State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems

Zubair Md. Fadlullah, *Senior Member, IEEE*, Fengxiao Tang, *Student Member, IEEE*, Bomin Mao, *Student Member, IEEE*, Nei Kato, *Fellow, IEEE*, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani, *Member, IEEE*

***Abstract*—Currently, the network traffic control systems are mainly composed of the Internet core and wired/wireless heterogeneous backbone networks. Recently, these packet-switched systems are experiencing an explosive network traffic growth due to the rapid development of communication technologies. The existing network policies are not sophisticated enough to cope with the continually varying network conditions arising from the tremendous traffic growth. Deep learning, with the recent breakthrough in the machine learning/intelligence area, appears to be a viable approach for the network operators to configure and manage their networks in a more intelligent and autonomous fashion. While deep learning has received a significant research attention in a number of other domains such as computer vision, speech recognition, robotics, and so forth, its applications in network traffic control systems are relatively recent and garnered rather little attention. In this paper, we address this point and indicate the necessity of surveying the scattered works on deep learning applications for various network traffic control aspects. In this vein, we provide an overview of the state-of-the-art deep learning architectures and algorithms relevant to the network traffic control systems. Also, we discuss the deep learning enablers for network systems. In addition, we discuss, in detail, a new use case, i.e., deep learning based intelligent routing. We demonstrate the effectiveness of the deep learning-based routing approach in contrast with the conventional routing strategy. Furthermore, we discuss a number of open research issues, which researchers may find useful in the future.**

***Index Terms*—Machine learning, machine intelligence, artificial neural network, deep learning, deep belief system, network traffic control, routing.**

I. Introduction

RECENTLY, the rapid development of the current Internet and mobile communications industry has contributed to increasingly large-scale, heterogeneous, dynamic, and systematically complex networks [1]–[3]. The core networks have grown substantially larger as greater switching capacities are introduced in the Internet core and more, bigger routers with more/faster radio links are deployed in the wireless enterprise backbone networks. Such complex network systems confront a myriad of challenges including management, maintenance, and network traffic optimization [4]. Furthermore, most of these packet-switched networks are experiencing a sharp growth in data traffic owing to the rapid development of mobile user equipment, social networking applications and services, and so forth. The existing network policies are not adequate to adapt to the continually changing network conditions arising from the explosive traffic growth. In these years of tremendous growth in the network traffic, while the network operators frequently expressed concern regarding declining profits [5], it is almost the perfect time to rethink how the network traffic control can be improved. Therefore, incorporating intelligence into network traffic control systems can play a significant role in guaranteeing Quality of Service (QoS) in Internet Protocol (IP)-based networks [6]. Over the past few decades, machine learning (ML) has been exploited to intelligently dictate traffic control in wired/wireless networks [7]–[10]. Since the early excitement stirred by machine intelligence in the 1950s, smaller subsets of machine intelligence have been impacting a myriad of applications over the last three decades as shown in Fig. 1. Recently, an even smaller subset of Machine Intelligence and ML techniques, known as deep learning, has emerged with the potential of creating even larger disruptions [11]–[13]. Notice from Fig. 1 that our focus is different from the traditional one as we aim to investigate how the state-of-the-art deep learning applications may disrupt computer networks, particularly the network traffic control systems. In order to understand why deep learning systems are anticipated to replace their predecessors (i.e., conventional ML techniques), refer to the various types of ML techniques (supervised, unsupervised, or reinforcement learning) and different algorithms in Fig. 2, which may be used to implement the intelligent decision-making for network traffic control systems. Among the ML techniques, both supervised and unsupervised Artificial Neural Networks (ANNs) have been exploited in a variety of networking fields, ranging from routing to intrusion detection. While the conventional, shallow ANNs have frequently been utilized for traffic prediction for proactive network management, their performance is practically

Manuscript received December 19, 2016; revised April 16, 2017; accepted May 12, 2017. Date of publication May 23, 2017; date of current version November 21, 2017. *(Corresponding author: Nei Kato.)*

Z. M. Fadlullah, F. Tang, B. Mao, and N. Kato are with the Graduate School of Information Sciences, Tohoku University, Sendai 9808579, Japan (e-mail: zubair@it.is.tohoku.ac.jp; fengxiao.tang@it.is.tohoku.ac.jp; bomin.mao@it.is.tohoku.ac.jp; kato@it.is.tohoku.ac.jp).

O. Akashi, T. Inoue, and K. Mizutani are with the Nippon Telegraph and Telephone Corporation, Network Innovation Laboratories, Kanagawa 239-0847, Japan (e-mail: akashi.osamu@lab.ntt.co.jp; inoue.takeru@lab.ntt.co.jp; mizutani.kimihiro@lab.ntt.co.jp).

Digital Object Identifier 10.1109/COMST.2017.2707140

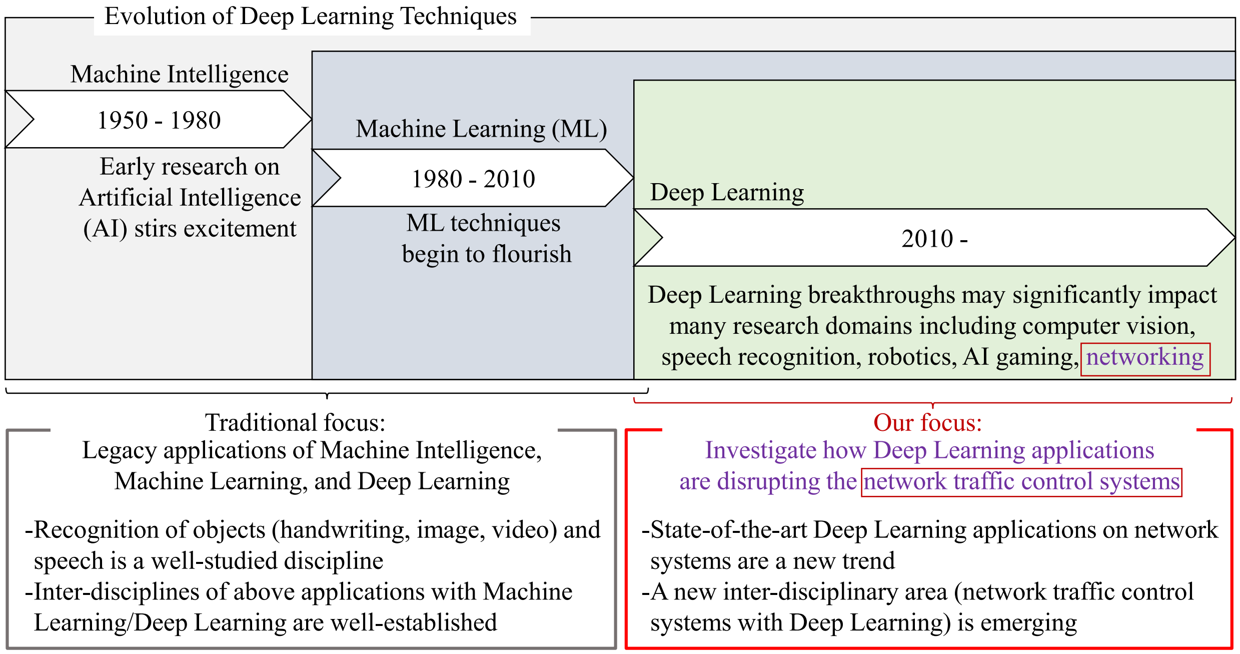


Fig. 1. Evolution of deep learning from conventional Machine Intelligence and Machine Learning paradigms. Our focus, in this paper, compared to the traditional one, is on exploring how deep learning applications are disrupting network traffic control systems.

