data, direct optimization for generalized performance cannot be achieved.
4. Convexity, smoothness, etc., of the target function used in the classical optimization are not usually applied.

Bergstra et al. [34] reported that in the domain of hyperparameters, a hyperparameter can take different data type. For example, learning rate is real number, number of layers is integer value, training stoppage criterion or the choice of optimizer is binary or categorical.

The systematic approach for determining the exact hyperparameters of the deep learning algorithms remains an open research problem. Therefore, a number of approaches for estimating the deep learning algorithms hyperparameters exist. Firs, manual method: many of the human experts still depend on manual procedure of determining model hyperparameters [35]. In manual approach, preliminary experiments are conducted with different hyperparameters settings of the deep learning algorithm using a small subset of the dataset to estimate the best hyperparameters setting. This is because increasing data size increases the complexity of the model which makes the optimization of the hyperparameters very difficult in large data size [33]. In manual method of setting hyperparameters values, the running of the model is terminated prematurely if it is expected to produce poor output so as to save time [35]. For example, ConvNet requires the following parameters: learning rate, momentum, activation function, pooling size, convolution, drop out, number of dense layers, etc., to be set for running the ConvNet to solve machine learning problems. So during the preliminary experiment, different settings of the hyperparameters will be tried on small subset of the dataset and the settings that produce the best result are adopted for use in the main experiment. The manual method is tedious, expensive and time-consuming. In some cases, the hyperparameters of the deep learning algorithms are estimated during conducting the main experiment without preliminary experiments but is highly challenging.

Secondly, the application of global optimization algorithms are cuckoo search algorithm, artificial bee colony, GA, firefly algorithms to automatically estimate the hyperparameters of the deep learning algorithms. In this approach, any of the suitable global optimization algorithm can be applied to automatically determine the best estimated hyperparameters of the deep learning algorithms. In this approach human intervention in estimating the hyperparameters settings is highly minimal. For example, Baldominos et al. [36] applied GA for automatic estimation of the complete architecture of the ConvNet: number of layers, connectivity, batch size, learning rate, learning rule, dropout, weight regularization, activation function, number of neurons on each layer, number of convolution layer, number of kernels on each convolution, pooling size, etc. Huang et al. [37] applied PSO to estimate the hyperparameters of DBN by optimizing the number of input neurons, number of hidden neurons and learning rate of the DBN. Liu et al.[38] used GA for the optimization of DBN hyperparameters. Similarly, Papa et al. [39] applied Harmony Search algorithm for the estimation of DBN hyperparameters of a Discriminative Restricted Boltzmann Machines.

Thirdly, deep learning hyperparameters can be adopted from a study that solves relatively similar problem or hyperparameters suggested in the literature. However, it is not necessary that the adopted estimated hyperparameters settings will produce the desired output in solving another similar problem. The default settings of the hyperparameters provided by the deep learning library can be adopted for training the deep learning model. However, the tuned hyperparameters are mostly found to be better than the default settings provided by the machine learning libraries [33].

Fourthly, Grid search or full factorial hyperparameter optimization, the modeler or user supplied pre-define sets of values for each of the algorithm hyperparameter before the grid search evaluate the Cartesian product of the defined set of the values supplied by the modeler or user. This approach has limitation arising from the dimensionality curse in view of the fact that the number of function required to be evaluated increases exponentially with the dimension. Fifth, random search, in this approach of hyperparameter optimization, the configuration of the hyperparameters are randomly search continuously until the budget of the search is exhausted. It works well in a situation where some hyperparameters are more important than others [33].

Sixth, Bayesian optimization is an optimization framework used for the optimization of deep learning hyperparameters. It gains attention in recent times in the tuning of deep learning architecture for image processing, speech recognition and natural language processing [33]. Type of Bayesian optimization includes: sequential model-based Bayesian optimization [40], surrogate models, configuration space and constrained Bayesian optimization [33]. Ilievski et al. [41] propose deterministic surrogate with radial basis function as the surrogate for deep learning hyperparameter optimization that requires the evaluation of few functions for the hyperparameters optimization.

3.9 Motivation and advantages of deep learning in smart city over traditional techniques

The motivation and benefits of the deep learning over traditional approaches are presented in Sect. 3.1. In the context of SC, deep learning has recently gained popularity in SC applications due to its ability to handle large-scale amount of datasets with high accuracy compared to the traditional techniques. The data generated from SC is complex in nature and very large because it comes from different devices which can be images, videos, text and speech. The data comes in different forms such as structured, semi-structured and unstructured. Therefore, it can suitably be analyze using the deep learning depending on the data analytic problem intended to solve in the SC. The deep learning can be applied to any of the following data analytics: classification, clustering and regression. Object recognition and speech recognition are task that can be accomplished in the SC using the deep learning architecture. As already discussed in Sect. 3.1, these categories of data can be processed better with deep learning than the traditional techniques. For example, Amato et al. [42] used ConvNet for the classification of images in parking space for smart cameras. It was found that the ConvNet outperforms the traditional technique. Similar study by Valipour et al. [43] shows that the ConvNet performs better than the traditional ANN. Mynhoff et al. [44] present comparative analysis between deep learning methods and traditional techniques: Hidden Markov, SVM and Factored Hidden Markov to predict electrical energy consumption in SC environment. The analysis of the day-ahead and week-ahead energy prediction demonstrates that different prediction methods present significantly different levels of accuracy, with the DBN offering the most consistent performance over the compared traditional techniques. Similarly, recurrent neural networks have shown promising performance in natural language processing and language modeling tasks compared to traditional machine learning algorithms [45]. Van der Veen et al. [46] used a 3D ConvNet validation to semi-automate delineation of OARs with respect to delineation accuracy. The result shows that the automated ConvNet OAR delineation in head and neck cancer was shown to be more efficient and consistent compared to manual delineation.

The deep learning has the capability to effectively recognize objects and speech within the SC environment. The data acquired from the SC means nothing if the data is not processed to uncover important new information in the acquired data been generated from different devices in the SC. That is where the deep learning plays a critical role in utilizing the data for solving classification, regression and clustering problems. Since the deep learning is a machine learning algorithm, it has to learn from the data set to infer from it. So, if the deep learning is to be used in the SC for any of the problems listed previously, the data has to be acquired in line with the intended problem to solve.

The domain of application in the SC also differs because the deep learning need data from the domain it intended to solve the problem. For example, the deep learning can perform passenger prediction only if it is trained with the data set from the transportation domain of the SC. Similarly, applying deep learning for solving problem in energy, rail system, air quality, traffic flow, sewage, accident, health, agriculture will definitely require different data set depending on the domain of application. The data analytics in the SC will help the decision makers to fully understand the effective and efficient ways for proper planning, sharing of resources, formulation of regulatory framework, taking of critical decision and monitoring in the SC.

4 The smart city initiative and architecture

The SC was referred to a digital city, but later, it was gradually replaced by SC [1]. A digital city refers to a connected community that gathers broadband communications service, a pliable, service-oriented computing service based on open industry standards. It also encompassed innovative services to comply with the needs of Governments' employees, citizens and businesses [47]. Figure 8 shows the layers of the SC in which several technologies are connected to illustrate the relationship between various sensor devices, wireless sensor networks and the city infrastructure. The first layer introduced the interactions among various city infrastructure and sensors devices. This interaction demonstrates how sensors are embedded within the city. The second layer illustrates how sensor networks, internet, and cloud services are used for smart applications.

With the rapid development of science and technology, the world is suiting "smart". Living in such a smart world [3], people will automatically work together with the smart devices (e.g., watches, mobile phones, etc.), computers, smart transportation (e.g. cars, buses, trains, smart environments e.g. homes, offices, factories etc.). The key notion of the SC is to find the right information at the exact place and on the valid device to make the city connected with simplicity [48]. A SC function as an ICT to lift its survivability, effort and sustainability [1]. The city takes care of the conditions of the basic facilities and services such as roads, bridges, subways, tunnels, communication, airports, seaways including building to improve its resources and security. Giffinger and Pichler-Milanović

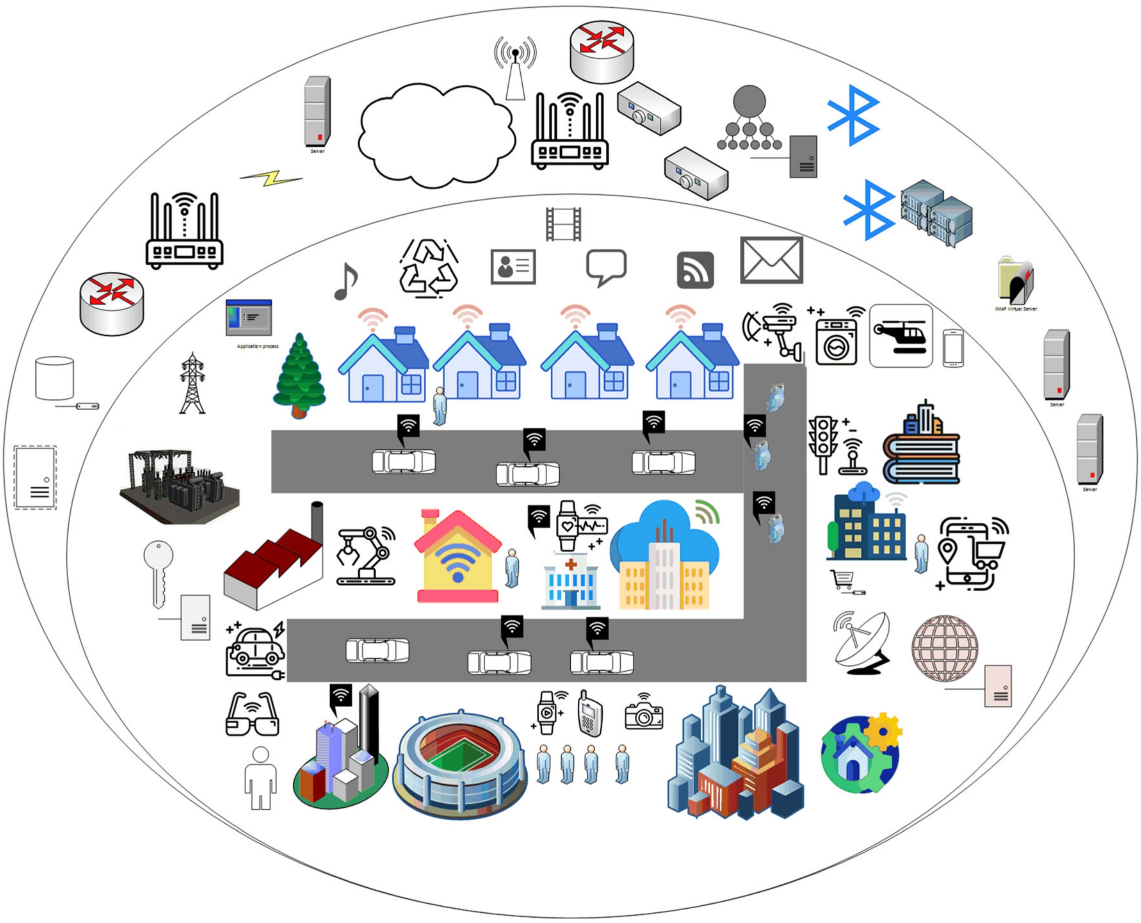

Fig. 8 The basic architecture of the smart city

[49] identified four areas in realizing the SC: industry, education, participation, and technical infrastructure. [50] explored six dimensions of the SC and their related aspect of urban life: smart economy (industry), smart people (education), smart government (e-democracy), smart mobility (logistics and infrastructures), smart environment (efficiency and sustainability) and smart living (security and quality).

From the technological surface, Eremia et al. [1] defined SC as a city with a great achievement of ICT technologies. These had spread through the commercial application of intelligent acting products and services, artificial intelligence, and thinking machines. The examples of systems dressed up with number of mobile terminals and embedded devices including connected sensors and actuators are smart homes and smart buildings. The SC becomes smarter depending on the nature of the digital technology which is needed by different electronic services used by the various applications such as the street cameras for surveillance system, sensors for the transportation system. However, several initiatives utilize objects to provide different value added services, such as Google street view and global positioning system (GPA) [48].

The common characteristics of SC as reported in Nam and Pardo [51] include:

- The utilization of networked infrastructure to improve economic and political efficiency and enable social, cultural, and urban development.
- An underlying emphasis on business-led urban development.
- A strong focus on the aim of achieving the social inclusion of various urban residents in public services.
- A stress on the crucial role of high-tech and creative industries in long-run urban growth.
- Profound attention to the role of social and relational capital in urban development.
- Social and environmental sustainability as a major strategic component of the SC.

Zygiaris [52] identified six layers of SC as presented in Table 2.

Sullivan [50] identified the key parameters that will change the concept of SC in 2020: smart energy, smart building, smart mobility, smart technology, smart healthcare, smart infrastructure, smart governance and education, smart security and smart citizens. Bergh and Viaene [53] highlight the key challenge for the SC: Ecosystem position, SC leadership locus, internal coordination mechanisms, business IT alignment, shaping organizational culture and going beyond experiments. Therefore, cities should found their SC models on three main pillars: infrastructure, human capital and information [54].

At the layer 0 of the SC, operations and processes in the conventional city are tailored toward achieving the SC initiatives in form of responding to the city challenges. The challenges include embedding of the urban planning with smart operations, infrastructure intervention, identity of the smart priorities that is compatible with the city planning and innovations, etc. In layer 1, these challenges outlined in layer 0 are moderated into the green city priority and it requires innovation in green governance, integration of policies and financial resources allocation for developing the appropriate and sustainable green urban ecosystem. The challenge of the policy makers in layer 1 is the issue of CO_2 emission reduction and alternative form of energy sources. The applications in any of the SC layer should have the capability for intercommunication, layer 4 to share data, services, content, etc. The SC monitoring processes have the ability to moderate, integrate and ensure the availability of digital resources in the SC. For instance, layer 4 provides the enabling environment for platforms to reside on fiber-optics network, the layer 2 that houses pipe lines of water embedded with sensors in the layer 3 allows the flow of real-time data of optimal water usage applications in layer 5. The effective and efficient integration of different SC applications for openness of resources is a challenged. The real-time devices in the SC are connected onto the fiber-optics through layer 2 interconnection, monitored through the layer 4 integration layer and subsequently, layer 5 nourishes the intelligent applications with real-time data. Lastly, layer 6 of the SC creates the enabling environment for new opportunities of doing business and makes the SC fertile for new innovations [52].

From the interaction of the SC layers from layer 0 to 6, it clearly pointed out that data can be generated, acquired and analyze. The deep learning algorithms can be applied for data analytics for the data acquired from the SC as a result of interaction of the various layers. Analyzing the data by applying deep learning can provide an insight and uncover new knowledge which can be used to enhance decision making which in turn improves the efficiency of the SC in general.