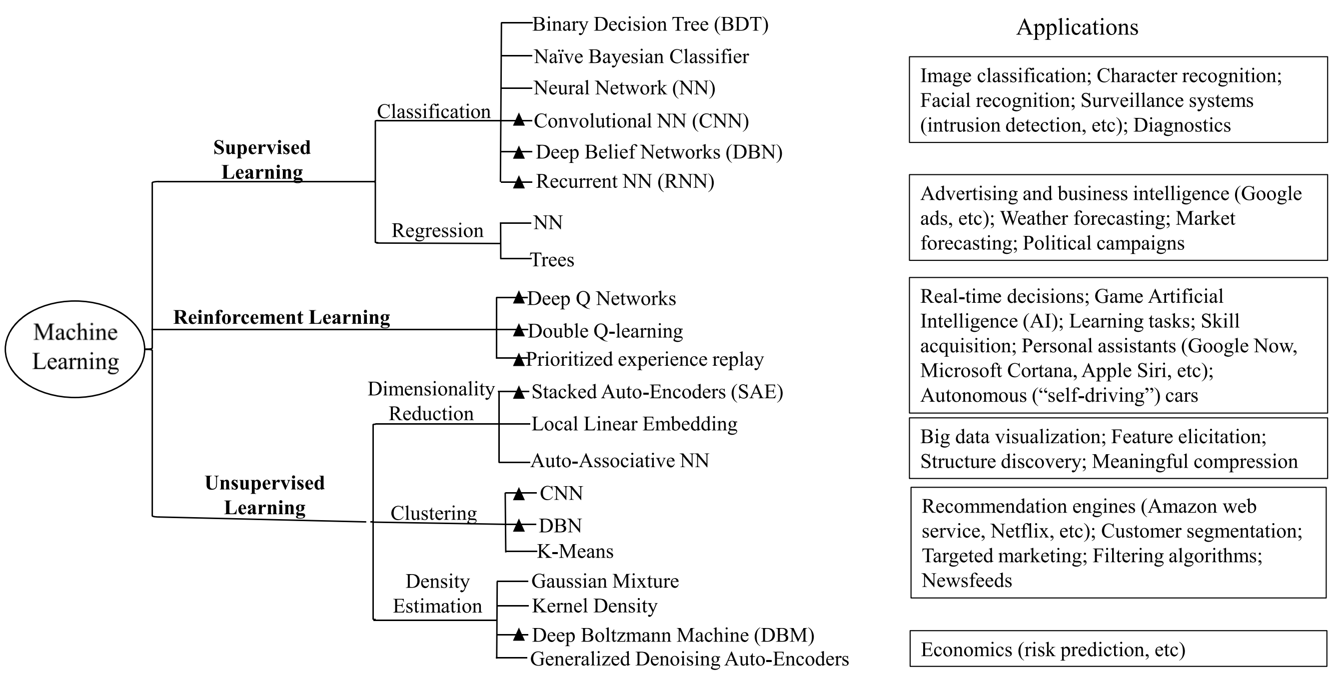


Fig. 2. Various Machine Learning techniques exploited for solving a myriad of computer science problems. It may be noticed that deep learning techniques which are shown with ▲ labels have emerged recently with their use mainly restricted to objects recognition and have not been applied to intelligent network traffic control systems extensively.

limited [6], [14], [15]. This limitation arises due to the fact that increasing the number of hidden layers of the ANNs do not have an impact on the performance pertaining to improved decisions on network-operations (e.g., scheduling, routing, and so forth). Very recently, however, there has been a breakthrough in the way deep learning systems such as Deep Belief Networks, Deep ANNs, and Deep Boltzmann Machines can lead to significant performance gain [16]–[19]. However, the applications of such deep learning systems have been limited mainly to image/character/pattern recognition and natural language processing. Inspired by the way these deep learning systems work to provide much better performance in contrast with the contemporary ML algorithm, we conduct a survey on the state-of-the-art deep learning techniques for intelligent network traffic control. The contributions of our paper are as follows. First, we provide an overview of the state-of-the-art deep learning techniques. Second, we identify the various network traffic control themes where deep learning techniques have been applied in a rather scattered manner. In our survey, for each of these networking themes, we provide a brief overview of the traditional ML techniques, and then describe how the deep learning methodology can achieve much-improved performance in contrast with the traditional ML approaches. Third, among the various networking themes, we stress on a new deep learning application for intelligent routing operations of a backbone network by providing detailed system model of the deep neural network computational model, step by step training and execution details, and experimental results. We also show how the deep learning based intelligent routing technique can outperform the conventional routing strategies such as the Open Shortest Path First (OSPF) routing strategy. Fourth, we provide an elaborate discussion on various open research issues related to deep learning applications in networking problems.

The remainder of the paper is organized as follows. In Section II, we provide an overview of the state-of-the-art deep learning techniques which may be useful for network traffic control systems. In Section III, the deep learning enablers for network systems are discussed. Several commercially available deep learning platforms are also described in the section. Next, in Section IV, the state-of-the-art deep learning applications in various networking related systems are extensively surveyed. In Section V, a new application of deep learning in the network traffic control system, i.e., deep learning based routing, is discussed. The open research issues related to network-centric deep learning applications are discussed in Section VI. Finally, the paper is concluded in Section VII.

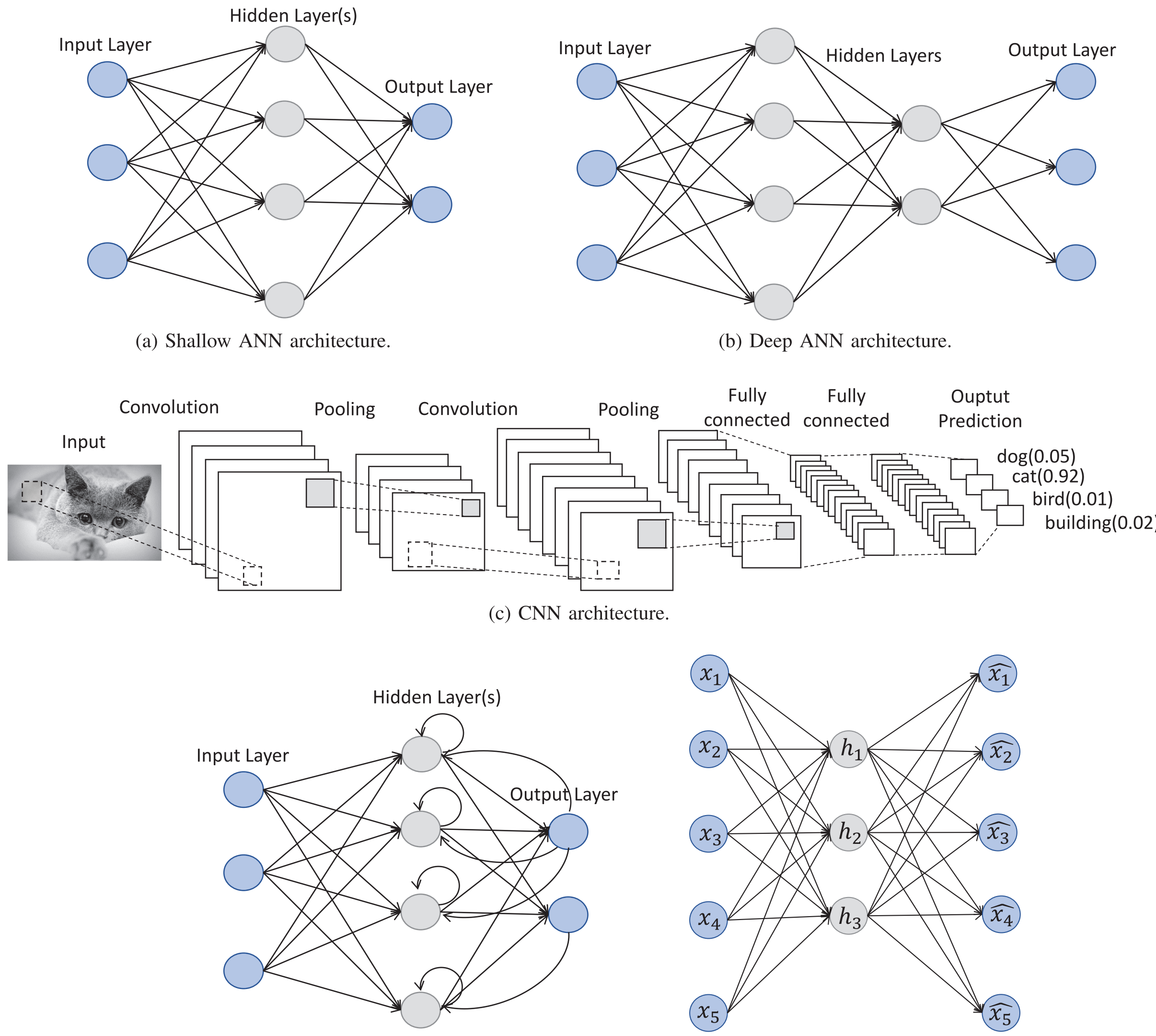
II. Overview of Deep Learning Architectures and Applications

Deep learning is a branch of ML based on a set of algorithms, which construct computational models aiming to represent high-level data abstractions. In the literature, deep learning has also been referred to as deeply structured learning, hierarchical learning, deep feature learning, and deep representation learning. One of the most common and popular deep learning structures is the multiple-layered models of inputs, commonly known as deep neural networks, which comprise multiple levels of non-linear operations. Other deep learning architectures also exist in [20]. Prior to 2006, searching the parameter space of the deep architecture was a formidable research challenge. Recently, the deep learning algorithms have significantly solved this problem [21]. In this section, we first present a brief overview of the state-of-the-art deep learning architectures and algorithms, followed by their applications in several prominent fields to highlight the research gap between deep learning applications in network traffic control systems in contrast with other disciplines.

A. Deep Learning Architectures

Deep learning architectures are bio-inspired in the sense that brains have a deep architecture [22], [23], and computers can emulate such deep architectures. Human brains organize their concepts in a hierarchical fashion. For instance, the human brain first learn simpler concepts, and then combine them to represent more abstract ideas. Motivated by this learning technique, researchers have devoted a lot of efforts for using many levels of abstraction and processing to solve computational problems. Depending on how the deep architectures are intended for use, they can be broadly categorized into three types, namely, generative, discriminative, and hybrid deep architectures. A generative deep architecture aims to characterize the high-order correlation properties of the input data for synthesis purposes. On the other hand, a discriminative deep architecture is used for pattern classification or recognition purposes. By combining the generative and discriminative deep architectures, a hybrid model may be constructed, particularly to carry out discrimination tasks, which are aided by the optimized outputs obtained from the generative architecture [20]. It is worth noting that the hybrid deep architecture is not the same as feeding the outputs of a traditional neural network to a Hidden Markov Model (HMM) [24]–[26]. However, regardless of their purpose, with the exception of Convolution Neural Networks (CNNs), deep architectures could not be successfully trained before 2006. The state-of-the-art deep learning algorithms, on the other hand, still hinge upon multi-layer architectures according to the work conducted by Bengio *et al.* [27]. The key difference is the introduction of Greedy Layer-Wise unsupervised pre-training, which aims to learn a hierarchy of features from a massive, unlabeled dataset, one level at a time (a use example is described in Section II-A3). In other words, in each level, a new transformation of the previously learned features is performed that is served as the input to the next level [28]–[31]. The features obtained through the process can, then, be used as the input to either a standard supervised ML predictor (e.g., Support Vector Machines (SVMs), Conditional Random Field (CRF), and so forth) or to a deep supervised neural network. Since we are interested in the deep learning architectures, consider the latter. Each iteration of unsupervised feature learning creates a layer of weights to the deep neural network. At the final stage, the set of layers comprising the learned weights can be used to initialize a deep supervised predictor (e.g., a neural network classifier or a deep generative model like a Deep Boltzmann Machine (DBM) [18]). For interested readers, we briefly overview the relevant deep learning models such as deep neural networks, deep reinforcement learning, stacked auto-encoders, DBMs, and deep neural networks.

1) Convolutional Neural Networks (CNNs): The Convolutional Neural Network, also known as CNN or Convnet, is a discriminative deep architecture [32]. At the first glance, a CNN may appear to be quite similar to an ordinary ANN since both architectures consist of neurons having learnable (i.e., tunable) weights and biases. Readers familiar with a traditional feed-forward ANN may recall that it receives an input (i.e., a single vector) and transforms the input through a number of hidden layers as depicted in



(a) Shallow ANN architecture.

(b) Deep ANN architecture.

(c) CNN architecture.

(d) RNN architecture.

(e) Auto-Encoder.

Fig. 3. The architectures of shallow (i.e., regular), deep, convolutional, and recurrent neural networks.

Fig. 3(a). Each hidden layer comprises a set of learning units called neurons. The neurons of a hidden layer are connected in a fully meshed fashion with those of the previous layer. The last fully connected layer is referred to as the output layer. Each neuron receiving several inputs takes a weighted sum of them, passes it through an activation function, and responds with an output. In contrast with the traditional feed-forward neural networks, i.e., a shallow ANN where the input is typically a vector, a deep ANN architecture is shown in Fig. 3(b). The deep ANN structure is a conventional training method in many areas, but with full connectivity between nodes suffer from the burden of dimensionality, when the input becomes too large and complex such as the training of high resolution images, the conventional ANN cannot process well. Therefore, the convolution layers were proposed instead of full connectivity in the neural network layers. The architecture of a deep CNN is shown in Fig. 3(c) that deals with a multi-channeled image as the input. To deal with this complex input, the CNN consists of several layers of convolutions with nonlinear activation functions to compute the output. As a consequence, the CNN comprises localized connections whereby each region of the input is connected to a neuron in the output. Each layer applies different filters (in the order of hundreds to thousands) and combines their results. In addition, the CNN consists of the pooling layers for subsampling. During the training phase, a CNN automatically learns the values of its filters based on the given task. For instance, for classifying images, assume that raw pixels, e.g., the image of a cat as shown in Fig. 3(c) are the input of a CNN. In its first layer, the CNN may learn to detect the edges from the raw pixels. Then, in the second layer, the CNN employs the edges to detect simple shapes. In this fashion, in the subsequent (i.e., higher) layers, by using these shapes, the CNN may be able to learn higher-level features like facial shapes and so forth. In the final layer, a classifier is used to exploit these high-level features. Deep CNNs have proved to be quite successful in learning task specific features, which have provided much improved results, particularly on different computer vision tasks, in contrast with contemporary ML techniques. Generally, the CNNs are trained by employing supervised learning methods whereby a large number of input-output pairs are essential. However, obtaining a substantially large training set has been a key challenge in applying CNNs for solving a new task. In addition, CNNs have been reported to be extensively used with unsupervised methods [33]–[35].

2) Recurrent Neural Networks (RNNs) or Long Short Term Memory Networks (LSTMs): The Recurrent Neural Network (RNN) can be considered to be a deep generative architecture [36] as shown in Fig. 3(d). The depth of an RNN may be as large as the length of the input data sequence. Therefore, the RNN is particularly useful for modeling the sequence data in text and speech. Despite the potential strength, their use was restricted until recently due to the so-called "vanishing gradient" problem [37]. New optimization methods to train generative RNNs that modify stochastic gradient descent have appeared in the literature recently [38]–[40].

3) Stacked Auto-Encoder: Stacked auto-encoders can be a good example to demonstrate how the earlier-mentioned Greedy Layer-Wise unsupervised pre-training can be exploited. An auto-encoder refers to an ANN aimed to learn efficient coding [41] by encoding a set of data as depicted in Fig. 3(e). The encoded data conveys a compressed representation of the data set. In other words, the auto-encoder can be exploited to perform data compression or dimensionality reduction. The architecture of a typical auto-encoder is as follows. There is an input layer followed by a number of significantly smaller hidden layers, and finally an output layer. The hidden layers encode the input data set while the output layer attempts to reconstruct the input layer. According to Bourlard and Kamp [42], if the auto-encoder architecture consists of just linear neurons or just a single sigmoid hidden layer, the optimal solution obtained by the auto-encoder is correlated to the Principal Component Analysis (PCA). Then, the feature which is learned by the auto-encoder is used to train another layer of auto-encoder, and so forth. The learned weights, via this process, are eventually used to initialize a deep neural network [43].

4) Deep Boltzmann Machines: The next deep architecture worth mentioning is the Deep Boltzmann Machines or the DBMs. However, readers may first need to briefly review the shallow Boltzmann machine architectures, i.e., the basic Boltzmann machines and the restricted Boltzmann machines [44].

A Boltzmann machine (BM), is a network of binary, stochastic units with an "energy" defined for the network. A typical BM architecture is depicted in Fig. 4(a). While learning is ineffective in a shallow BM, it can be made quite efficient in an architecture called the Restricted Boltzmann Machine (RBM), which does not permit the connections among units on the same layer (refer to Fig. 4(b)). After training one RBM, the activities of its hidden units can be considered as data for training a higher-level RBM. This method of stacking RBMs permits training many layers of hidden units in an efficient manner, and this consists in one of the most common deep learning strategies. As each new layer is added the overall generative model of the RBM improves significantly. In addition, RBMs [45] offer a popular architecture to carry out the pre-training. For instance, Hinton and Salakhutdinov [28], considered a stack of RBMs whereby the learned feature activations of a single RBM are served as the input data for training the following layer of RBM. Once the pre-training is completed, the RBMs are unfolded to construct a deep network that is fine-tuned by employing the back-propagation algorithm [46]. The RBMs stack is also referred to as the Deep Boltzmann machines(DBMs) or Deep Belief Networks (DBNs). According to [28] and [47], the pre-trained DBM is used to initialize a deep neural network and train with backpropagation similar to that in the stacked auto-encoder as explained in Section II-A3.