For example, result of applying deep learning for data analytics in layer 1 can be used for planning the reduction of CO_2 emission because of the deep learning prediction of CO_2 emission and energy consumption. As such, improve energy efficiency. Deep learning analysis of data from layer 2 has the potential to indicate where deployment of smart systems is needed or otherwise. In layer 4, the deep learning can be applied for analytics to provide information on where resources are concentrated which can help the policy makers to device ways for diversification.

The real-time data can be analyze with the deep learning for adaptation of the intelligent applications in layer 5. The deep learning data analytics result can help the decision makers in formulating policy and regulatory framework for digital inclusive in layer 0. Therefore, improve community involvement in the SC initiatives. The deep learning analytics output in layer 3 can be helpful in providing the decision makers with insight on usage of the communication technologies such as but not limited to sensors, RFID

Table 2 The six layers of Smart cities with their corresponding concept

S/n	Layer	Concept/definition
1.	Layer 0: The city layer	This layer is the layer where the information is generated to improve the quality of life
2.	Layer 1: The green city layer	This layer contains natural resources that are reproducible such as heat, water, and wind power
3.	Layer 2: The interconnection layer	The SC applications should be able to intercommunicate and share data, content and services
4.	Layer 3: The Instrumentation layer	This layer is the layer for cleaning, maintenance, property, leasing, technology, and operating modes associated with facilities in the urban area
5.	Layer 4: The open integration layer	In this layer, all the components of the system are properly integrated
6.	Layer 5: The application layer	It shows the mirrors of the real-time city operation pulse as system with systems
7.	Layer 6: The innovation layer	This layer of SC creates a producible innovation environment for new business opportunities

and activators. Therefore, aid decision makers to take decision on effective sharing of digital resources. The deep learning when applied for analytics on data from business operations in the SC, it can give understanding on new business opportunities and areas that require innovations in layer 6.

4.1 Data generation from the smart city

As a result of interconnected devices in the SC, a large amount of data has been generated from the devices as a result of activities. Mazhar Rathore et al. [55] pointed out that the interconnected thousands of the IoT devices while communicating with each other over the Internet to establish a smart system results in the generation of large amount of data. The results of the data analytics can be useful for many purposes and benefited the society, businessman and the government authorities. Kök et al. [56] with the increase in number of IoT based SC applications, the amount of data produced by these applications is increasing tremendously. Governments and city stakeholders take early precautions to process these data and predict future effects to ensure sustainable development. Smart meters have become one of the main components of the SC strategies. Smart meters generate large amounts of fine-grained data using pressure and released model that is used to provide useful information to consumers and utility companies for decision making [57]. The core of it is the collection, management, analysis and visualization of large amount of data that is generated every minute in an urban environment due to socioeconomic activities [4].

4.1.1 Big data in smart city

With the adoption of IoT in the urban cities, a huge amount of data is being generated by the smart city applications is rising, but companies and governments are utilizing only a part of the data generated. There is a need for possibilities to improve the efficiency of the city services. AI-based analytic applications are rapidly offering various numbers of opportunities to individuals and industrial organizations. Such applications can be classified into smart homes, smart transportation, smart healthcare, smart wasted management, etc. Such applications are the key requirement nowadays, and it can be considered as a primary requirement in big data generation. Kitchin [58] provided a review on the use of digital devices in SC to produce big data. The collected data through these devices enable real-time analytics, new modes of urban governance and enacting more efficient sustainability. Moreover, the author discussed critical reflection on the implications of big data and smart urbanism, examining five emerging concerns. Al Nuaimi et al. [20] presented the applications of big data to support SC. The authors compared various definitions of the SC and big data to identify some of the opportunities and challenges with respect to big data application for SC. Sood et al. [59] used big data and high-performance computing (HPC) to propose smart flood monitoring and forecasting architecture from the social collaborative IoT. The purpose of the study is to categorize the geographical areas into a web of hexagonal for effective installation of energy efficient IoT devices. The authors utilize a clustering algorithm like K-mean to predict the current state of flood and flood rating in any location. Studies by Krieg et al. [60] have shown that urban data gathered can be used to develop a parking system to reduce the traffic congestion in cities. In this regard, the authors developed and implemented SmartPark in San Francisco cities. The system depends on pervasive Wi-Fi and cellular infrastructure, which is capable of providing drivers with real-time parking availability information. Applying big data technologies in the SC allow the storage and processing of the data to provide new information. Subsequently, the information can be utilized to improve the distribution of services in the SC [61].

5 Deep learning applications in smart cities

This section presents deep learning solutions in SC from various projects. The section is classified based on deep learning architecture and shallow ANN. For the deep learning architecture, it includes: ConvNet, DBN, DRL, DRNN and Hybrid. From the perspective of the shallow ANN, the architecture involved SVM, BPNN and others. This section comprised of Tables 3, 4, 5, 6, 7, 8, 9, 10 and 11 that presents the summary of the projects discussed in each of the section. The tables have 4 columns each. The first column is the references of the project, second column is the algorithm that is propose in the project to solve problem in the SC. Third column presents the algorithms used to evaluate the effectiveness and efficiency of the algorithm propose in the project as it is the typical practice in machine learning. In machine learning, any algorithm can be proposed to solve a problem in a project; its performance must be compared with the classical algorithms to prove that the algorithm is suitable for solving the problem. The comparison will show the advantage or otherwise the proposed algorithm in solving that particular problem. This is necessary as the performance of machine learning algorithms is not uniform across different domain and problems. Many factors determine the suitability of algorithm in solving a problem such as nature of the data, size of the data, objective function, computing time, problem to tackle, dateline to deliver result and hardware

Table 3 The summary of the ConvNet application in smart city

References	The proposed algorithm	Algorithm compared	Contribution
Valipour et al. [43]	D-ConvNet	ANN	D-ConvNet is more accurate and robust than the compared algorithm
Amato et al. [42]	D-ConvNet	SVM	D-ConvNet outperforms shallow SVM
Amato et al. [62]	ConvNet	SVM	ConvNet has good generalization capabilities than the SVM
Giyenko et al. [63]	ConvNet	Not compared	ConvNet has good accuracy
Rajput et al. [64]	DNN-C	Canny edge detector	DNN-C is more robust and efficient than the compared algorithm
Liu et al. [65]	ConvNet	ConvNet with dropout layer	ConvNet is more accurate than the compared algorithm
Kuang et al. [66]	ConvNet	R-ConvNet	ConvNet has more detection rate than the R-ConvNet
Jiang et al. [67]	D-ConvNet	superpixel segmentation algorithm.	D-ConvNet has good performance the compared algorithm
Li et al. [68]	ConvNet	Multiscale structural features and SVM	ConvNet has good detection rates than the compared algorithm
	D-ConvNet	VGG16 and AlexNet	D-ConvNet outperforms VGG16 and AlexNet
Lwowski et al. [69]	D-ConvNet	feature pyramids algorithm	D-ConvNet outperforms other detection techniques
Guo et al. [70]	ConvNet	SVM	ConvNet has better accuracy than the SVM
Liu et al. [71]	ConvNet	Traditional human hand crafting method	ConvNet has better accuracy than the compared algorithm
Yeshwanth et al. [72]	D-ConvNet	BS method	D-ConvNet outperforms the compared method
Zhang et al. [10, 11]	D-ConvNet	sparse coding method	D-ConvNet outperforms the compared method
Aqib et al. [73]	D-ConvNet	Disaster management methods	D-ConvNet outperforms other disaster management methods
Huh and Seo [74]	ConvNet	Not applicable	The ConvNet is found to be good for the task

Table 4 The summary of the DBN application in smart city

References	The proposed algorithm	Algorithm compared	Contribution
Huang et al. [37]	DBN	SVR	The DBN outperform the SVR
Han and Sohn [75]	DBN	*k*-means	The DBN outperform the *k*-means

Table 5 The summary of the application of RNN in smart city

References	The proposed algorithm	Algorithm compared	Contribution
Song et al. [76]	D-LSTM	Shallow LSTM	D-LSTM is more accurate than the shallow LSTM
Kök et al. [56]	LSTM	SVR	LSTM outperforms SVR
Struye et al. [77]	GRNN	NP	GRNN has better accuracy than the NP
Pan [78]	LSTM-RNN	RW, SVM, FFNN and SAE	LSTM-RNN perform better the RW, SVM, FFNN and SAE
Qolomany et al. [79]	D-LSTM	ARIMA	The D-LSTM is more accurate than the ARIMA
Fu et al. [80]	LSTM-G-RNN	ARIMA	LSTM-G-RNN outperforms the ARIMA
Alahi et al. [81]	LST	Hand-crafted function method	The LSTM outperforms the compared method
Salvador et al. [82]	D-RNN	HA	The D-RNN outperform HA
Camero et al. [83]	RNN-DNE	LR, GP and SVM	RNN-DNE performs better than the compared algorithms

Table 6 The summary of the ANN application in smart city

References	The proposed algorithm	Algorithm compared	Contribution
Jain and Shah [84]	ANN	SVM	ANN gives better accuracy than SVM
Gupta et al. [85]	ANN	Not specified	ANN is tested with good accuracy.
Vlahogianni et al. [86]	ANN	Not specified	ANN performs better with accuracy
Sharad et al. [87]	ANN	MLR	ANN outperforms MLR
Mehmood et al. [88]	D-ANN	NB, KNN, DT, SVM and ANN	D-ANN performs better than the compared algorithms
Balchandani et al. [89]	D-ANN	Conventional ML algorithms	The D-ANN outperforms the compared algorithms
Yuan et al. [90]	BPNN-DFIG	Not specified	The BPNN-DFIG improve the stability of the wind turbine

Table 7 The summary of the SVM applications in smart city

References	The proposed algorithm	Algorithm compared	Contribution
Dambhare and Karale [91]	SVM	Sentiment analysis	SVM has better performs
Gu et al. [92]	SVM	Secure key-based, resource testing–based, reputation system and position verification	SVM outperforms the compared methods
Yan and Yu [93]	Improved SVM	SVM and TBSVM	Improved SVM has good accuracy than the compared algorithm
Hanifah et al. [94]	SVM	Feature selection and Weighting methods	SVM outperforms the compared algorithm

Table 8 The summary of the hybrid neural network as applied in smart city

References	The proposed algorithm	Algorithm compared	Contribution
Liang et al. [95]	VH-RNN	NARNNX	The VH-RNN outperforms NARNNX
Chen et al. [96]	D-ConvNet	Traditional DNN	The D-ConvNet significantly outperforms traditional DNN
Zheng et al. [97]	RT-ANN-SVR	ANN and SVR	The RT outperforms ANN and SVR
Niu et al. [98]	RBM-SVM	Temporal supplementary and deep learning approaches	The RBM-SVM achieves better accuracy than the compared algorithm
Aryal and Dutta [99]	DNN-RBM-CMPN	ARIMA	The DNN-RBM-CMPN outperforms the compared algorithm
Zhu et al. [100]	ConvNet-RNN	CrowdNet	The ConvNet outperforms the CrowdNet
Dinh et al. [101]	ConvNet-ReNN	Sliding window method	The ConvNet-ReNN has good accuracy than the compared method
Jindal et al. [102]	ConvNet-SVR	Single ConvNet single SVR	ConvNet-SVR does better than single ConvNet, single SVR
Chackravarthy et al. [103]	NN-HDL	Crime forecasting and mapping methods	NN-HDL outperforms the compared methods
Camero et al. [104]	D-RNN-EA	Not specified	D-RNN-EA outperforms other predictors
Atef and Eltawil [105]	SVR-DL	SVR	SVR-DL is more accurate than SVR

Table 9 The summary of the EANN application in smart city

References	The proposed algorithm	Algorithm compared	Contribution
Feng [106]	GA-NN	WNN	GA-NN has good performance than the compared algorithm
Lei and Shangzheng [107]	GA-RBN	Standard GA	GA-NN is better than the standard GA
Belhajem et al. [108]	EKF-ANN-GA	Simple EKF	EKF-ANN-GA outperform the simple EKF

Table 10 The summary of the DRL application in smart city

References	The proposed algorithm	Algorithm compared	Contribution
Heo et al. [109]	DeepRL	Q-L	DeepRL perform better than Q-L
Jang et al. [110]	DeepRL	Not specified	DeepRL performs much better
Wu et al. [111]	DeepRL	ConRL	DeepRL outperforms ConRL
Zhao et al. [112]	DeepRL	MAC and BEB	DeepRL outperform the OSPF and EOSPF
Ali et al. [31]	DeepRL-QL		DeepRL-QL outperform BEB and MAC
Mohammadi et al. [113]	SSDRL	SDRL	The SSDRL performs better than the SDRL

Table 11 The summary of the applications of other deep and shallow neural networks in smart city

References	The proposed algorithm	Algorithm compared	Contribution
Sun et al. [114]	SWANN	Conventional ANN	The SWANN is more accurate than the conventional ANN
Xu and Gade [115]	SD-NN	CMLR	The SD-NN outperforms CMLR
Gupta et al. [116]	HNN	GA, PSO and DE	The HNN has higher flow rates than the compared algorithm
Ghoneim and Manjunatha [117]	DNN	SVM, ANN and GLM models	The DNN is superior than the compared algorithm
Bura et al. [118]	DNN	Standard AlexNet and mAlexNet	The DNN has higher accuracy and much faster than the compared models
Elsaeidy et al. [119]	RBM	SVM	The RBMs performs better than the SVM
Li et al. [120]	IDS-DMLM	ELM and BPNN	The IDS-DMLM is found to perform better than the ELM and BPNN
Serrano and Bajo [121]	DNN	Random forest, K-NN, Logistic, C4.5, AdaBoost and RIPPER	The DPP performs better than the compared algorithms
Zhang et al. [122]	Fine-grained DNN	Coarse-grained DNN	The fine-grained DNN outperforms the coarse-grained DNN in terms of energy efficiency

availability. The last column is the contribution of the project showing the advantage of the propose algorithm.

5.1 Convolutional neural network

The ConvNet has been found to be effective in the SC, for example, Valipour et al. [43] propose deep ConvNet (D-ConvNet) to detect vacant and occupied parking slots in smart parking stall. The D-ConvNet is used in classifying vacant and occupied parking slots. The result shows that D-ConvNet is more accurate and robust than the ANN. Amato et al. [42] propose D-ConvNet to classify images of parking space occupancy for smart cameras. The study exploited large number of ground-truth label data to discover which features to discriminate for each class of the objects to be recognized, and builds a combined feature extraction and classification model based on the D-ConvNet. The D-ConvNet is applied in classifying images of the parking space occupancy. It was found that the D-ConvNet outperforms the SVM.