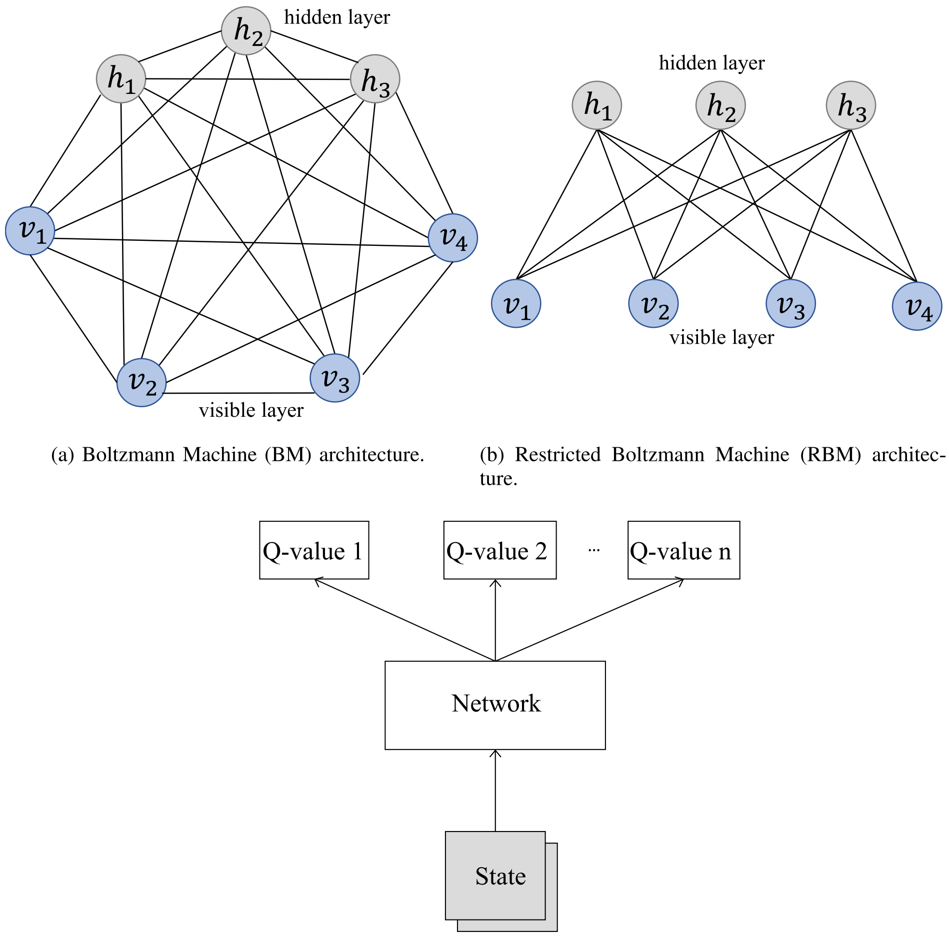
5) Deep Reinforcement Learning: Reinforcement learning combines the strength of both supervised and unsupervised learning methods. In a reinforcement learning method, sparse and time-delayed labels (referred to as "rewards") are used based on which the agent has to learn how to behave in a given environment. In other words, reinforcement learning enables an "agent" to learn by interacting with its environment. In this manner, the agent continues to build upon its experience so as to maximize its long-term rewards. The most well-known reinforcement learning technique is Q-learning [48], a simplified architecture of which is presented in Fig. 4(c). Recently, deep Q networks, patented by Google, have emerged to represent the Q-function of the Q-learning with deep neural network architectures [49]. Many improvements to the deep Q networks have recently been proposed that include Double Q-learning [50], prioritized experience replay [51], dueling network architecture [52], the extension to continuous action space [53], and so forth.

B. Significance of Deep Architectures

A key reason to use the afore-mentioned deep architectures is because they can more efficiently represent a non-linear function [21] compared to the shallow ML architectures. In other words, for a non-linear function to be compactly represented, a significantly large (i.e., deep) architecture is required. Another reason of using deep architectures is that they can support transfer learning, i.e., the ability of a learning algorithm to share knowledge across various tasks. Because deep learning algorithms learn features which capture underlying factors, they may be useful for carrying out other tasks. This idea of knowledge sharing was demonstrated in [27]. In addition, in the two transfer learning challenges held in 2011, learning algorithms leveraging deep architectures exhibited superior performance compared to existing learning algorithms [54], [55]. Furthermore, the works conducted by Glorot *et al.* [56] and Chen and Liu [57] demonstrated the successful application of deep learning architectures in fields related to transfer learning such as domain adaptation.

C. Deep Learning Applications

In recent years, deep learning research has gained a remarkable momentum in both academia and industry. In particular, deep learning techniques had a terrific impact on several



(a) Boltzmann Machine (BM) architecture.

(b) Restricted Boltzmann Machine (RBM) architecture.

(c) Deep Q-networks.

Fig. 4. Simplified architectures of BM, RBM, and deep Q-networks. Note that the BMs and RBMs can be used to construct the deep BM and the DBN, respectively.

computer science and engineering fields including object recognition, speech recognition, natural language processing, robotics, driverless cars and AI gaming. The state-of-the-art deep learning applications in these areas are briefly discussed below.

1) Object Recognition: Object recognition has been regarded as a non-trivial task for computers [60], [61]. The Mixed National Institute of Standards and Technology (MNIST) database [62] comprising a training set of 60,000 examples and a test set of 10,000 examples of handwritten digits have frequently been considered as the benchmark for many prominent ML techniques. Until 2006, SVMs were known to be the most suitable among the contemporary ML methods to recognize the handwritten digits available in the MNIST database. Then, Hinton and Salakhutdinov [28], Srivastava *et al.* [63], and Bengio [21] demonstrated that deep learning algorithms are able to outperform the SVMs by a substantial margin for carrying out the object recognition tasks. Rifai *et al.* [64] and Ciresan *et al.* [65] demonstrated further effective applications of deep learning systems to solve MNIST digit image classification problems with 0.81% and 0.27%, respectively. In addition, researchers have shown that deep learning can be applied for more sophisticated object recognition in natural images, e.g., those available in the ImageNet dataset [66]. In particular, Krihevsky *et al.* [33] demonstrated that deep learning algorithms can improve the state-of-the-art error rate in image recognition from 26.1% to 15.3%. As a practical example, with which the readers may be easily able to connect, consists of Google's deep learning project of integrating image recognition ability with the Google Photos [67].

2) Speech Recognition and Signal Processing: There exist many applications of convolutional neural networks for speech recognition and signal processing in [68]–[70]. The recent research in deep learning architectures and algorithms have revived the interest in neural networks and representation learning, particularly in the speech recognition area [71]–[73]. In addition to research endeavors in academia, industry-based researchers are also devoting a lot of effort toward deep learning applications for speech recognition and signal processing. For instance, the Microsoft Audio Video Indexing Service (MAVIS) speech system based on deep learning showed a significant drop in the error rate compared to contemporary ML techniques (e.g., Gaussian mixtures for the acoustic modeling) used for recognizing speech in a dataset comprising audio records over 300 hours [74]. Despite their good performance, the input vectors used in the state-of-the-art deep neural network are of fixed dimension. Therefore, such a deep architecture acts a static classifier that may not be suitable for speech sequence recognition. This is because the dimensionality of inputs and/or outputs may be variable in sequence recognition. On the other hand, the HMM, based on dynamic programming operations, is particularly useful to model speech sequence data with variable length. Therefore, to exploit their respective advantages, researchers [75] have considered the joint use of the static classifier (i.e., the deep neural network) and the HMM. Furthermore, deep RNNs and CNNs have also been utilized to deal with the afore-mentioned dimensionality problem involving the inputs and/or outputs [58], [76], [77]. These methods were applied to several musical datasets, and the results demonstrated a relative error improvement from 5% to 30% compared to the existing polyphonic transcription method [39], [64]. Furthermore, a practical example of the deep learning application for speech recognition can be found in Apple's deep neural networks-based upgrade to the smart assistant "Siri", which significantly improved its accuracy [78].