Amato et al. [62] propose ConvNet to detect car parking occupancy. The ConvNet is used in classifying the car parking occupancy. The ConvNet has good generalization capabilities than the SVM as shown by the experimental results. Giyenko et al. [63] propose ConvNet to predict atmospheric visibility from weather monitoring system of a SC. The ConvNet has three convolutional layers and max pooling layer that act as the convergence of the network. The ConvNet is used to predict the atmospheric visibility. The result shows that the ConvNet has good accuracy for predicting the atmospheric visibility. Rajput et al. [64] propose a deep ANN (DNN) with convolutional layer (DNN-C) to model the human driver capabilities for autonomous taxi in SC. The DNN-C is responsible for navigating the vehicles autonomously and determining whether the road is straight or turning or intersection. The DNN-C classifies the road for determining the movement of the autonomous vehicle. The result of the proposed model shows that it is more robust and efficient than the canny edge detector. Liu et al. [65] propose ConvNet to recognize plate number and characters without segmentation. The ConvNet derives effective representations of the original image, which makes it to identify the visual rules directly from the original pixels. The ConvNet is applied in training number plate samples and recognize characters without segmentation. The result shows that ConvNet has better accuracy than the ConvNet with dropout layer.

Kuang et al. [66] propose ConvNet to detect pedestrian in images. The ConvNet is responsible for analyzing the different region of the images to extract feature information. The ConvNet is used in the detecting of the pedestrians. The result shows that ConvNet provides superior detection rate than the recurrent ConvNet (R-ConvNet). Jiang et al. [67] propose D-ConvNet to detect vehicle in satellite images. The D-ConvNet is applied in classifying vehicle in satellite images. The result shows that D-ConvNet outperforms the superpixel segmentation algorithm. Li et al. [68] propose ConvNet to predict road region. The ConvNet captures big hold of local structures of road. The ConvNet is utilized in predicting the road region. It has been found that the ConvNet has good detection rates than the multiscale structural features and SVM. Aqib et al. [73] propose D-ConvNet to predict urban traffic behavior in a SC. The data was collected on traffic and extracts the flow, speed, occupancy and density. The D-ConvNet is applied to predict traffic flow. Result indicated that D-ConvNet outperforms the disaster management methods. Huh and Seo [74] propose ConvNet to classify shoe, provide automatic storage and recommend shoe in a shoe cabinets for a smart home. The ConvNet was trained by shoe image data. The ConvNet was found to be a good algorithm for the task.

Mauro et al. [123] propose D-ConvNet to detect the availability of parking stall occupancy. The D-ConvNet extracts space and checks whether the parking space is available or empty. It was found that the D-ConvNet has higher accuracy than the VGG16 (VGG16 is ConvNet with 16 convolution layers that run on ImageNet to win image processing competition in 2014) and AlexNet. Lwowski et al. [69] propose D-ConvNet to detect pedestrian in smart communities. The D-ConvNet is used to notify the driver if a pedestrian is on the road or not. The result shows that D-ConvNet outperforms fast feature pyramids algorithm. Guo et al. [70] propose ConvNet to detect traffic signals and light in an area. The ConvNet uses small filter kernel to increase the rate of computation, decrease the feature maps' thickness and number of parameters. The ConvNet is applied in classifying traffic signals and light in an area. The result shows that ConvNet outperforms the SVM. Liu et al. [71] propose ConvNet to detect pedestrian heads and compute the number of the pedestrian heads. The ConvNet performs feature extraction, selection and applies SVM to count the number of people in the pedestrian crowd. The result shows that ConvNet outperforms the traditional human hand crafting method.

Yeshwanth et al. [72] propose D-ConvNet to detect traffic network to decrease congestion and increase the total travel experience of passengers. The D-ConvNet classifies the vehicles into best resolution region and estimates the traffic density. The D-ConvNet has better accuracy than the background subtraction (BS) method. Chen et al. [96] propose D-ConvNet to detect vehicles from satellite images. The division maps of the last convolutional layer and the max pooling layer of D-ConvNet into multiple blocks of variable receptive field sizes to enable the proposed method to extract variable scale features resulted to D-ConvNet. The D-ConvNet is used to detect vehicles from satellite images. The D-ConvNet significantly outperforms the traditional deep neural network (DNN) on vehicle detection. Zhang et al. [10, 11] propose D-ConvNet to detect vehicle type in traffic videos. The D-ConvNet extracts features and separates the various types of vehicle into different classes. The result shows than the D-ConvNet has better accuracy than the sparse coding method. The summary of the application of ConvNet in analyzing data extracted from SC is presented in Table 3.

Use case

Recent years have perceived an increasing trend in image processing due to convolutional neural network, which has become a significant approach for handling large amount of images produced by the smart cities. An example of use cases is presented in Lwowski et al. [69], which introduced a pedestrian detection system for smart communities using deep convolutional natural network. The idea is to overcome the challenges of autonomous driving, search and rescue, surveillance and robotics by recognizing the pedestrian. The study is based on fast regional detection cascaded with deep convolution networks which enable real-time pedestrian detection. The result shows that using convolutional neural network the system is able to accurately detect the pedestrian by 95.7% with a processing rate of about 15 frames per second on a low performance system without a graphical processing unit (GPU). Moreover, convolutional neural network has been constantly used in medical analysis as report in Anwar et al. [124]. The study concluded that convolutional neural network based on deep learning is being utilized in multiple fields of medical image analysis including segmentation, classification and detection.

5.2 Deep belief networks

The DBN is among the deep learning architecture found to be applied in the SC for data analytics. The DBN is not heavily used in the SC compared to the ConvNet. However, the DBN has been successfully applied in SC and found to be effective and efficient, for example, Huang et al. [37] propose DBN to predict traffic flow. The DBN is for unsupervised feature learning which has the learning unit activations act as the data. The DBN is used to predict the traffic flow. It was found that the DBN improved the accuracy of traffic flow prediction over the support vector regression (SVR).

Han and Sohn [75] propose DBN to cluster zones of Seoul metropolitan area in travel patterns transit passengers. The DBN separates an input space into a predefined number of clusters, in such a way that distances within clusters are minimized while distances between clusters are maximized in a feature space. The DBN is applied in clustering zones of Seoul metropolitan area in travel patterns of transit passengers. The result shows that DBN has more realistic information than the k-means algorithm.

Use case

A great success has been observed in many studies utilized deep belief networks approach for applications such as speech recognition and image classification (Dedinec et al. [125]). For instance, Dedinec et al. [125] have applied deep belief network to short-term electricity load forecasting based on the Macedonian hourly electricity consumption data in the period 2008–2014. The study shows that deep belief network provides a better result compared to traditional methods used in forecasting electricity. In addition, in monitoring flood in smart city the deep belief network has been used. Mishra et al. [126] used IoT enabled camera and deep learning to monitor gullies and drainages based on image classification. The purpose of the study is to classify blockage images into different class labels based on the severity. The result obtained shows an improvement in the classification accuracy.

5.3 Deep recurrent neural network

The deep structure of the RNN has been used in the smart city for analyzing data to improve the concept of the SC. For instance, Song et al. [76] propose deep long short-term memory (D-LSTM) to predict human mobility and transportation mode. The D-LSTM has multiple hidden layers. The D-LSTM is utilized in predicting human mobility in transportation system. The result shows that D-LSTM is more accurate than the shallow LSTM. Qolomany et al. [79] propose D-LSTM to predict the number of occupants at a given location and time in a smart building. The D-LSTM is trained using Wi-Fi time-series data. The D-LSTM is used in predicting the number of occupants at a given time and location. Results of the experiment indicated that the D-LSTM outperforms the auto regression integrating moving average (ARIMA). Other forms of the RNN such as LSTM are found to be applied in the SC, for example, Kök et al. [56] propose LSTM networks to predict quality of air in a SC. The LSTM is a special type of RNN for supervised temporal sequence learning. The proposed algorithm is applied in predicting future values of air quality in the SC. The result shows that the LSTM-based model has a better performance than the SVR.

Pan [78] proposes LSTM to predict short-term traffic flow in a SC. The LSTM is for memorizing long historical input data and automatically determines the optimal time lags desirable in short-term traffic flow prediction. The LSTM is used in predicting the short-term traffic flow. The result shows that the LSTM achieves higher accuracy than the random walk (RW), SVM, single layer feed forward NN (FFNN) and stacked autoencoder. Fu et al. [80] propose LSTM gated RNN (LSTM-G-RNN) to predict short-term traffic flow. The LSTM-G-RNN is applied in predicting short-term traffic flow. The LSTM-G-RNN is responsible for language models and has the ability to memorize long-term dependencies. The result shows that the LSTM-G-RNN performs better than the ARIMA. Salvador et al. [82] propose D-RNN to automatically map a GPS trace onto a transport network. The samples of a space of all possible journeys within the network were acquired to get large amount data to train D-RNN. The D-RNN is applied for the mapping as classification problem. The performance of D-RNN is compared with heuristic algorithm (HA). The result shows that the D-RNN performs better than the HA. Camero et al. [83] propose hybrid of RNN and deep neuroevolutionary (RNN-DNE) to predict the filling level of the waste containers in a real city. The RNN is designed with deep neuroevolutionary (DNN) to create RNN-DNE for predicting the waste containers filling level. The performance of the RNN-DNE is compared with the linear regression (LR), Gaussian processes (GP), and

SVM. The result shows that the RNN-DNE has a higher predictive accuracy than the LR, GP and SVM.

Alahi et al. [81] propose LSTM to predict people and their future trajectories in crowded area. The LSTM learn position of the person and predicts their future positions to incorporate within the trajectories of the pedestrians. The LSTM is applied in predicting human mobility and their future trajectories in crowded area. The result shows that LSTM performs better than the hand-crafted function method. Struye et al. [77] propose gated RNN (GRNN) to predict interference data from the city environment of a SC. The RNN recognizes temporal patterns by remembering aspects of past inputs in their internal memory, the gates were classified into reset gates which decide which aspects of the state to forget, and the update gates decide how to incorporate the new data into the state. The GRNN is applied in predicting the interference data of city lab testbed. It was found that the GRNN is more accurate than the naïve predictor (NP). The summary of the application of RNN in SC is presented in Table 5.

Use case

Because of increasing progress analytics for smart city, it has become necessary to develop applications based on deep recurrent neural network. Lin et al. (2020) presented study on the implementation of deep recurrent natural network to forecast the opening price in the Financial Index. The authors compared the results with the previous studies used machine learning; it has been found that the deployment of deep recurrent natural network on some case such as S&P500 (Standard & Poor's 500) and the Dow Jones stock indices can show better performance. In Taiwan due to climate conditions issues (Wei and Cheng 2020) two-step wind-wave prediction is proposed. The Guishandao Buoy Station located off the northeastern shore of Taiwan was used for a case study. The results achieved have shown that the proposed approach is more accurate compared with other approach such as MLP network.

5.4 Backpropagation neural network and deep neural networks

The BP algorithm for ANN is still active in the literature for training the ANN. The BP is applied in ANN for application in the SC. For example, Mehmood et al. [88] propose deep ANN (D-ANN) to detect and control teaching and learning in SC. The ANN is trained using BP and is responsible for identifying user components of ubiquitous teaching and learning. The result shows that the D-ANN has superior accuracy than the Naïve Bayesian (NB), *k*-nearest neighbor (KNN), decision tree (DT), (SVM) and ANN. Balchandani et al. [89] propose D-ANN to detect litter object on the smart street cleaning place. The D-ANN checks the street images and finds out whether the streets are dirty or not. It was found that D-ANN has better accuracy than the conventional ML algorithms. Jain and Shah [84] propose ANN to detect the anomalous location of air pollution based on air quality index in a SC. The ANN is used to classify the data as either normal or anomalous. The result shows that ANN has better accuracy than the SVM. Yuan et al. [127] propose BPNN based on double-fed induction generators (BPNN-DFIG) integrated with supercapacitor energy storage system (SCESS) to reduce power instability. The BPNN-DFIG transmitted sensor node to the aggregation, the aggregation node passed to the gateway and the data is uploaded to the cloud environment through the internet by the gateway. The BPNN-DFIG is applied for clustering the architecture to enhance stability of wind turbines. The ANN-DIFG is evaluated and found that it has higher stability improvements.

Gupta et al. [85] propose ANN to detect the possibility of waterlogging and its intensity in the future areas of a SC. The ANN is used in predicting the future intensity and possibility of waterlogging areas. It was found that the ANN has good accuracy in determining the intensity of the waterlog. Vlahogianni et al. [86] propose ANN to predict available parking space in short- and long-term period. The ANN is used in predicting available parking space in short- and long-term period. The result shows that the ANN performs better with accuracy. Sharad et al. [87] propose ANN to predict an estimate of arrival time to commuter using smart bus. The ANN is trained using BP to estimate the arrival time of the bus. The result shows that ANN outperformed the standard multiple linear regression (MLR). Table 6 presents the summary of the BP-ANN in processing SC data.

Use case

Audio data classification based on deep neural networks and multilayer perceptron neural networks have been implemented in Zamil et al. [128]. The idea is to propose an affective network topology using hidden layers, neurons and fitness function approaches. The results have produced a high-performance classifier to improve the accuracy of large audio datasets that collected from different modalities in a smart city. Moreover, network traffic prediction is designed for smart cities using back propagation and neural network (Pan et al. [129]). The concept is based on impact factor of network as the input layer and the network traffic as the output. The results show that the proposed approach has accurately predicted network traffic in smart city environment.

5.5 Support vector machine

The SVM is used in the SC for data analytics in different domain within the SC community. For example, Gu et al. [92] applied SVM to detect road safety and driving patterns. The SVM differentiates between the bad nodes and the good nodes in vehicular network. The SVM is applied to classify the road safety and driving pattern. The result shows that the SVM has better detection rate than the secure key-based, resource testing-based, reputation system and position verification. Similarly, Dambhare and Karale [91] propose SVM for classification in mapping the world. The SVM is responsible for the classification whether the object is positive or negative. The result shows that the SVM has better performance than the sentiment analysis.

Yan and Yu [93] propose an improved SVM to predict short-term traffic congestion. The improved SVM computes the total speed of each region in the urban road network and divide the rod congestion into different traffic congestion levels. The improved SVM is used in predicting the short-term traffic congestion. The result shows that the improved SVM outperforms the conventional SVM and twin bounded SVM (TBSVM). Hanifah et al. [94] propose SVM to detect twitter information. The SVM extracts the information of traffic congestion and classifies the tweet as either the positive or negative. It was found that SVM has better accuracy than the feature selection and weighting methods. Table 7 presents the summary of the SVM in SC.

Use case

Support vector machine algorithm is considered one of the important algorithms in many applications related to image processing. Hossain et al. [130] have designed a framework to deal with media-related healthcare data for handling healthcare in smart city. The authors introduced a cloud-oriented smart healthcare monitoring framework that interacts with surrounding smart devices, environments, and smart city stakeholders for affordable and accessible healthcare (Hossain et al. [130]). Moreover, remote sensing extraction techniques based on support vector machines are introduced in order to improve the soil salinization maps. The results show that the proposed method outperformed the existing methods in extracted information of soil salinization.

5.6 Hybrid neural networks

In this section, the hybrid of the deep learning architecture, shallow ANN and other classes of ANN are presented in relation to their applications in SC. Artificial intelligent algorithms are hybridize to avoid their weaknesses and strengthen performance. In the SC, many hybrid systems are found to be applied for processing the SC data. For example, Niu et al. [98] propose a hybrid of RBM and SVM (RBM-SVM) to predict traffic flow. The RBM is trained to reduce dimensional feature, whereas the SVM performs feature extraction. The RBM-SVM is applied in predicting road traffic flow. The result shows that the RBM-SVM outperforms temporal supplementary and deep learning approaches. Aryal and Dutta [99] propose a hybrid of DNN based on restricted Boltzmann machine and conventional multi-layer perceptron network (DNN-RBM-CMPN) to detect objects from images. The DNN-RBM-CMPN extracts features and classifies the object from the image. The result shows that DNN-RBM-CMPN classifies much better than the ARIMA. Zhu et al. [100] propose a hybrid of ConvNet and RNN (ConvNet-RNN) to predict the number of people and count the number of pedestrians passing through a place in a specific time. The ConvNet learns the body features and distinguishes between the people close to the camera and the people that are far away. The RNN counts the number of people passing through a place in a specific time. It was found that the ConvNet-RNN outperforms the CrowdNet. Dinh et al. [101] propose a hybrid of ConvNet and recursive NN (ReNN) (ConvNet-ReNN) to detect pedestrian. The ConvNet learn low level features and recognize image color. However, the ReNN learns high-level features and classifies the present of pedestrian. The ConvNet-ReNN outperforms the sliding window method.

Jindal et al. [102] propose a hybrid of ConvNet and SVR (ConvNet-SVR) to predict the total load consumption of all smart homes in a SC. The ConvNet learns the hidden patterns in the data and outputs different load curves while the SVR trained the load curves. The hybridized model is compared with single ConvNet and single SVR. The result shows that ConvNet-SVR does much better than the single ConvNet and single SVR. Chackravarthy et al. [103] propose a combination of ANN with hybrid of DL (NN-HDL) algorithms to detect crime activities in a SC. The NN-HDL algorithms analyze video stream data in order to control criminal activity which in turn helps to minimize workload on the supervising officials. The model is compared with crime forecasting and mapping methods. It was found to be more accurate than the compared methods. Camero et al. [104] propose D-RNN with EA (D-RNN-EA) to predict car park occupancy rate. The D-RNN-EA aggregates parking lots and predicts car park availability. The model is evaluated and compared with other predictors. It was found to be more accurate than the compared methods. Atef and Eltawil [105] propose a hybrid of support vector regression (SVR) and DL (SVR-DL) to predict the hourly price for electricity price forecasting in Smart Grids. The SVR-DL was trained with a data that represents one-hour observation and indicates the changing price range based on the power consumption. The hybridized model is compared with SVR. The result indicates that SVR-DL is more accurate than the SVR.

Liang et al. [95] propose a hybrid of vanilla-RNN and hyper-RNN (VH-RNN) to predict the number of metro passengers entering stations. The VH-RNN is responsible for capturing key information of previous states from different input sources; VH-RNN performs weigh sharing scheme among the models which resulted to VH-RNN. The VH-RNN is applied to predict the number of metro passengers. The results show that the performance of the VH-RNN is better compared with the nonlinear autoregressive ANN with external input (NARNNX). Zheng et al. [97] propose a hybrid of regression tree (RT), ANN and SVR (RT-ANN-SVR) to predict the parking occupancy rate. The RT-ANN-SVR forecasts the occupancy rate of parking lots given a specific date and time. It was found that RT-ANN-SVR performs better than the ANN and SVR. Table 8 presents the summary of the application of hybrid algorithms in SC.

Use case

Lee et al. [131] used deep hybrid neural network for event-based Optical Flow estimation with energy. The authors developed Spike-FlowNet system to overcome the issue of handling the performance due to spike vanishing phenomenon. Spike-FlowNet outperforms its corresponding ANN-based method in terms of the optical flow prediction capability while providing significant computational efficiency. Moreover, Sokolov et al. [132] introduced the results of simulation which include an assessment of the reliability of calculating the Earth's magnetic field parameters of the magnetic control system for the spacecraft momentum.

5.7 Evolutionary artificial neural network

The evolutionary artificial neural network (EANN) is the application of evolutionary algorithms such as genetic algorithm (GA) for the optimization of ANN parameters. The EANN has started penetrating the SC; few studies were found to apply the EANN in the SC. For example, Feng [106] propose the optimization of ANN through GA (GANN) to predict and model network traffic in SC. The GA is applied to optimize the ANN weights and threshold. The GANN is used in predicting traffic network. Its performance is compared with that of wavelet NN (WNN). The result shows that the GANN has good performance than the WNN. Lei and Shangzheng [107] propose GANN to predict network traffic flow. The GA is used to optimize the parameters of radial bases function (RBF) NN. The GA-RBF is applied in predicting the network traffic flow. The result shows that the GA-RBF is better than the standard GA.

Ikram Belhajem proposes a hybrid of extended kalman filter (EKF) and ANN based on GA (EKF-ANN-GA) to predict the real-time car positioning in a SC. The EKF estimates the current state of a system. The ANN acts as a massively parallel distributed processor that uses flexible computing paradigms to solve complex and uncertain problems. The GA facilitates the connection weights optimization in the training phase of the ANN. This hybridization resulted to EKF-ANN-GA. The EKF-ANN-GA is applied in predicting the real-time car positioning in a SC. The result shows that the EKF-ANN-GA outperforms the simple EKF. The summary of the evolutionary ANN is presented in Table 9.

5.8 Deep reinforcement learning

DeepRL is the hybrid of reinforcement learning and deep learning architecture. It has been found to be applied in SC to solve problems. For example, Heo et al. [109] propose DeepRL to monitor ventilation system for air quality in a smart building. The DeepRL monitors the dynamic properties of the control variables and reduces the energy usage. The proposed algorithm is compared with quality learning (Q-L) and found that DeepRL does better than the Q-L. Jang et al. [110] propose DeepRL to reduce traffic-jammed areas in a SC. The number of existing vehicles and the average speed of vehicles passing through the intersection were gathered. The DeepRL T is applied to predict the traffic. Result shows that the DeepRL achieves a good performance.

Wu et al. [111] propose DeepRL to improve fuel economy of hybrid electric vehicles. The algorithm was trained with a large amount of driving cycles that was generated from traffic simulation. The result shows that DeepRL performs much better than the conventional reinforcement learning (ConRL) approach. Zhao et al. [112] propose DeepRL to reduce the network congestion and balance the network load for supporting SC services. The DeepRL improves the crowd management of various sections in SC by learning the request distribution pattern of the crowd and select the right decision for it. The DeepRL is found to be better than open shortest path first (OSPF) and enhanced-OSPF (EOSPF)

Ali et al. [31] propose DeepRL-Q-learning (DeepRL-QL) resource allocation for accessing channel in a dense wireless local area network in a SC. The DeepRL uses intelligent Q-learning to learn unknown wireless network platform and resource allocation in small heterogeneous networks. The performance of the DeepRL-QL in terms of throughput, access delay and fairness are better than the media access control (MAC) protocol and binary exponential backoff (BEB) mechanism. Mohammadi et al. [113] propose semi-supervised deep reinforcement learning (SSDRL) model to improve the accuracy of learning agent of indoor localization in smart building of a SC. The SSDRL maximizes supervised learning task using unlabeled data. The SSDRL is applied in classifying the learning agent of indoor localization. The result shows that the SSDRL has higher performance than the supervised DRL (SDRL) model. Table 10 presents the summary of DeepRL applications in SC.

5.9 Other types of deep neural networks

In this section, the ANN that could not be classified in the previous section because of uniqueness is presented in this section. The section comprises of different ANN both deep and shallow. For example, Xu and Gade [115] propose structured deep NN (SD-NN) for the prediction of house prices in SC. The SD-NN is for predictive analytics which contains only the needed connections among neurons and also match with the layered knowledge graph. The SD-NN is applied in predicting property values. The result shows that the SD-NN outperforms conventional multivariate linear regression (CMLR) model. Ghoneim and Manjunatha [117] propose DNN to predict ozone pollution. The DNN is trained with the help of grid search technique. The DNN is applied in predicting the feature level of the ozone pollution. The result shows that the DNN outperforms the SVM, ANN and GLM models. Bura et al. [118] propose DNN to detect the time and place of parking a car. The DNN checks the availability of the parking space. The DNN is applied to classify the time and place of parking a car. The result shows that the DNN has better accuracy than the AlexNet and mAlexNet.

Elsaeidy et al. [119] propose restricted Boltzmann machines (RBM) to detect distributed denial of service (DDoS) attacks in SC applications. The RBMs are applied to detect distributed denial of service attacks. The results obtained indicated that the RBM is accurate in detecting denial of service than the SVM. Li et al. [120] propose intrusion detection system (IDS) based on deep migration learning model (DMLM) (IDS-DMLM) to detect anomalies in the network. The IDS aggregates sensor node by taking the collected data through the wireless transmission while the DMLM unifies the data definition domain and reduces the high-dimensional data. The IDS-DMLM performance is compared with evolutionary learning model (ELM) and BPNN. It was found that the IDS-DMLM is more accurate than ELM and BPNN. Serrano and Bajo [121] propose DNN to predict drug response in cancer and provide automatic notifications of social exclusion in a SC. The DNN is applied for the prediction of the drug response. The result shows that the DNN outperforms the random forest, K-NN, Logistic, C4.5, AdaBoost and RIPPER. Zhang et al. [122] propose DNN based on fine-grained architecture to improve the efficiency of building energy. The DNN was trained with comfort factor of temperature and combined the factor values of fine-grained models to evaluate the comfort score. The fine-grained DNN outperforms the coarse-grained DNN in terms of energy efficiency.

Sun et al. [114] propose sliding window ANN (SWANN) to predict future energy consumption at both household level and the community level in SC. The SWANN is applied in predicting future trend of energy consumption. The result shows that the SWNN is more accurate than the conventional ANN. Gupta et al. [116] propose Hopfield NN (HNN) to predict intelligent control of traffic lights. The HNN provides the optimal sequence for the traffic lights. The HNN is applied in predicting the intelligent control of the traffic lights for traffic management. The HNN outperforms the GA, particle swarm optimization (PSO) and differential evolution (DE). Table 11 presents the summary of the other deep learning and shallow ANN applied for data analytics in SC.

6 Deep learning platforms

The survey on deep learning for the ANN in SC shows different platforms were used both software platforms and hardware platforms. Table 12 presents the summary of the platforms found during the survey. The list of the platforms in Table 12 is not exhausted because we only present those found to be used in the studies surveyed. We only present projects that clearly state the software or hardware used for implementation. It was found that Tensorflow, Keras, MATLAB and Caffe are the most frequently used platforms for implementing deep learning in SC. Nevertheless, many projects did not disclose the platforms used to implement their projects. The major features of the four main platforms are briefly discussed as follows:

6.1 MATLAB for deep learning

The MATLAB for deep learning platform has the ability to build deep learning models with few lines of codes without the pre-requisite requirement to be an expert. The MATLAB for deep learning can create modify and analyze deep learning model based on visual tools and applications. The MATLAB has a tool for pre-processing of data and tool for automatic labeling of images, video and audio. It can also accelerate algorithms on NVIDIA, GPU and datacenters using applications without programming specialization. Collaborative framework of the MATLAB for deep learning allows collaboration with TensorFlow, PyTorch

Table 12 Summary of the implementation tools found in the study

References	Platform	Hardware configuration
Kök et al. [56]	TensorFlow and Keras	Not specified
Struye et al. [77]	Keras, Tensorflow	Not specified
Jain and Shah [84]	MATLAB	Not specified
Giyenko et al. [63]	TensorFlow	Intel i7-4770 CPU, 24 GB of DDR3 RAM and NVIDIA GTX1070
Rajput et al. [64]	Nvidia Tesla K 40	Core i-7 processor, GPU
Sun et al. [114]	MATLAB and Tensorflow	Intel(R) i5-5257U processor (with a clock speed of 2.70 GHz), 8 GB RAM and running OS X El Capitan ver. 10.11.6
		Not specified
Gupta et al. [85]	Not specified	
Liu et al. [65]	Tensorflow	Not specified
Gupta et al. [116]	MATLAB	Not specified
Mohammadi et al. [113]	TensorFlow	Not specified
Aryal and Dutta [99]	eCognition 9.0 software	Not specified
Jiang et al. [67]	Caffe	Not specified
Li et al. [68]	Caffe	Not specified
Bura et al. [118]	Raspberry Pi	Not specified
Mauro et al. [123]	Caffe	12 Gb of DDR5 RAM
Mehmood et al. [88]	Not specified	230 teraflops, comprising 11,904 cores in 496 nodes
Balchandani et al. [89]	Torch and Tensorflow	Not specified
Guo et al. [70]	Tensorflow, Pycharm	TitanX 12 GB memory
Gu et al. [92]	MATLAB	Not specified
Zhang et al. [10, 11]	Theano	CPU core i7 and GPU K1000M
Alahi et al. [81]	Theano	Not specified
Zhu et al. [100]	Caffe models and PyTorch	Intel i7-7700, 3.60 GHz CPU with 32 GB RAM and a NVIDIA GeForce GTX 1080 GPU
Dinh et al. [101]	Not specified	PC with an i7 3.6 GHz CPU and 32 GB of RAM
Heo et al. [109]	Gray-box mode	Not specified
Jang et al. [110]	RL4J in Java-based IDEs	Not specified
Wu et al. [111]	Quadstone Paramics	Not specified
Jindal et al. [102]	MATLAB	Not specified
Chackravarthy et al. [103]	MATLAB	Not specified
Salvador et al. [82]	TensorFlow	NVIDIA GeForce 1080 and Titan X GPUs
Aqib et al. [73]	Keras	Not specified
Elsaeidy et al. [119]	MATLAB	Intel core i5 CPU 2.4 GHz with 16 GB of RAM and Windows 10 (64-bit)
Camero et al. [104]	Python 3, keras and tensorflow	Not specified

Table 12 (continued)

References	Platform	Hardware configuration
Li et al. [120]	MATLAB R2010b	Microsoft Windows 10 system
Serrano and Bajo [121]	Tensorflow	Intel Core Processor CPU with 8 cores, 2297 MHz, and 128 GB of RAM
Yuan et al. [127]	MATLAB	Not specified
Atef and Eltawil [105]	Not specified	Not specified
Zhao et al. [112]	Python	Intel® Xeon®, two processors workstation dual Intel Xeon E5-2630V4 2.2 GHz CPUs, 128 GB random access memory (RAM), Nvidia Titan 12G GPU, and Ubuntu 14.04 64-bit operating system
Zhang et al. [122]	Python 3.6	64-bit Windows 10 Pro, Intel Core i5-7300U
Ali et al. [31]	ns-3 network simulator	Not specified
Camero et al. [83]	Python's scikit-learn library	Not specified
	keras and tensorflow	Not specified

and MxNet. There is reinforcement learning in the MATLAB for simulating and training of a dynamic system. Data set can be generated from the MATLAB for deep learning and Simulink for training and testing the deep learning models to model physical system. In addition, the MATLAB for deep learning support interoperation with deep learning framework is an open source using the ONNX import and export features. The deep learning developed in MATLAB for deep learning can be deployed anywhere including CUDA, C code, cuDNN, TensorRT and ARM compute library. The MATLAB for deep learning covers speech processing, computer vision and reinforcement learning [133]. However, MATLAB for deep learning requires license for each of the modules; it is not an open source software. Cross compiling or the conversion of MATLAB code to other language is difficult and requires deep level of MATLAB coding skills to fix error. Application deployment task is lacking in MATALAB [134].