3) Natural Language Processing: Natural Language Processing, frequently referred to as NLP in the literature, is another domain where both state-of-the-art ML techniques and deep learning algorithms have been reported to be extensively utilized. For instance, the concept of symbolic data representation through distributed representation was introduced way back in 1986 [79]. However, the idea of using deep architectures to represent symbolic data, also known as "word embedding", was successfully realized only in 2003 via statistical language modeling [80]. The practical application was, however, not possible, until 2008 when Collobert and Weston [81] exploited deep CNNs to carry out the word embedding task. Furthermore, they demonstrated the transfer learning capability of deep architectures through the SENNA system [59] to share representations across various NLP tasks such as named-entity recognition, learning language models, chunking, part-of-speech tagging, semantic role-labeling, and so forth. While the language model recognition was shown to require labeled data for training, the training of the other tasks could be carried out by adopting unsupervised methods. On the other hand, conventional NLP approaches heavily depend upon supervised methods, i.e., manual extraction of features from the language dataset that are passed to a shallow classification algorithm (e.g., an SVM with a linear kernel). Thus, the deep learning paradigm significantly differs from the state-of-the-art NLP approaches, and exhibits much superior performance compared to the shallow architectures using supervised methods for complex tasks. In addition, deep RNNs applications have also appeared in the recent literature to successfully address various aspects of NLP. For example, Socher *et al.*, in their works in [76] and [82], exploited RNNs for carrying out sentiment analysis for semantic compositionality and parsing, respectively. Furthermore, RNN models were used to achieve improved word prediction accuracy [83] using the Wall Street Journal benchmark task [84] and to carry out statistical machine translation [85]. On the other hand, other deep learning structures have also been used to solve NLP problems. For instance, recursive Auto-Encoders were utilized by Socher *et al.* [77] to significantly improve full sentence paraphrase detection. Also, Google's deep learning initiative is worth-noting for the speech recognition capabilities of Google Translate [86].

4) Robotics: In recent years, there has been a growing interest in using deep learning architectures in advanced robotics [87], [88]. For example, deep neural networks as function approximators in reinforcement learning for physical systems like robots were used in [89]. CNNs exploiting RGB-D data have been used by a plethora of robotics applications, e.g., object detection, scene semantic segmentation, and grasping [90]. The work in [91] proposed a deep learning architecture to detect the palm and fingertip positions of stable grasps by employing partial object views. The deep architecture was shown to be particularly useful for generalizing grasp experiences for both known and novel objects. Furthermore, a novel vision-based robotic grasping system based on a max-pooling CNN (MPCNN) appeared in [92]. On the other hand, a new real-time deep learning algorithm for dynamically training several robots was presented in [93]. The real-time learning of high-dimensional features for robotics applications via deep learning techniques is another important topic, which was discussed by [94]. In addition, other topics in robotics such as obstacle detection [95] and context-dependent social mapping [96] are also being addressed by researchers through deep learning methods.

5) Automated Vehicles: Automated vehicles, also referred to as self-driving vehicles or driverless cars, are becoming popular due to the advancement of autonomous navigation and situational awareness systems. Such systems require the integration of numerous sensors on board the vehicle that makes it challenging for the ML algorithms to provide real-time driving decisions [97]. As a consequence, deep neural networks are recently being utilized for analyzing the multi-modal sensor inputs [13], [17], [98], [99] for autonomous vehicles.

6) AI Games: In addition to the afore-mentioned applications, deep learning algorithms are extensively used for playing AI games. For instance, the game of "Go", with an enormous search space and an incredibly high number of board positions/moves, deep neural networks harnessing the combined strength of supervised learning from human experts and reinforcement learning, the AlphaGo program [100] achieved a 99.8% winning rate against other state-of-the-art ML-driven Go programs. Also, for the first time in history, AlphaGo defeated the champion Go player that attests to the success of the underlying deep learning method. The winning streak of AlphaGo against human champions all over the world attracted a lot of media attention. As a result, there is a huge interest among researchers in utilizing the deep learning architectures and algorithms to solve a myriad of AI gaming challenges [101], which were considered to be benchmark problems in the recent past.

D. Research Gap - Deep Learning Implication for Computer Networks

As discussed in the earlier segment of this section, it is evident that the deep learning applications are continuing to span across a wide range of topics in computer science and engineering. From the networking domain, the use of deep learning architectures and algorithms for solving various problems of communication networks have not been formally surveyed. In other words, recently, a number of research works have exploited deep learning methodologies in networking themes. However, they have been rather scattered in the literature. The implication of using deep learning for computer networks is enormous in the sense that this may create a new interdisciplinary research area. In contrast with the other computer science areas such as object recognition, speech recognition, NLP, and so forth, why exactly the deep learning applications have not received a holistic attention from the network researchers? What existing networking applications can be complemented by the deep learning techniques? How to characterize the networking metrics to be measured and predicted using deep learning techniques? What applications exist in computer vision, speech recognition, robotics, and so forth, that may also be analogously applicable to the network traffic control systems? Hence, it is important to thoroughly review which networking areas are currently overlapping with the deep learning space. To the best of our knowledge, no prior research work has surveyed the deep learning applications on computer networks, particularly network traffic control systems. Therefore, in this paper, we stress on surveying the state-of-the-art deep learning applications for network traffic control systems. However, before delving into the survey, we need to understand the key deep learning enablers for network systems which are discussed in the following section.

III. Deep Learning Enablers for Network Traffic Control Systems

In this section, we discuss the key deep learning enablers for network traffic control systems. In order to ultimately provide intelligent network traffic control, i.e., the so-called Knowledge Defined Networking [102], a joint consideration of Software Defined Networking (SDN), network analytics, and deep learning is essential. In today's highly networked society, vast amounts of data are generated, which can be utilized to effectively train the deep learning systems. Due to the fact that the conventional ANNs and other ML techniques are not inherently efficient and/or scalable enough to cope with the large volume data, the networking community did not pursue their use in the long run. If sufficient labeled training data is available, the deep learning techniques could be exploited to explicitly predict the traffic delivery routes and make intelligent decisions on scheduling, bandwidth reservation, and so forth. Nowadays, the network datasets, comprising information such as inbound traffic patterns history, packet drops, failure of network nodes, and so forth, are much bigger compared to the past. If deep learning could bring the improvements in performance to network traffic control systems that it has brought to computer vision, speech recognition, and the other are as mentioned in Section II-C, it could, in essence, enable the Knowledge Defined Networking.

Another deep learning enabler for the network traffic control systems is the better algorithms that have appeared in the recent literature that include stochastic gradient descent (RMSprop, Adagrad, Adam, and so forth), efficient regularizers (e.g., Dropout, L1/L2, and so on), Greedy Layer-Wise training, and so forth. In order to support these much-improved learning algorithms, the recent advancement in commodity hardware (i.e., in terms of processing, memory, and Input/Output bus speeds) is also acting as a key enabler for leveraging deep learning architectures for network traffic control systems. For instance, the Graphics Processing Units (GPUs) with low-latency shared memory system provide a massive computation platform. This is anticipated to drive the next generation Software Defined Routers (SDRs) in a significant manner.additionally, benefiting from the hardware innovation, recent GPU-based SDRs are reported to have a much improved packets processing capability. Furthermore, the GPUs on an SDR may be particularly useful for executing deep learning algorithms for learning various networking situations and acting according to the acquired knowledge. Therefore, the grounds for using deep learning algorithms for network control traffic systems are, now, ready. In addition, technology giants including Google, Microsoft, Apple, Facebook, Nvidia, Amazon, and so on are heavily investing on GPU-accelerated deep learning initiatives with their own deep learning platforms. For instance, Google's open-source platforms called TensorFlow and DeepMind (which was used to successfully train the earlier-mentioned AlphaGo program in Section II-C6), Microsoft's Computational Network Toolkit (CNTK), Facebook's DeepText, Nvidia's DGX-1 and Amazon's Deep Scalable Sparse Tensor Network Engine (DSSTNE) are GPUs to parallelize the deep network training [103]. Broadly speaking, the deep neural networks in these libraries carry out the training phase via data and model parallelism. In other words, a massive dataset is split into batches, which are distributed to parallel models executing on separate hardware in order to conduct simultaneous training. For interested readers, the prominent platforms available for driving real-time packets processing capabilities in SDRs are described in the remainder of this section.

A. TensorFlow

Google's TensorFlow is an open-source interface for accessing the state-of-the-art ML and deep learning algorithms. It permits the use of both CPUs and GPUs in a scalable manner. In other words, it can be used by individual users (e.g., on smart phones/tablets) and also by large-scale distributed systems. TensorFlow has been used for conducting research and deploying deep learning systems into production in the applications described in Section II-C. However, to the best of our knowledge, the TensorFlow API has not been used for developing SDRs. Perhaps, the reason behind this is the cloud-based execution of the deep learning algorithms in TensorFlow, which may not fit for the SDR needs.

B. Torch

Based on the Lua programming language, Torch offers several machine/deep learning algorithms with fast and efficient GPU support. The interesting feature of Torch is the way it can be embedded in mobile operating systems and FPGA backends. Torch has already been used within Facebook, Google, Twitter, IDIAP, and several other social messengers and research hubs. However, its use for network traffic control systems is rather limited, similar to the case of TensorFlow.