6.2 Tensorflow

The Tensorflow is a platform that offers a paradigm for deep learning technologies. The Tensorflow is the product of Google developed in 2015. The Tensorflow includes Java, Python, API, C++ and Go. The developers of Tensorflow design it for computation on the flow graph. The nodes of the graph represent the operations, and the graph edges are the multidimensional array commonly referred to as the Tensors. The Tensorflow has compatibility capabilities because it is compatible for computation on multiprocessor systems with either CPU or GPU and it has option of CUDA [135]. However, the TensorFlow has the following disadvantages: It doesn't support GPU other than the Nvidia. The TensorFlow is lagging in terms of computational speed compared to Torch, CNTK, Caffe and Theano. It doesn't support windows environment except installation through Conda or Python and symbolic loop lack support in TensorFlow [136].

6.3 Keras

The Keras is deep learning library written in Python and is a high-level ANN API. The Keras has the capability to run on TensorFlow, Theano and Microsoft cognitive toolkits. It allows fast and easy prototyping because of its friendliness: User experience and centeredness are at the core of Keras. Its API is simple and consistent; with regard to user error, it gives an unambiguous action feedback modularity: It has different modules that can be combined to create a new model of deep learning models. For instance, the Keras has cost function, optimizers, activation functions, regularization, and neural layers each as a module. These modules can be combined to create a new model of deep learning. It supports the combination of ConvNet and RNN as well as running of the constituent algorithms. Extensibility: It allows the addition of new modules in a simple way without difficulties. In addition, new modules can be created easily and this makes Keras suitable for conducting high-level research. As regard to hardware platform, Keras runs on both CPU and GPU systems [137]. However, it lacks flexibility and not optimal for studying new deep learning model. Yet, it is not 100% effective and efficient in multi-GPU environment [138].

6.4 Caffe

The Caffe is a deep learning framework based on C++ and CUDA with capability to operate using the interface of command line, Python and MATLAB. The Caffe runs on mobile platforms and has the ability to be extended to work on Apache Hadoop ecosystem with Spark. Caffe2 is based on the Caffe and is part of Facebook open source project which has additional Python API and runs across different OS platforms, such as Mac, Windows, Linux, IoS, Android [139]. The Caffe2 is a modular framework for deep learning that comes with the capability of training deep learning architectures and deployment. The mobile feature of the Caffes makes it fit to support the new generation hardware. It powers one of the largest mobile deep learning that has more than a billion devices. Facebook supports Caffe2 and PyTorch for variety of artificial intelligence applications. The major focus of the Caffe2 is performance and multi-platform compatibility. On the other hand, the PyTorch mainly focuses on flexibility for the purpose of fast prototype and research. The Caffe2 is now part of PyTorch [140]. However, the Caffe activeness in development has reduced. It only supports C++ when it comes to custom layers. The static model graph of the Caffe doesn't support many of the RNN applications that require inputs with variable size. It is cumbersome to define very large deep model and modular DNN model in Caffe [138].

Different project different hardware and software were used for the implementation; the last column of Table 10 presents the hardware configuration used in each of the projects. The computational time of algorithm is affected by hardware because of the differences in configuration of the hardware. For example, the convergence time of the algorithm that run on CPU system cannot be compared with the convergence time of the same algorithm that run on GPU system despite the fact that the algorithms are the same. This is because the hardware configuration of the two systems differs.

7 The smart city datasets

The purpose of this section is to provide the readers with the sources and nature of the datasets used in different projects. Readers interested in a similar research can easily identify the sources and nature of different SC datasets in proposing a novel deep learning model for further development of the SC concept.

The main factor in data analytics is the dataset. Some projects used real-world datasets while others applied benchmark datasets. Real-world datasets are generated by performing experiment. Benchmark datasets is the dataset extracted from public repository and mostly freely available for public use. Table 13 presents different types of datasets used in different project; only the project that reveled their datasets and source were reported in the table. Most of the project used real-world datasets collected from the SC.

Benchmark datasets are typically clean, very easy to obtain, not expensive to gather and ready to apply algorithm. On the other hand, real datasets are expensive to gather, require data engineering before the application of algorithm, difficult to gather and contain missing points. Algorithms can work perfectly on benchmark datasets, but the performance of the algorithm can be otherwise on real-world datasets because the two categories of the datasets have different behavior and scale. As such, the algorithm intended for application in real-world scenario should be tested on real-world datasets not only on the assumption of the algorithm performance on only benchmark datasets. A good algorithm is expected to have a very good performance on both benchmark and real-world dataset.

It is observed in Table 13 that the main datasets used in the projects includes: Traffic, video, image and text datasets. Brief comments on the different datasets presented in Table 13 are as follows: *Traffic data* Inductive loop for identifying and counting the number of vehicles flowing in the SC in every 5 min. Similarly, pneumatic tube counts the number of vehicles in every 15 min. In each second, GPS in the SC provide information about traffic density, speed of vehicles and direction of movement. The video vehicle detection equipment classified and count vehicles at thirty frames per second. Magnetometer provides information about vehicles estimated speed and traffic density [142]. *Image data* In SC, images can be collected from different sources within the SC. For instance, images can be collected from satellite meant for surveillance and unmanned Aerial vehicle as an autonomous vision system equipped with cameras for collecting images. *Video data* Different sources of video data are available in SC because of the technologies embedded in the SC for different purposes. The SC is equipped with video related technology such as the CCTV, surveillance camera, video vehicle detection, unmanned aerial vehicles and drones. These equipments collect video data within the SC for various purposes such as security, tracking traffic flow. *Textual data* Communication by citizens in SC is in natural language through different emerging technologies. The textual data in SC can be collected from different sources such as the mobile phone short message service, social network such as Twitter, online forum, and online customer conversation. These platforms generate a lot of unstructured text data, the unstructured data need to be structured before processing to provide new information.

Table 13 The summary of the datasets found in different project

References	Dataset	Source
Liang et al. [95]	Caller detail record (CDR)	Mobile users in interested regions
Valipour et al. [43]	PKLot	Federal University of Parana (UFPR) and Pontifical Catholic University of Parana (PUCPR)
Amato et al. [42]	PKLot and CNRPark-EXT	9 Smart cameras
Song et al. [76]	GPS traces of mobile device, CDR data and online social networking data	Transportation mode labels: Stay, Walk, Bicycle, Car, Train
Huang et al. [37]	PeMS data set and Entrance–Exit Station data	Inductive loops, and China entrance-exit of a road segment, respectively
Kök et al. [56]	CityPulse EU FP7 Project	Road traffic, pollution, weather, cultural, library and parking data were collected from the cities of Aarhus and Brasov in Denmark and Romania, respectively
Amato et al. [62]	PKLot and CNRPark	Raspberry smart cameras
Chen et al. [96]	Image datasets	Google earth
Feng [106]	High resolution traffic dataset	BellCore Laboratory
Struye et al. [77]	Raw data dumps	Created from chipset
Jain and Shah [84]	Citypulse and real-time dataset of air pollutant	City of Aarhus, Denmark
Xu and Gade [115]	Real estate data	Zillow.com
Giyenko et al. [63]	CCTV images dataset	surveillance cameras from four different points in South Korea, near the sea side
Rajput et al. [64]	Recorded videos	NCC Lab
Sun et al. [114]	Belgian wind farm data	Advanced metering devices
Pan [78]	Performance Measurement System (PeMS)	Caltrans performance measurement system
Gupta et al. [85]	Travel time records data	GPS-enabled cars, operated by the taxi aggregator company
Qolomany et al. [79]	Wi-Fi network dataset	University of Houston main campus
Liu et al. [65]	Chinese number plate records	Collected through cameras
Han and Sohn [75]	Smart Card	Seoul metropolitan
Kuang et al. [66]	INRIA pedestrian pictures database	PASCAL VOC2007 and 2012
Fu et al. [80]	PeMs dataset	Bay Area, California
Vlahogianni et al. [86]	Network Sensors	Located in the area of Santander
Sharad et al. [87]	GPS data	Not specified
Dambhare and Karale [91]	Twitter review dataset	Restaurant, Hotels and Dominos
Gupta et al. [116]	Network traffic data	Not specified
Zheng et al. [97]	Parking sensor information dataset	Collected from San Francisco and Melbourne
Belhajem et al. [108]	GPU and Odometer data	Institut Pascal Data Sets
Mohammadi et al. [113]	iBeacon RSSI values and smartphone dataset	iBeacons in a campus library
Aryal and Dutta [99]	Geo Ortho-kit imagery dataset	Space Imaging LLC
Jiang et al. [67]	Prior knowledge, like GIS	Not specified
Li et al. [68]	Satellite Image dataset	Shaoshan city
Ghoneim and Manjunatha [117]	Pollution and weather datasets	CityPulse project

Table 13 (continued)

References	Dataset	Source
Bura et al. [118]	PKLot and CNRPark. CNRPark datasets	San Jose State University's parking lot
Mauro et al. [123]	PKLot dataset	Three different parking areas at Catania
Mehmood et al. [88]	Twitter and 50 published online news articles	Well-known newspapers and magazines including Guardian, Fortune, BBC, Wired, and Times
Lwowski et al. [69]	INRIA Person dataset	Collected from GRAZ 01, Personal digital image collections and Google Images
Balchandani et al. [89]	Image and video datasets	City of San Jose
Niu et al. [98]	Taxi trajectories and road network datasets	Wuchang district road network in Wuhan
Guo et al. [70]	2015 Traffic sign recognition competition dataset (TSD-2015) and the 2016 Traffic sign recognition competition dataset (TSD-2016)	Traffic sign detection and recognition competition
Liu et al. [71]	2000 image	UCSD crowd counting
Yan and Yu [93]	Traffic	Not specified
Dogru and Subasi [141]	Traffic	VANET, analyzed through expert system
Hanifah et al. [94]	Tweets	Twitter API
Yeshwanth et al. [72]	Video	induction loops, wireless ground sensors, passive infrared detectors, high resolution camera systems and radars
Zhang et al. [10, 11]	Vehicle images dataset	Qingdao city
Alahi et al. [81]	ETH and UCY dataset	Real-world crowded pedestrians
Zhu et al. [100]	UCF_CC_50 and DH302 datasets	Stadiums, concerts and political rallies
Dinh et al. [101]	Cyclist Detection Benchmark and stereo pedestrian datasets primary particle pollutants dataset	Disparity Map
Heo et al. [109]	Traffic dataset	Korea Environment Institute (Air Korea website)
Jang et al. [110]	Traffic simulation dataset	Not specified
Wu et al. [111]	Open energy information dataset	Traffic sensor
Jindal et al. [102]	Video dataset	Zone PCLO for a day of summer in a smart homes
Chackravarthy et al. [103]	Bus vehicle dataset	Not specified
Salvador et al. [82]	Real-world open road traffic dataset	Shanghai, biggest city in China
Aqib et al. [73]	Smart water dataset	City of UK, Department for Transport
Elsaeidy et al. [119]	29 car parks dataset	smart water distribution plant
Camero et al. [104]	Knowledge Discovery in Databases (KDD) CUP 99 dataset	National Car Parks (NCP) in the City of Birmingham in the U.K.
Li et al. [120]	KDD	sensor data of the Internet of Things system
Serrano and Bajo [121]	wind power generation equipment operation dataset	Oracle object-relational database management system
Yuan et al. [90]	Humans, social network sensor	Not specified
Atef and Eltawil [105]	Real-time electricity prices dataset	Not specified
Zhao et al. [90]	Real-world online network topology	City of Denmark
Zhang et al. [122]	IoT instrumented building dataset	Topology Zoo], NSFCNET in Beijing, China
Ali et al. [31]	Network dataset	office building in Center City Philadelphia, USA
Camero et al. [83]	Real data of electricity consumption 217 paper containers dataset	Not specified in ISEP/IPP, Porto, Portugal, Andalusian city (Spain)

8 General discussion

This section presents general discussion about the survey, the strength of the deep learning in SC, domain of application including taxonomy, publication trend and graphical representation of the algorithms used in the SC.

8.1 Strengths

The survey revealed that the deep learning has being successfully applied in SC for data analytics in different domain of applications. Remarkable achievements were recorded by the deep learning models (see Tables 3, 4, 5, 6, 7, 8, 9, 10, 11). Different architectures of the deep learning were applied to solve machine learning problems such as classification, prediction and clustering. It was found from the survey that the deep learning architecture is powerful, robust and effective in dealing with large volume of data generated from activities within the SC compared to the shallow ANN. It is believed that the deep learning models are the compatible analytic tools suitable for handling data generated from the SC. The deep learning models have been applied for image processing, network traffic management, pollution, parking space, detection of vehicles among many others. The performance of the deep learning models was compared with the conventional ANN and other machine learning algorithms; it was consistently found that the deep learning always performs better than the conventional ANN and other machine learning algorithms. It was observed that deep learning is a very good candidate for data analytics in SC. However, there are few studies that do not present performance comparison in their studies, as such, it is difficult to ascertain the effectiveness of the algorithm propose in their study. The survey indicated that the shallow ANN are still active in SC for data analytics including evolutionary ANN. The ConvNet has an inbuilt data engineering mechanisms—convolution and pooling layers; therefore, ConvNet doesn't require manual data engineering or separate technique for feature selection unlike the shallow ANN. These capabilities of the ConvNet make it suitable for image processing in the SC. There are problems in the SC where the output of the systems depends on the computation previously performed, the DRNN is a perfect match for such problems because the DRNN has the ability to memorize events sequentially. The DBN is generative model with a capacity to solve both supervise and unsupervised learning problems which can make the DBN fit for developing cybersecurity solutions in the SC.

8.2 Domain of application

Figure 9 presents the taxonomy of deep learning application domain in SC. It is found that the deep learning architectures were applied across different domain within the SC. The taxonomy was created based on the survey conducted. In each study, the domain of application in the SC was extracted, compiled and the taxonomy was created to give the readers easy representation of the domains within the SC where deep learning models have been applied for data analytics. The taxonomy can easily allow the reader to know the domain that has not been exploited.

Figure 9 shows the diversity of the domain of deep learning applications in the SC. As shown in Fig. 9, it is clear that transportation in general received more attention on the applications of deep learning compared to other domain in the SC. This is likely because of the significant of transportation systems in supporting innovations and development of economy in a SC. The transportation system eases movement of humans, goods and services. Effective transportation systems encouraged the location of industries, thereby, open job opportunities. The domains that require more investigation on the applications of deep learning models to enhance their effectiveness and efficiency are visually shown in the taxonomy. This can help researchers to easily track the domain that has not received attention or less attention regarding the application of deep learning.

8.3 Publication trend

Recently, almost 6 years, many projects on the application of deep learning and shallow ANN in SC across different domains have been published. The publications trend is depicted in Fig. 10 showing the publication trend from 2014 to 2019. The survey has been conducted up to 2019 as it is the current active year. However, we included future edition, 2020 project to show the activeness of the research area. The data used to depict the publication trend as shown in Fig. 10 are extracted from the number of publications in each year starting from 2014. For example, the number of papers published in 2014 was found to be 4. The summaries of the published works are presented from Sects. 5.1–5.9; going through the summaries on yearly bases it reveals that 4 papers were published in 2014. The same procedure is applied to papers published in 2015 up to 2020. Figure 10 indicates that the application of deep learning in SC is saturated because it shows declined pattern of publishing research project in the research area as observed in the last 2 years. The lowest number of publications was recorded in 2014 and subsequent years recorded more number of publications than 2014, clearly

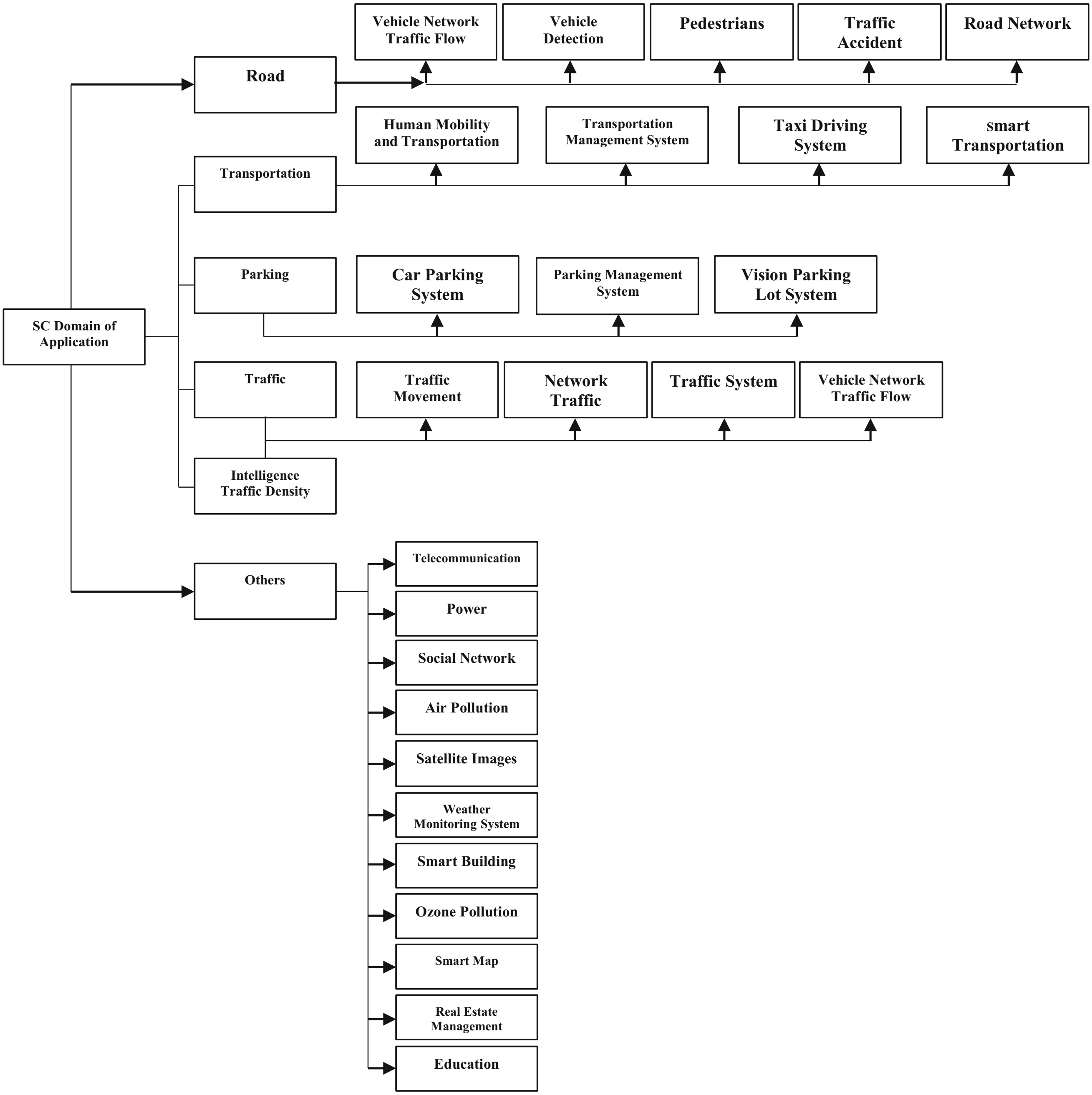

Fig. 9 Taxonomy for the domain of application in SC

indicating acceptability and popularity in the research community. The pattern is expected to continue into the future as SC keeps evolving and more studies are required to fully understand the SC and increase its robustness and efficiency. Already 2020 publication has start springing up. The bars representing the years 2019 and 2020 were marked with different color in view of the fact that publications for these years may still appear.

Figure 11 shows the frequency of different deep learning architectures and the shallow ANN applied in the SC. The data used to depict Fig. 11 is the frequency of each algorithm in the survey extracted from Tables 3, 4, 5, 6, 7, 8, 9, 10 and 11 in Sects. 5.1–5.9. For example, the frequency of applying DeepRL in SC is 5 extracted from Table 10. The same procedure is applied to ConvNet extracted from Table 3, DBN extracted from Table 4, etc. The presence of shallow ANN in the SC indicated that the shallow ANNs are still relevant in solving machine learning problem in the SC as some of the deep learning algorithms were hybridized with the shallow ANN to improve the performance of the deep learning algorithms.

Fig. 10 Publications trend

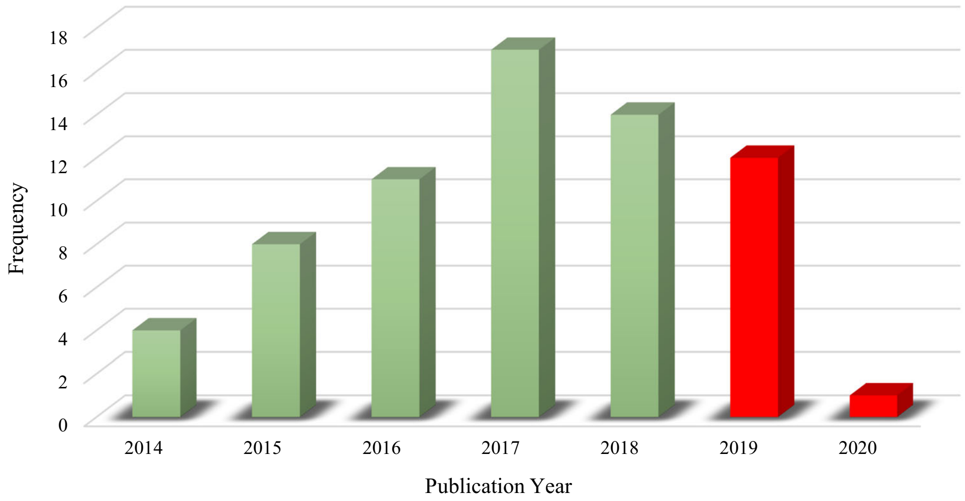

Fig. 11 The deep learning architectures and shallow ANN

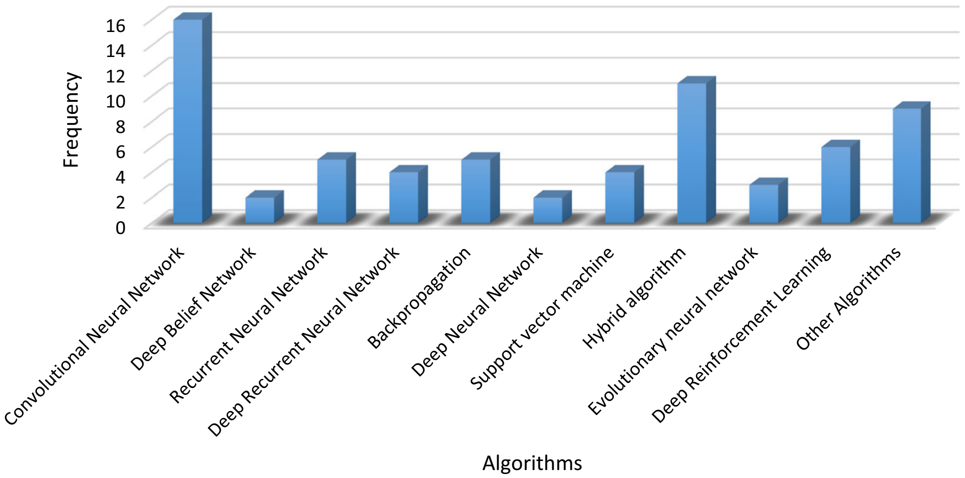

Figure 11 shows that the deep learning architecture gained tremendous attention from the research community possibly because of the advantages of the deep learning over the shallow ANN in data analytics. For example, the shallow ANN require different technique for dimension reduction, whereas the deep learning does not require different technique for dimension reduction as the dimension reduction can be done automatically by the deep learning without requiring extra data engineering effort. The ConvNet is the most popular deep learning algorithm in solving problem in the SC as clearly shown by the longest bar likely because of its robustness and suitability in handling image processing problem. The DeepRL is an emerging deep learning architecture gaining fast popularity in the SC environment with initial appearance in 2018 (see Table 10). The trend is likely to continue in the future with already a future edition—2020 publication. It is indicated that hybrid algorithms are raising and it has the possibility of exceeding the ConvNet in the future because it has been proven from the literature that the hybrid deep learning algorithms are found to perform better than the constituent algorithms in solving SC problems. Figure 11 can help researchers to easily view the unexploited deep learning algorithms in SC and architectures that require more investigation in the SC.

The visualization of the application of different deep learning architectures per year in SC over 6 years (2014–2020) is shown in Fig. 12. The area started gaining attention in 2014 with DBN and hybrid as the only deep learning architecture. An increase in interest of researchers in the area was observed from 2015 onwards as more deep learning architectures were adopted. It is observed that ConvNet has the highest attention in 2017 and 2018. The DBN has least attention and was only adopted in 2014 and 2016. All the deep learning architectures were adopted in 2019 with the exception of DBN. The DRNN seems to maintain consistency of adoption for five consecutive years (2015–2019). The hybrid deep leaning architecture starts

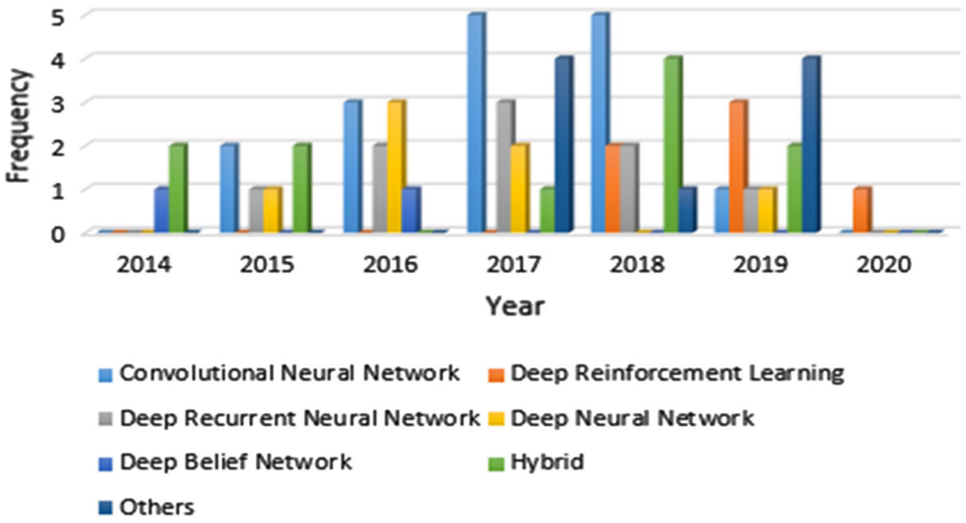

Fig. 12 Frequency of deep learning algorithms in SC per year

attracting attention especially in 2018 probably due to their better performance than single constituent algorithms. The first algorithm to be applied in 2020 is the DRL and hopefully more publications is expected in the future. The visualization in Fig. 12 gives an overview of the rate at which different deep learning algorithms have been applied in SC from 2014 to 2020.

9 Challenges and future research opportunities

The survey clearly indicated some challenges in the applications of deep learning for ANN in SC. This section focuses its discussion on the major challenges found during the survey and propose possible direction for future research.

It was found from the survey that the application of nature inspired algorithms in deep learning for data analytics in SC is an untapped research area. In spite of the fact that the use of nature inspired algorithms such as cuckoo search algorithm, firefly algorithm, harmony search, and particle swarm optimization in optimizing the parameters of the deep learning architecture improves its efficiency and effectiveness, it is yet to witness its application in SC. However, evidence in [12] indicated that the deep learning model optimize through nature inspired algorithms outperform the conventional models of the deep learning in many cases. In the future, researchers are suggested to apply nature inspired algorithm for the optimization of deep learning for ANN in solving problems in SC e.g. classification, clustering, and prediction.

As shown in Sect. 8.2 deep learning for ANN were applied in different domain of the SC. However, the aspect of internet of vehicles within SC remains untapped. The internet of vehicles comprises of fully autonomous vehicles, semi-autonomous vehicles and conventional vehicles equipped with autonomous vehicle facilities. The internet of vehicles generates large volume of data as a result of communications. Such data are required to be analyze to unravel new insight for inform decision and improving the effectiveness of the internet of vehicle technology in the SC. In the future, we propose for researchers to apply deep learning for ANN in the processing of the large-scale data generated from the internet of vehicles within the SC.

Another major issue with the SC is the dynamic nature of technological advancement; as technology evolves, the concept of the SC also evolved. The continue evolving of the SC can pose a challenge to deep learning models because it can create challenges to the deep learning models in handling the new concept of the SC data analytics from the future generation SC. This is possible especially when the deep learning technology remains without being modified to adapt to the constant changes in the concept of the SC. In the future, we suggest researchers to continue to modify the deep learning models as the concept of the SC changes so as to adapt to the dynamic nature of the SC. By so doing, the deep learning can remain relevant in the phase of technological advancement.