C. Caffe

The Convolutional Architecture for Fast Feature Embedding, referred to as Caffe, provides the researchers with a robust framework of state-of-the-art deep learning algorithms. Currently, Caffe drives a number of startup prototypes in vision, speech, and multimedia. It is anticipated that it may be able to power large-scale industrial applications including network traffic control systems.

D. Deeplearning4j

Deeplearning4j (DL4J) offers an open source neural network library for Java and Scala programming languages. This distributed deep learning framework integrates well with Apache Hadoop [104] and Spark [105] using an arbitrary number of CPUs and/or GPUs. In terms of speed, its performance is comparable with that of Caffe exploiting multiple GPUs. On the other hand, it exhibits better performance in contrast with Google's Tensorflow or Torch.

E. WILL

WILL is a high performance deep learning framework [106], which is supported by C++ and compatible to other interfaces of C, Python and assembly language. It has comparable learning performance and speed with Caffe and DL4J, and is more convenient in configuration. Furthermore, WILL is a cross-platform framework, suitable for windows, Unix and even embedded systems. The work in [107] demonstrated the first ever deep learning based routing strategy based on WILL.

F. CUDA

Nvidia's CUDA offers a parallel computing platform and programming model based on GPUs. The deep neural network library in CUDA comprises a library of primitives for the standard deep learning routines, e.g., forward and backward convolution, pooling, normalization, activation of layers, and so forth. It provides one of the fastest state-of-the-art deep learning libraries for deep CNNs and RNNs. In addition, it is one of the top ranking image-processing benchmarks conducted by Chintala *et al.* [108]. An interesting feature of CUDA-based SDRs is the feature of writing the deep learning code directly in the high-level languages (e.g., C/C++ and so forth) and sending it to the GPU without the need of assembly language programming. The NVIDIA Titan X comprising 3584 CUDA cores and 12GB of on-board GDDR5X memory is anticipated to efficiently parallelize the training process of the deep neural networks. Such GPUs are being used to drive real-time image/packets processing capabilities [109].

It is evident from the discussion in this section that currently researchers have access to a wide range of deep learning libraries, which they are able to apply to solve the problems of various research domains. However, the deep learning applications in network traffic control systems have not been surveyed as extensively as those in other research domains. In the following section, we identify the various networking areas which have witnessed applications of deep learning.

IV. Applications of Deep Learning in Network Related Areas

In this section, we first identify a number of network related systems where state-of-the-art deep learning has been applied, namely wireless ad hoc and sensor networks, network traffic classification, origin flow prediction, social networks, mobility prediction, and cognitive radio networks. Then, for each of the mentioned areas, we briefly discuss the shortcomings of the existing ML approaches and then survey the relevant deep learning applications. In addition, for each of the aforementioned networking areas, we provide an insightful discussion on the relevant deep learning applications.

A. Deep Learning in Wireless Sensor Networks

Over the decades, the Wireless Sensor Networks (WNSs) [110]–[117] have enjoyed a plethora of ML applications [118]. Statistical learning algorithms such as Bayesian inference [119] and Gaussian Process Regression (GPR) [120] have been used to the limited extent for various purposes in the WSNs. While they require a significantly small number of training samples, these methods require accurate statistical knowledge and are, therefore, not widely adopted in the WSN context [118]. In particular, reinforcement learning, neural networks, and decision trees have been popular ML algorithms used in WSNs, which consist of typically many autonomous, tiny, low power, and low-cost sensor nodes to collect various types of information (e.g., thermal, acoustic, pressure, chemical, and other ambient data). For instance, supervised learning methods to address localization and objects targeting in WSNs have been extensively employed in the works conducted by [121]–[123]. The prominent ML-based works dealing with intelligent scheduling in the Medium Access Control (MAC) layer of the WSN can be found in [124]–[126]. Additionally, security, particularly intrusion detection in the WSN, has been widely studied using Machine Intelligence theory that has appeared in [127]–[130]. Furthermore, other research areas in the WSN have also enjoyed ML applications that include event detection and query processing [131]–[133], and QoS, data integrity, and fault detection [134]–[136]. In addition, the K-Nearest Neighbor (K-NN), which is a light-weight supervised learning algorithm, was applied extensively in the WSNs, particularly for the query processing tasks [131], [132]. However, when the WSNs produce a significantly high-dimsensional data (i.e., exceeding 10 dimensions), the K-NN is reported to lead to inaccurate results [137]. Decision tree is another classification method which has been frequently applied to WSNs [138]. However, those applications are limited to linearly separable data and suffer from high complexity since constructing complete optimal learning trees is an NP-complete problem [139]. Furthermore, the use of ANNs, particularly deep neural network architectures, has been limited in [140] due to the high computational requirement to learn the weights and high management overhead in the distributed WSN topologies. For node localization of the WSNs, neural networks with self-organizing maps (i.e., Kohonen's maps) and Learning Vector Quantization (LVQ) [141] have been used. However, the data set required for training such neural networks was reported not to be sufficient by Hinton and Salakhutdinov [28]. As a consequence, instead of deep learning architectures, many researchers opted to employ SVM for classifying data points using labeled training samples [138], [142]. In particular, the SVM algorithm, which typically optimizes a quadratic function with linear constraints, has been extensively exploited for node localization [143]–[145] and security applications [127], [130], [146]–[148] in WSNs.

In addition to the supervised learning methods, unsupervised learning approaches have also been widely used in WSNs, particularly to classify the sample set into various groups based on their similarity. Naturally, the unsupervised ML algorithms have been widely adopted for clustering of WSN nodes and data aggregation problems [149]–[155]. Among the most frequently used clustering methods in WSNs, the K-means clustering [156] is notable. However, for reduction of dimensionality, researchers used a multivariate method exploiting the PCA [157]. However, to exploit the distributed nature of the WSNs, reinforcement learning, particularly Q-learning, is used extensively and efficiently for WSN routing problems [158]–[161]. On the other hand, the state-of-the-art deep Q networks described in Section II-A5 are yet to be exploited for WSNs. Furthermore, Alsheikh *et al.* [118] recommended deep learning methods [21] and the non-negative matrix factorization algorithm [162] for more efficient unsupervised learning methods for the WSNs.

B. Network Traffic Classification and Deep Learning

Traffic classification by network traffic control systems is another area where ML continues to be of long-term interest to the networking community [163]–[171]. This is because the state-of-the-art ML, and now deep learning techniques, are anticipated to efficiently perform network traffic classification useful for network monitoring, QoS, intrusion-detection, and so forth in a wide variety of network settings [172]. Researchers used supervised ML techniques to label all available network traffic traces with previously known applications in [173]–[175]. However, the supervised learning techniques were considered to be impractical by Wang *et al.* [172] because of the limited training data and the manner in which new applications continue to appear. On the other hand, unsupervised ML techniques were investigated in [170], [171], and [176]. The works in [170] and [171] provided automatic network traffic classification based on exploiting the traffic features, e.g., time-interval between packets arrival, packet-size, recurring traffic patterns, and so forth. While classifiers like naive Bayes, decision trees, and neural networks [177]–[179] are extensively used in these works, the training processes are not real-time. This rendered the automatic traffic classification inefficient, similar to the unsupervised method. In order to remedy the issue, in [176], Erman *et al.* employed an unlabeled database for the training. However, due to its high complexity, a semi-supervised learning technique was proposed by Erman *et al.* [180]. Such unsupervised and/or semi-supervised learning algorithms, however, may not be directly applied to next generation Software Defined Networks (SDNs) without taking into consideration the decoupled control and data planes. The work in [172] presented a traffic classification engine to first conduct local traffic identification at the SDN edge switches, and then perform a "global" traffic classification via the network controller, which is responsible for training, constructing, and refining QoS policies based on the learned traffic information. Once a large amount of data on network traffic flows and their corresponding labels are available, the problem of protocol classification was demonstrated to be efficiently solved by deep learning architectures like deep neural networks [15] and/or stacked Auto-Encoders [181]. It was also mentioned in [181] and [182] that the latter performs better in contrast with the deep neural networks for classifying any flow data to a predefined protocol with an accuracy enough to be used in a real application. Also, the use of deep architectures was shown to reveal the highly probable anomalous and/or disguised flows. Furthermore, the unknown network flows were estimated to be over 17%. The work clearly showed that the deep learning models are able to distinguish more than half of the flows, which are difficult to be identified by the traditional ML algorithms.