Despite the penetration of deep learning in SC and remarkable achievement, other aspect of the deep learning remained unexploited in the SC. Some cases of the deep learning concept not exploited in SC includes but not limited to generative adversarial network, neural abstract machine, memory augmented neural network, interaction based deep network, attentive network and deep extreme learning machine. It will be interesting to exploit these deep learning aspects in the SC to unravel their effectiveness in solving problems in SC. In the future, we enjoin researchers to exploit the outlined unexploited deep learning concepts in the SC.

The tuning of many parameters required by the deep learning still remain an open research problem as it lacks a systematic way of getting the optimal parameter settings. In addition, the performance of the deep learning model depends on the optimal setting of the parameters. Another open problem is the issue of computational time required by the deep learning architectures, the training of the deep learning typically takes long time before convergence. However, there are issues that time is critical in taken decision. For example, in SC the notification of crime to the security agencies, alerting accident to the relevant authority if they occurred, fire outbreak and health related issues, time is highly critical, as any delay can lead to loss of life. In the future, researchers should work on increasing the convergence speed of the deep learning models in solving problems in the SC.

Performance analysis: inconsistency of the deep learning models across different domain of application (see Sect. 8.2) in the SC. The performance of the deep learning differs depending on the domain of application. The need for researchers to know the best deep learning architecture for a particular domain is critical to avoid trial and error that consume a lot of resources and it is tedious. The best deep learning architecture for each domain in the SC remains an open problem. Researchers should perform a rigorous performance analysis for different deep learning architecture in different domain of application to unravel the best deep learning model for each of the SC domain of application.

10 Conclusions

The survey intends to present recent progresses, taxonomy, challenges and future research opportunities on the application of deep learning in SC. In the paper, a dedicated rigorous survey on deep learning in SC was conducted. Recent progress made on the application of deep learning in SC is clearly outlined. Two new taxonomies based on deep learning and domain of applications in the SC were created. Publication trend indicated that this research area is rapidly gaining popularity and the trend is expected into the future with already a future edition appearing in the literature. Generally, ConvNet is highly popular in SC; hybrid algorithms are gaining acceptability over the single constituent algorithms and DeepRL is fast gaining popularity in SC over other single deep learning algorithms with the exception of ConvNet. The challenges currently mitigating against the smooth application of deep learning in SC were pointed out, and suggestion on how to deviate from the challenges pointed in the paper was outlined as opportunities for future research work. The survey is intended for novice researchers to use it as entry point into the research area; experts' researchers can use it as a benchmark for further development of the research area.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Eremia M, Toma L, Mihai S (2017) The smart city concept in the 21st century. Proc Eng 181:15–21
2. Choudhary SK, Sathe RB, Kachare AE (2017) Smart Cities based on Internet of Things (IoT). Int J Eng Trends Technol (IJETT) 48(8):434–439
3. Zhu C, Leung VCM, Shu L, Ngai EC (2015) Green Internet of Things for smart world. IEEE Access 3:2151–2162. https://doi.org/10.1109/ACCESS.2015.2497312
4. Khan Z, Anjum A, Kiani SL (2013) Cloud based big data analytics for smart future cities. In: 2013 IEEE/ACM 6th international conference on utility and cloud computing, pp 381–386. https://doi.org/10.1109/ucc.2013.77
5. Wani A, Ahmad F, Saduf B, Asif A, Khan I (2020) Advances in deep learning. Springer, Singapore
6. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge. https://doi.org/10.4258/hir.2016.22.4.351
7. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
8. Zhang Q, Yang LT, Chen Z, Li P (2018) A survey on deep learning for big data. Inf Fusion 42:146–157
9. Fayek HM, Lech M, Cavedon L (2017) Evaluating deep learning architectures for speech emotion recognition. Neural Netw 92:60–68
10. Zhang W, Chen L, Gong W, Li Z, Lu Q, Yang S (2015) An integrated approach for vehicle detection and type recognition. In: Paper presented at the 2015 IEEE 12th International conference on ubiquitous intelligence and computing and 2015 IEEE 12th International conference on autonomic and trusted computing and 2015 IEEE 15th International conference on scalable computing and communications and its associated workshops (UIC-ATC-ScalCom)
11. Zhang Y, Li X, Zhang Z, Wu F, Zhao L (2015) Deep learning driven blockwise moving object detection with binary scene modeling. Neurocomputing 168:454–463
12. Chiroma H, Gital AY, Rana N, Shafi'i MA, Muhammad AN, Umar AY, Abubakar AI (2020) Nature inspired meta-heuristic algorithms for deep learning: recent progress and novel perspective. In: Paper presented at the advances in computer vision, Cham
13. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford
14. Bello AA, Chiroma H, Gital AY et al (2020) Machine learning algorithms for improving security on touch screen devices: a survey, challenges and new perspectives. Neural Comput Appl. https://doi.org/10.1007/s00521-020-04775-0
15. Chiroma H, Abdullahi UA, Alarood AA, Gabralla LA, Rana N, Shuib L, Hashem IA, Gbenga DE, Abubakar AI, Zeki AM, Herawan T (2018) Progress on artificial neural networks for big data analytics: a survey. IEEE Access 7:70535–70551

16. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J, Chen T (2018) Recent advances in convolutional neural networks. Pattern Recognit 77:354–377
17. Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J (2001) Gradient flow in recurrent nets: the difficulty of learning long-term dependeces. In: Kremer SC, Kolen JF (eds) A field guide to dynamical recurrent neural networks. IEEE Press, New York, pp 1–15
18. Pascanu R, Gulcehre C, Cho K, Bengio Y (2013) How to construct deep recurrent neural networks. arXiv:1312.6026
19. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527
20. Al Nuaimi E, Al Neyadi H, Mohamed N, Al-Jaroodi J (2015) Applications of big data to smart cities. J Internet Serv Appl 6(1):25
21. Hinton GE (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14(8):1771–1800
22. François-Lavet V, Henderson P, Islam R, Bellemare MG, Pineau J (2018) An introduction to deep reinforcement learning. Found Trends Mach Learn 11(3–4):219–354
23. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016). Asynchronous methods for deep reinforcement learning. In International conference on machine learning, pp 1928–1937
24. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529
25. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: David ER, James LM, Group CPR (eds) Parallel distributed processing: explorations in the microstructure of cognition, vol 1. MIT Press, Cambridge, pp 318–362
26. Hanson SJ, Pratt LY (1986) Comparing biases for minimal network construction with back-propagation. Bell communications research Morristown, pp 177–185
27. Kristensen T, Patel R (2003) Classification of eukaryotic and prokaryotic cells by a backpropagation network. In: Paper presented at the proceedings of the international joint conference on neural networks, 2003
28. Zweiri YH, Whidborne JF, Seneviratne LD (2003) A three-term backpropagation algorithm. Neurocomputing 50:305–318
29. Siegelmann HT, Sontag ED (1991) Turing computability with neural nets. Appl Math Lett 4(6):77–80. https://doi.org/10.1016/0893-9659(91)90080-F
30. Vapnik V (1999) An overview of statistical learning theory. IEEE Trans Neural Netw 10(5):988–999
31. Ali R, Zikria YB, Kim B-S, Kim SW (2020) Deep reinforcement learning paradigm for dense wireless networks in smart cities. In: Al-Turjman F (ed) Smart cities performability, cognition, & security. Springer, Cham, pp 43–70
32. Vapnik V (2013) The nature of statistical learning theory. Springer, New York
33. Feurer M, Hutter F (2019) Hyperparameter optimization. In: Hutter F, Kotthoff L, Vanschoren J (eds) Automated machine learning. Springer, Cham, pp 3–33
34. Bergstra JS, Bardenet R, Bengio Y, Kégl B (2011) Algorithms for hyper-parameter optimization. In: Shawe-Taylor JR, Zemel RS, Bartlett PL, Pereira F, Weinberger KQ (eds) Advances in neural information processing systems. Curran Associates Inc., Granada, pp 2546–2554
35. Domhan T, Springenberg JT, Hutter F (2015) Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In: Twenty-fourth international joint conference on artificial intelligence
36. Baldominos A, Saez Y, Isasi P (2018) Evolutionary convolutional neural networks: an application to handwriting recognition. Neurocomputing 283:38–52
37. Huang W, Song G, Hong H, Xie K (2014) Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans Intell Transp Syst 15(5):2191–2201. https://doi.org/10.1109/TITS.2014.2311123
38. Liu K, Zhang LM, Sun YW (2014) Deep Boltzmann machines aided design based on genetic algorithms. In: Yarlagadda P, Kim Y-H (eds) Applied mechanics and materials, vol 568. Trans Tech Publications Ltd, pp 848–851
39. Papa JP, Rosa GH, Costa KA, Marana NA, Scheirer W, Cox DD (2015) On the model selection of Bernoulli restricted boltzmann machines through harmony search. In: Proceedings of the companion publication of the 2015 annual conference on genetic and evolutionary computation, pp 1449–1450
40. Hutter F, Lücke J, Schmidt-Thieme L (2015) Beyond manual tuning of hyperparameters. KI-Künstl Intell 29(4):329–337
41. Ilievski I, Akhtar T, Feng J, Shoemaker CA (2017) Efficient hyperparameter optimization for deep learning algorithms using deterministic RBF surrogates. In: Thirty-first AAAI conference on artificial intelligence
42. Amato G, Carrara F, Falchi F, Gennaro C, Meghini C, Vairo C (2017) Deep learning for decentralized parking lot occupancy detection. Expert Syst Appl 72:327–334. https://doi.org/10.1016/j.eswa.2016.10.055
43. Valipour S, Siam M, Stroulia E, Jagersand M (2016) Parking-stall vacancy indicator system, based on deep convolutional neural networks. In: Paper presented at the 2016 IEEE 3rd world forum on Internet of Things (WF-IoT), pp 655–660
44. Mynhoff PA, Mocanu E, Gibescu M (2018) Statistical learning versus deep learning: performance comparison for building energy prediction methods. In: IEEE PES innovative smart grid technologies conference Europe (ISGT-Europe), pp 1–6
45. Wang D (2014) Deep learning in speech and language processing. http://166.111.134.19:7777/wangd/talks/pdf/deeplearning.pdf. Accessed 2 June 2019
46. Van der Veen J, Willems S, Deschuymer S, Robben D, Crijns W, Maes F, Nuyts S (2019) Benefits of deep learning for delineation of organs at risk in head and neck cancer. Radiother Oncol 138:68–74
47. Albino V, Berardi U, Dangelico R (2015) Smart cities: definitions, dimensions, performance, and initiatives. J Urban Technol 22:1723–1738
48. Rathore MM, Ahmad A, Paul A, Rho S (2016) Urban planning and building smart cities based on the Internet of Things using big data analytics. Comput Netw 101:63–80. https://doi.org/10.1016/j.comnet.2015.12.023
49. Giffinger R, Pichler-Milanović N (2007) Smart cities: ranking of European medium-sized cities. Centre of Regional Science, Vienna University of Technology, pp 7–25
50. Sullivan F (2008) Strategic opportunity analysis of the global smart city market smart city market is likely to be worth a cumulative $1.565 trillion by 2020. The Growth Partnership Company, vol 19, pp 1–19
51. Nam T, Pardo T (2011) Conceptualizing smart city with dimensions of technology, people, and institutions, pp 1–5
52. Zygiaris S (2013) Smart city reference model: assisting planners to conceptualize the building of smart city innovation ecosystems. J Knowl Econ 4(2):217–231
53. Bergh J, Viaene S (2015) Unveiling smart city implementation challenges: the case of Ghent. Inf Polity 21:20–25
54. Bakıcı TY, Almirall E, Wareham J (2012) A smart city initiative: the case of Barcelona. J Knowl Econ 4:23–30
55. Rathore MM, Ahmad A, Paul (2013) IoT-based smart city development using big data analytical approach, pp 1–8

56. Kök İ Şimşek MU, Özdemir S (2017) A deep learning model for air quality prediction in smart cities. In: Paper presented at the 2017 IEEE international conference on big data (big data), pp 1983–1990
57. Iftikhar N, Liu X, Nordbjerg FE, Danalachi S (2016) A prediction-based smart meter data generator. In 2016 19th international conference on network-based information systems, pp 173–180. https://doi.org/10.1109/nbis.2016.15
58. Kitchin RJG (2014) The real-time city? Big data and smart urbanism. GeoJournal 79(1):1–14
59. Sood SK, Sandhu R, Singla K, Chang V (2018) IoT, big data and HPC based smart flood management framework. Sustain Comput Inform Syst 20:102–117
60. Krieg J-G, Jakllari G, Toma H, Beylot A-LJP, Computing M (2018) Unlocking the smartphone's sensors for smart city parking. Pervasive Mob Comput 43:78–95
61. Hashem IAT, Chang V, Anuar NB, Adewole K, Yaqoob I, Gani A, Ahmed E, Chiroma H (2016) The role of big data in smart city. Int J Inf Manag 36(5):748–758
62. Amato G, Carrara F, Falchi F, Gennaro C, Vairo C (2016) Car parking occupancy detection using smart camera networks and deep learning. In: Paper presented at the 2016 IEEE symposium on computers and communication (ISCC), pp 1–6
63. Giyenko A, Palvanov A, Cho Y (2018) Application of convolutional neural networks for visibility estimation of CCTV images. In: Paper presented at the 2018 international conference on information networking (ICOIN), pp 875–879
64. Rajput NS, Mishra A, Sisodia A, Makarov I (2018) A novel autonomous taxi model for smart cities. In: Paper presented at the 2018 IEEE 4th world forum on Internet of Things (WF-IoT), pp 625–628
65. Liu J, Li X, Zhang H, Liu C, Dou L, Ju L (2017) An implementation of number plate recognition without segmentation using convolutional neural network, pp 246–253
66. Kuang Ping, Ma Tingsong, Li Fan, Chen Ziwei (2018) Real-time pedestrian detection using convolutional neural networks. Int J Pattern Recognit Artif Intell. https://doi.org/10.1142/S0218001418560141
67. Jiang Q, Cao L, Cheng M, Wang C, Li J (2015) Deep neural networks-based vehicle detection in satellite images. In: Paper presented at the 2015 international symposium on bioelectronics and bioinformatics (ISBB)
68. Li P, Zang Y, Wang C, Li J, Cheng M, Luo L, Yu Y (2016) Road network extraction via deep learning and line integral convolution. In: Paper presented at the 2016 IEEE international geoscience and remote sensing symposium (IGARSS)
69. Lwowski J, Kolar P, Benavidez P, Rad P, Prevost JJ, Jamshidi M (2017) Pedestrian detection system for smart communities using deep convolutional neural networks. In: Paper presented at the 2017 12th system of systems engineering conference (SoSE)
70. Guo J, Lu J, Qu Y, Li C (2018) Traffic-sign spotting in the wild via deep features. In: Paper presented at the 2018 IEEE intelligent vehicles symposium (IV)
71. Liu S, Zhai S, Li C, Tang J (2017) An effective approach to crowd counting with CNN-based statistical features. In: Paper presented at the 2017 international smart cities conference (ISC2)
72. Yeshwanth C, Sooraj PA, Sudhakaran V, Raveendran V (2017) Estimation of intersection traffic density on decentralized architectures with deep networks. In: Paper presented at the 2017 international smart cities conference (ISC2)
73. Aqib M, Mehmood R, Albeshri A, Alzahrani A (2018) Disaster management in smart citiesby forecasting traffic plan using deep learning and GPUs, pp 1–17
74. Huh JH, Seo K (2019) Artificial intelligence shoe cabinet using deep learning for smart home, pp 825–834. https://doi.org/10.1007/978-981-13-1328-8_108
75. Han G, Sohn K (2016). Clustering the Seoul metropolitan area by travel patterns based on a deep belief network. In: 2016 3rd MEC international conference on big data and smart city, pp 1–6
76. Song X, Kanasugi H, Shibasaki R (2017) DeepTransport: prediction and simulation of human mobility and transportation mode at a citywide level. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence (IJCAI-16), Center for spatial information science, The University of Tokyo, Japan, pp 2618–2624
77. Struye J, Braem B, Latré S, Marquez-Barja J (2018) The CityLab testbed—large-scale multi-technology wireless experimentation in a city environment: neural network-based interference prediction in a smart city. In: Paper presented at the IEEE INFOCOM 2018—IEEE conference on computer communications workshops (INFOCOM WKSHPS), pp 529–534
78. Pan YTL (2015) Predicting short-term traffic flow by long short-term memory recurrent neural network. In: 2015 IEEE international conference on Smart City/SocialCom/SustainCom together with DataCom 2015 and SC2 2015, pp 153–158. https://doi.org/10.1109/smartcity.2015.63
79. Qolomany B, Al-Fuqaha A, Benhaddou D, Gupta A (2017) Role of deep LSTM neural networks And Wi-Fi networks in support of occupancy prediction in smart buildings. In: 2017 IEEE 19th international conference on high performance computing and communications; IEEE 15th international conference on smart city; IEEE 3rd international conference on data science and systems, pp 50–57. https://doi.org/10.1109/hpcc-smartcity-dss.2017.7
80. Fu R, Zhang Z, Li (2016) Using LSTM and GRU neural network methods for traffic flow prediction, pp 324–328
81. Alahi A, Goel K, Ramanathan V, Robicquet A, Fei-Fei L, Savarese S (2016) Social LSTM: human trajectory prediction in crowded spaces. In: Paper presented at the 2016 IEEE conference on computer vision and pattern recognition (CVPR)
82. Salvador MM, Budka M, Quay T (2018) Automatic transport network matching using deep learning. Transp Res Procedia 31:67–73. https://doi.org/10.1016/j.trpro.2018.09.053
83. Camero A, Toutouh J, Ferrer J, Alba E (2019) Waste generation prediction in smart cities through deep neuroevolution. Springer, Cham, pp 192–204. https://doi.org/10.1007/978-3-030-12804-3-15
84. Jain R, Shah H (2016) An anomaly detection in smart cities modeled as wireless sensor network. In: Paper presented at the 2016 international conference on signal and information processing (IConSIP), pp 1–5
85. Gupta A, Bansal A, Gupta R, Naryani D, Sood A (2017). Urban waterlogging detection and severity prediction using artificial neural networks. In: 2017 IEEE 19th international conference on high performance computing and communications; IEEE 15th international conference on smart city; IEEE 3rd international conference on data science and systems, pp 42–49. https://doi.org/10.1109/hpcc-smartcity-dss.2017.6
86. Vlahogianni E, Kepaptsoglou K, Tsetsos V, Karlaftis M (2015) A real-time parking prediction system for smart cities. J Intell Transp Syst 20:1–29
87. Sharad S, Sivakumar PB, Narayanan VA (2016) The smart bus for a smart city—a real-time implementation. Department of Computer Science and Engineering Amrita School of Engineering, Coimbatore Amrita Vishwa Vidyapeetham, Amrita University, India, pp 1–6
88. Mehmood R, Alam F, Albogami NN, Katib I, Albeshri A, Altowaijri SM (2016) UTiLearn: a personalised ubiquitous

teaching and learning system for smart societies. IEE Access. https://doi.org/10.1109/access.2017.2668840

89. Balchandani C, Hatwar RK, Makkar P, Shah Y, Yelure P, Eirinaki M (2017) A deep learning framework for smart street cleaning. In: Paper presented at the 2017 IEEE third international conference on big data computing service and applications (BigDataService)
90. Wang H, Li L, Pan P, Wang Y, Jin Y (2019) Online detection of abnormal passenger out-flow in urban metro system. Neurocomputing 359:327–340. https://doi.org/10.1016/j.neucom.2019.04.075
91. Dambhare SS, Karale PSJ (2017) Smart map for smart city. In: International conference on innovative mechanisms for industry applications (ICIMIA 2017). IEEE, pp 622–626
92. Gu P, Khatoun R, Begriche Y, Serhrouchni A (2017), Support vector machine (SVM) based sybil attack detection in vehicular networks. In: Paper presented at the 2017 IEEE wireless communications and networking conference (WCNC)
93. Yan H, Yu D (2017) Short-term traffic condition prediction of urban road network based on improved SVM. In: Paper presented at the 2017 international smart cities conference (ISC2)
94. Hanifah R, Supangkat SH, Purwarianti A (2014) Twitter information extraction for smart city. In: Paper presented at the 2014 international conference on ICT for smart society (ICISS)
95. Liang VC, Ma RT, Ng WS, Wang L, Winslett M, Wu H, Ying S, Zhang Z (2016) Mercury: metro density prediction with recurrent neural network on streaming CDR data. In: Paper presented at the 2016 IEEE 32nd international conference on data engineering (ICDE), pp 1374–1377
96. Chen X, Xiang S, Liu C, Pan C (2014) Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geosci Remote Sens Lett 11(10):1797–1801. https://doi.org/10.1109/LGRS.2014.2309695
97. Zheng Y, Rajasegarar S, Leckie C (2015) Parking availability prediction for sensor-enabled car parks in smart cities. In: 2015 IEEE tenth international conference on intelligent sensors, sensor networks and information Processing (ISSNIP) Singapore, 7–9 April 2015, pp 1–6
98. Niu X, Zhu Y, Zhang X (2014) DeepSense: a novel learning mechanism for traffic prediction with taxi GPS traces. In: Paper presented at the 2014 IEEE global communications conference
99. Aryal J, Dutta R (2015). Smart city and geospatiality: hobart deeply learned. In: Paper presented at the 2015 31st IEEE international conference on data engineering workshops
100. Zhu J, Feng F, Shen B (2018) People counting and pedestrian flow statistics based on convolutional neural network and recurrent neural network. In: Paper presented at the 2018 33rd youth academic annual conference of Chinese association of automation (YAC)
101. Dinh TT, Vinh ND, Wook JJ (2018) Robust pedestrian detection via a recursive convolution neural network. In: Paper presented at the 2018 19th IEEE/ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD)
102. Jindal A, Aujla GS, Kumar N, Prodan R, Obaidat MS (2018) DRUMS: demand response management in a smart city using deep learning and SVR. In: Paper presented at the 2018 IEEE global communications conference (GLOBECOM)
103. Chackravarthy S, Schmitt S, Yang L (2018) Intelligent crime anomaly detection in smart cities using deep learning. In: Paper presented at the 2018 IEEE 4th international conference on collaboration and internet computing (CIC)
104. Camero A, Toutouh J, Stolfi DH, Alba E (2019) Evolutionary deep learning for car park occupancy prediction in smart cities. In: Paper presented at the learning and intelligent optimization, Cham
105. Atef S, Eltawil AB (2019) A comparative study using deep learning and support vector regression for electricity price forecasting in smart grids. In: Paper presented at the 2019 IEEE 6th international conference on industrial engineering and applications (ICIEA)
106. Feng G (2015) Network traffic prediction based on neural network. In: Paper presented at the 2015 international conference on intelligent transportation, big data and smart city, pp 527–530
107. Lei Y, Shangzheng L (2015) Simulationstudy of genetic algorithm optimized neural network controller. In: 2015 International conference on intelligent transportation, big data & smart city, pp 721–724. https://doi.org/10.1109/icitbs.2015.182
108. Belhajem I, Maissa YB, Tamtaoui A (2016) A robust low cost approach for real time car positioning in a smart city using extended kalman filter and evolutionary machine learning, pp 806–811
109. Heo S, Nam K, Loy-Benitez J, Li Q, Lee S, Yoo C (2019) A deep reinforcement learning-based autonomous ventilation control system for smart indoor air quality management in a subway station. Energy Build 202:109440. https://doi.org/10.1016/j.enbuild.2019.109440
110. Jang I, Kim D, Lee D, Son Y (2018) An agent-based simulation modeling with deep reinforcement learning for smart traffic signal control. In: Paper presented at the 2018 international conference on information and communication technology convergence (ICTC)
111. Wu Y, Tan H, Peng J, Zhang H, He H (2019) Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series-parallel plug-in hybrid electric bus. Appl Energy 247:454–466. https://doi.org/10.1016/j.apenergy.2019.04.021
112. Zhao L, Wang J, Liu J, Kato N (2019) Routing for crowd management in smart cities: a deep reinforcement learning perspective. IEEE Commun Mag 57(4):88–93. https://doi.org/10.1109/MCOM.2019.1800603
113. Mohammadi M, Al-Fuqaha A, Guizani M, Oh J (2018) Semisupervised deep reinforcement learning in support of IoT and smart city services. IEEE Internet Things J 5(2):624–635. https://doi.org/10.1109/JIOT.2017.2712560
114. Sun L, Hu J, Liu Y, Liu L, Hu S (2017) A comparative study on neural network-based prediction of smart community energy consumption. In: paper presented at the 2017 ieee smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pp 1–8
115. Xu H, Gade A (2017) Smart real estate assessments using structured deep neural networks. In: Paper presented at the 2017 IEEE SmartWorld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), p 1-7
116. Gupta V, Kumar R, Reddy KS, Panigrahi BK (2017) Intelligent traffic light control for congestion management for smart city development. In: 2017 IEEE region 10 symposium (TENSYMP), pp 1–5
117. Ghoneim OA, Manjunatha BR (2017) Forecasting of ozone concentration in smart city using deep learning. In: Paper presented at the 2017 international conference on advances in computing, communications and informatics (ICACCI)
118. Bura H, Lin N, Kumar N, Malekar S, Nagaraj S, Liu K (2018) An edge based smart parking solution using camera networks and deep learning. In: Paper presented at the 2018 IEEE international conference on cognitive computing (ICCC)

119. Elsaeidy A, Munasinghe KS, Sharma D, Jamalipour A (2019) Intrusion detection in smart cities using Restricted Boltzmann Machines. J Netw Comput Appl 135:76–83. https://doi.org/10.1016/j.jnca.2019.02.026
120. Li D, Deng L, Lee M, Wang H (2019) IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning. Int J Inf Manag 49:533–545. https://doi.org/10.1016/j.ijinfomgt.2019.04.006
121. Serrano E, Bajo J (2019) Deep neural network architectures for social services diagnosis in smart cities. Future Gen Comput Syst 100:122–131. https://doi.org/10.1016/j.future.2019.05.034
122. Zhang W, Hu W, Wen Y (2019) Thermal comfort modeling for smart buildings: a fine-grained deep learning approach. IEEE Internet Things J 6(2):2540–2549. https://doi.org/10.1109/JIOT.2018.2871461
123. Mauro DD, Moltisanti M, Patanè G, Battiato S, Farinella GM (2017) Park smart. In: Paper presented at the 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)
124. Anwar SM et al (2018) Medical image analysis using convolutional neural networks: a review. J Med Syst 42(11):226
125. Dedinec A et al (2016) Deep belief network based electricity load forecasting: an analysis of Macedonian case. Energy 115:1688–1700
126. Mishra BK et al (2020) A novel application of deep learning with image cropping: a smart city use case for flood monitoring. J Reliable Intell Environ 6(1):51–61
127. Yuan Z, Wang W, Fan X (2019) Back propagation neural network clustering architecture for stability enhancement and harmonic suppression in wind turbines for smart cities. Comput Electr Eng 74:105–116. https://doi.org/10.1016/j.compeleceng.2019.01.006
128. Zamil MGA et al (2019) Multimedia-oriented action recognition in Smart City-based IoT using multilayer perceptron. Multimedia Tools Appl 78(21):30315–30329
129. Pan X et al (2019) Prediction of network traffic of smart cities based on DE-BP neural network. IEEE Access 7:55807–55816
130. Hossain MS et al (2019) Smart healthcare monitoring: a voice pathology detection paradigm for smart cities. Multimedia Syst 25(5):565–575
131. Lee C et al (2020) Spike-FlowNet: event-based optical flow estimation with energy-efficient hybrid neural networks. arXiv preprint arXiv:2003.06696
132. Sokolov S et al (2020) Hybrid neural networks in cyber physical system interface control systems. Bull Electr Eng Inform 9(3):1268–1275
133. MathWorks (2019) MATLAB for deep learning. https://www.mathworks.com/solutions/deep-earning.html?s_tid=hp_brand_deeplearning. Retrieved on 10 Oct 2019
134. Innat M (2018) Preferable tools for machine learning—Python—MatLab—R. https://www.codementor.io/innat_2k14/preferable-tools-for-machine-learning-python-matlab-r-jfozzpphz. Retrieved on 16 Oct 2019
135. Tensorflow (2017) An open-source software library for machine intelligence (Online). https://www.tensorflow.org/
136. Dataflair Team (2018) TensorFlow pros and cons—the bright and the dark sides. https://data-flair.training/blogs/tensorflow-pros-and-cons/. Retreived on 16 Oct 2019
137. Keras (2017) Keras: the python deep learning library. https://keras.io/. Retrieved on 16 Oct 2019
138. Nguyen G, Dlugolinsky S, Bobák M, Tran V, García ÁL, Heredia I, Malík P, Hluchý L (2019) Machine Learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artif Intell Rev 52(1):77–124
139. Hatcher WG, Yu W (2018) A survey of deep learning: platforms, applications and emerging research trends. IEEE Access 6:24411–24432
140. Caffe2 (2018) Caffe2 and PyTorch join forces to create a Research + Production platform PyTorch 1.0. https://caffe2.ai/blog/2018/05/02/Caffe2_PyTorch_1_0.html. Retrieved on 16 Oct
141. Dogru N, Subasi A (2018) Traffic accident detection using random forest classifier. In: Paper presented at the 2018 15th learning and technology conference (L&T)
142. Shukla SN, Champaneria TA (2017) Survey of various data collection ways for smart transportation domain of smart city. In: 2017 International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE, pp 681–685