C. Network Flow Prediction With Deep Learning

In addition to network traffic classification, flow prediction is another important area of the network traffic control systems, which has witnessed a growing number of deep learning applications recently. A network traffic flow is defined to be a sequence of data packets, which share the same context between source-destination pairs that include Transport Control Protocol (TCP) connections, media stream, and so forth. In order to manage the limited networking resources, information on the flow characteristics like the burst size (i.e., packets number and packet-size) and the inter-burst gap are often used. In the case of SDNs, the flow information is particularly useful for programming routers, mitigating wireless interferences, scheduling congested data traffic, and so on. Among traditional ML techniques, Basu *et al.* [183] investigated a wide range of time-series models for the Internet data traffic such as the auto-regressive moving average process. Claffy *et al.* [4] showed how to estimate the original packet-size distribution of a flow from the packet sampling performed at the routers. On the other hand, the work in [184] proposed a method to obtain the original frequencies of flow lengths from a sparse packet sampling. That work was similar in spirit with the face recognition system in [185] for performing classification via a sparse representation of features. Similar sparse representations of features based pooling to construct higher-level features for traffic flows forecasting were investigated by Rresenhuber and Poggio [186] and

Raina *et al.* [187]. The "origin" flow pattern inference is performed for the traffic generated from a Wireless Local Area Network (WLAN) in [188] by applying SVMs. On the other hand, deep learning architectures were introduced by Coates and Ng [189] to pool over multiple features for the flow prediction. On the other hand, the work in [12] presented a novel deep learning based sparse coding with forced incoherent dictionary atoms [190]–[192] for conducting origin flow prediction. Furthermore, Oliveira *et al.* [182] demonstrated the viability of stacked Auto-Encoders for short-term network traffic forecast based on the Internet traffic time-series obtained from the DataMarket dataset [193].

D. Deep Learning in Social Networks

Recently, social networks [195] have become a hot research topic as beyond 4G mobile networks continue to emerge attracting deep learning applications [194]. In the mobile social networks, based on the users' interaction, predicting their behavior toward a certain application, service, location, preference, and so forth is essential for the network operators. E-commerce sites, online marketing, advertisement display networks, and so forth use such information of the social network users' intentions and behaviors to maximize their efficiency [196]. The different patterns exhibited by the social network users consist of time spent per application, search frequency, recurring visits, and so on [197]. These patterns can be exploited to quantify their search behavior using ML techniques [198]. The work in [199] uses a probabilistic generative process to model user exploratory and purchase history. Also, they introduced the latent context variable to take into account both spatial and temporal features. Thus, the work aims to predict the social network users' decisions for specific contexts, and accordingly provide them with appropriate recommendations. This is also evident in the recent application of deep learning approaches to predict user activity within the Web content [199]. The deep learning algorithms are replacing the state-of-the-art ML techniques such as logistic regression and boosted decision trees. The deep neural networks, in particular, are superior to the conventional ML methods because of their capability to capture the non-linear relationship between the input features from the social network users. In addition, the deep learning architectures are shown to consist of a superior modeling strength compared to the existing ML approaches [199]. Furthermore, in the DeepWalk representation proposed by Perozzi *et al.* [200], a novel deep architecture to encode social relations in a continuous vector space was proposed. The advantage of the DeepWalk architecture consists in its ability to exploit local information obtained from truncated random walks to learn latent representations. The latent representation learned through the deep architecture was demonstrated to outperform several multi-label network traffic classification methods such as SpectralClustering [201], Modularity [202], EdgeCluster [203], and the weighted-vote Relational Neighbor (wvRN) [204] for social networks like BlogCatalog, Flickr, and YouTube. Also, the deep learning based representations were shown to be scalable and parallelizable, which can exploit the state-of-the-art GPUs for efficiently executing deep learning algorithms. On the other hand, Liu *et al.* [11] considered an unsupervised method with little samples from an online social network service model in order to perform the social links prediction. In addition, they developed a feature representation method. By combining the link samples and their values based on the link prediction and the obtained features, they further proposed the use of RBMs for link prediction with a significantly high accuracy. Moreover, Lazreg *et al.* [205] used deep learning for social media analysis in emergency situations. Their deep learning system was used to extract features and patterns related to the text and concepts available in crisis-related social media posts and harness them to obtain an idea regarding the crisis. Also, they demonstrated the great potential of deep learning architectures to play a substantial role in the future social networks in noisy emergency situations, i.e., during crises events. It is also worth mentioning that the government-based initiatives, e.g., the "anticipatory intelligence" [206], leverage deep neural networks to analyze social networks data to predict possible occurrence of social unrest. In the remainder of the section, we discuss deep learning applications in other network traffic control aspects, namely mobility prediction cognitive radio, and self-organized network.

E. Mobility Prediction With Deep Learning

With the proliferation of 4G (such as Long Term Evolution (LTE) and WiMax), beyond 4G, and heterogeneous cellular networks along with the recent advances in portable devices, the users can nowadays enjoy mobile network access. As a consequence, mobility prediction is a critical issue for the operators in order to determine capacity estimation, resource allocation, and so forth [216]–[219]. Chon *et al.* [220] stressed on predicting human mobility as a critical need for broad-domain applications (e.g., ranging from simple home preheating and sending dinner coupons to epidemic control [221], [222], urban planning [223]–[225], traffic forecasting systems [226]–[228], resource management of mobile communications [3], [229], [230]). With an aim to offer appropriate services to the mobile users in a timely fashion, they proposed time-resolved places and paths prediction via monitoring users' mobility. Such research works addressing mobility prediction have considered ML techniques for a while. With the emergence of the state-of-the-art deep learning methods, mobility prediction is anticipated to be even more accurate than the conventional ML approaches. For instance, Sundsøy *et al.* [231] considered a large, raw mobile phone dataset for the duration of over 3 months to construct a deep learning model. Compared with the traditional data mining algorithms, the deep learning based method showed a significantly accurate classification of the mobile users with different socio-economic groups that could be used for predicting mobility of the users. The result also indicated that the deep learning algorithms are able to capture the complex relationships between various dimensions of the massive data without suffering from the overfitting of the training data. In addition, using only a single dimension of the data in its raw form, the deep learning model was demonstrated to achieve a 7% better performance in contrast with the baseline constructed with custom engineered features from multiple data dimensions. Furthermore, Sundsøy *et al.* indicated that finding a general representation of mobile data can be exploited for a number of other prediction tasks. Ghouti [232] proposed a mobility prediction by using fully-complex extreme learning machines. These deep learning structures are based on fully-complex activation functions known as CELMs [233], which can operate without the need to tune the parameters of the connections between the input to hidden layers. Arbitrary computational nodes are applied irrespective of the training data so as to achieve a significantly low training error while estimating the output weights using a least-square solution. On the other hand, Song *et al.* [234] constructed a deep RNN to predict urban human mobility. However, Zhao *et al.* [219] argued that the conventional RNN fails to capture the long temporal features associated with the input sequence. Therefore, they designed a specific RNN architecture tailored for sequence prediction tasks in order to learn the time series with long time spans while automatically estimating the optimal time lags. Their deep RNN architecture was demonstrated to be able to predict the future movements and transportation mode of the users with an accuracy of 80% or more. On the other hand, Ouyang *et al.* [235] identified two representative works that pioneered in transforming the state-of-the-art deep learning algorithms into an effective online learning model. First, Zhou *et al.* [236] employed the denoising Auto-Encoders for online learning, which is capable of learning new patterns at the cost of slow learning speed due to large parameters space and a lack of support for parallel execution. As a remedy, Xiao *et al.* [237] introduced a hierarchical training algorithm using a deep CNN model, which supports parallel execution and is particularly suitable for the massive mobility datasets. Based on the lessons learned from these two works, Ouyang *et al.* [235] proposed an online deep learning framework called the "DeepSpace" in order to predict human moving paths by exploiting a deep CNN architecture that can deal with parallel online data. The "DeepSpace" framework, along with the traditional CNN architecture was applied to a mobile cellular network dataset obtained from a city of south-east China. The results demonstrated that the "DeepSpace" framework outperforms the traditional CNN-based approach by learning the spatio-temporal features of the network-level mobile data in a much more efficient manner.

F. Deep Learning in Cognitive Radio and Self-Organized Networks

In the remainder of this section, we first describe the applications of deep learning in Cognitive Radio Networks (CRNs) [238], [239], and then discuss how the deep learning methods are also becoming useful for Self-Organized Networks (SONs).

Traditionally, CRNs rely on intelligent learning techniques to learn from and adapt to their environment, and deep learning applications are recently appearing as a promising CRN enabler [240]. Most of the contemporary research works involving CRNs attempted to employ policy-based radios, which are hard-coded comprising a myriad of rules for the radios to behave in specific situations and/or applications [241]. In other words, in a typical CRN, there is a reasoning engine that permits the radios to remember lessons learned in the past and act quickly in the future. As a simple example of a rule-based CRN, consider an IEEE 802.11 modulation controller, which changes its modulation in response to the varying values of Signal-to-Noise-Ratio (SNR) [242]. This kind of cognitive decision making is based on reasoning rather than learning. The work in [243] points out the significant difference between reasoning and learning in CRNs. By doing so, the work aims to formalize the application of MLs to CRN to deal with problems like capacity maximization, dynamic spectrum access, and so forth. Abbas *et al.* [244] presented the evaluation and challenges of various learning techniques' applications. However, Zorzi *et al.* [245] stressed on the fact that the current ML techniques applied to the CRNs have traditionally adopted shallow architectures. Recently, they presented a novel perspective on CRN optimization by employing learning and distributed intelligence, and described how deep learning architectures can be utilized. Furthermore, O'Shea and Corgan [246] demonstrated the viability of using deep CNNs in CRNs. Their work builds upon successful ML algorithms for image and voice recognition to flexibly use the deep CNNs to flexibly learn features across many different tasks. In contrast with the conventional ML approaches, their deep learning based method exhibited much better performance when applied to the blind temporal learning on large and densely encoded time series datasets obtained from the CRN.

The work conducted by Zorzi *et al.* [245] focused on unsupervised training of stacked RBMs to introduce a variety of deep learning models for CRNs. The deep learning methods have also been considered to be currently the state-of-the-art in CRN modeling by Zorzi *et al.* Furthermore, they highlight on the CUDA framework refer to Section III-F, which is identified to an extremely powerful parallel computing architectures exploiting GPUs for efficiently constructing incredibly large-scale deep learning models containing millions of connection weights [247], [248] capable of being trained in an unsupervised manner by using the huge number of patterns in the currently available datasets [33]. In addition, the deep learning system is shown to be able to build rich and abstract representations, which can be served as input to a variant of the Q-learning algorithm as discussed in Section II to improve the overall behavior of the CRN. Another application of the deep learning architecture, in the form of a DBN, was given in [245] to classify the primary user agents in a CRN. This application, referred to as the COBANET, demonstrated a substantial reduction of the number of labeled data and improvement of the classification accuracy. With COBANET, the CRN performance in terms of spectrum sensing and handoff delay for changing channels improved in a significant manner compared to existing methods [249].

Also, in future wireless networks, self-organization has been identified as a critical issue by a number of researchers [250]. The key idea behind the so-called Self

Organized Networks (SONs) is for them to mimic the capabilities of biological systems (e.g., swarms of insects/fish) to autonomously adapt to the changes in the surrounding environment. Research attention is growing in developing ML algorithms to construct SONs [251] by not only autonomously adapting to varying conditions but also to learn based on experience. Zorzi *et al.* [245] hypothesized that deep learning techniques are likely to play a pivotal role in the learning aspect of the SONs. Furthermore, when combined with reinforcement learning, the deep learning systems for SONs are indicated to capable of performing network optimization [252]. This was implemented by Razavi *et al.* [253], [254] where they proposed reinforcement learning combined with fuzzy logic to optimize the down-tilt of the antennas of an eNB (i.e., a 4G base station) to achieve self-healing, self-configuration, and self-optimization functions. Their proposed learning architecture exhibited robustness to environmental changes and demonstrated improved performance over comparable heuristics. On the other hand, a Fuzzy Q-Learning algorithm to jointly optimize the coverage and capacity of a wireless cellular network was proposed by ul Islam and Mitschele-Thiel [255]. Indeed, the reinforcement learning based approach is becoming the mainstream direction for the CRNs as shown in the work conducted by Morozs *et al.* [256]. In that work, a heuristically accelerated reinforcement learning, referred to as HARL, was applied to the problem of dynamic spectrum sharing in LTE cellular networks. By utilizing external information, e.g., Radio Environment Msap (REM) for guiding and speeding up the learning process, HARL achieved a significant reduction in the secondary system's interference to the primary systems of the considered CRN. Furthermore, it resulted in a much higher system throughput in contrast with the state-of-the-art reinforcement learning approaches.

In this section, we extensively discussed the state-of-the-art deep learning applications for a number of network environments. In particular, we stressed on investigating how deep learning applications are disrupting the networking related systems in various settings, from basic WSNs to advanced CRNs. In the following section, we describe a new application of deep learning for routing in order to facilitate intelligent network traffic control systems.

V. New Area: Deep Learning Based Routing

As discussed in the earlier section, deep learning can be used in a wide range of networking related areas. Furthermore, with the development of new technique in deep learning, in this section, we present a new area of intelligent traffic control systems facilitated by deep learning based routing.

Since network traffic grows exponentially in the recent years, traffic control is essential to ensure the QoS, especially in the real-time multimedia networks where packet retransmissions due to the traffic congestion are not a sensible option [207]. Routing management is a crucial aspect for traffic control as the poorly chosen paths can lead to network congestion, and then the following retransmissions of the lost packets may further aggravate the congestion. In traditional routing protocols, the main concept is to choose

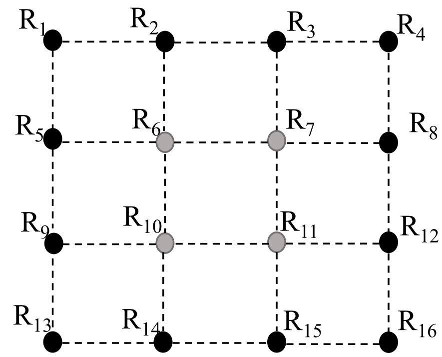


Fig. 5. The considered wireless mesh backbone network.

the path having the maximum or minimum value or metric, for instance, the Shortest Path (SP) algorithm [208]. However, these methods have different shortcomings. For example, the traditional SP algorithm has the problem of slow convergence, which is not suitable for dynamic networks since the slow response to the network changes can lead to severe congestion [209]. To solve this shortcoming of conventional routing methods, researchers have explored to adopt machine learning to manage the path intelligently [6]–[10]. However, to the best of our knowledge, contemporary researchers did not exploit deep learning for network traffic control. From hereon, we introduce a proof-of-concept of applying the deep learning technique for performing intelligent traffic control in future networks.

To clearly show how the deep learning architecture can be used for traffic control, we suppose a 4×4 wireless mesh backbone network as shown in Fig. 5. In the considered network, assume that the packets are generated only in edge routers and destined for other edge routers since the access terminals are all connected to the edge routers while the inner routers just play the role of forwarding packets. Each edge router is assumed to run several DBNs to construct the whole paths to other edge routers and attach its packets with the corresponding paths. The inner routers do not need to run the DBNs and just read the path to forward the packets. For each DBN, the units in its bottom layer are characterized as the traffic patterns of all routers in the network while the top layer represents the next node for an origin-destination pair. As the number of routers in the network is 16, the bottom layer consists of 16 units. We adopt a 16-dimensional vector format output to represent the next node path of the DBN such that each of its elements has a binary value. Furthermore, only a single element in this vector can be 1. The position or order of the element in the vector having the value of 1 indicates the next node. As a path is composed of several routers, several DBNs are needed to build a complete path. Since the number of edge routers is 12 and every DBN only outputs the next node from one router to a destination router, the number of DBNs in the network is 180. To relieve the training burden, each router in the network trains the DBNs giving the next node from itself for each of its destination routers.

As mentioned earlier, the training of DBN can be separated into two steps. The first step is pre-training of the

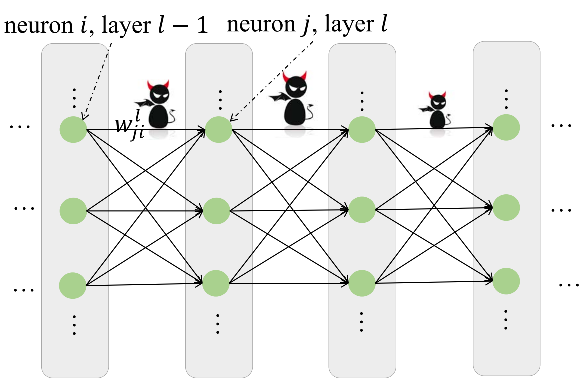


Fig. 6. The error propagates layer by layer and the value changes.

architecture with the Greedy Layer-Wise training method, and the following step is to fine-tune the architecture with the backpropagation method. The details can be referred to previous works [107]. Since the training of deep learning architecture is to output the desired values, we usually utilize the distance between the practical output and the desired output to measure the training errors. To find the method how to decrease the distance, we need to first know how the training errors are produced. To easily understand how the error is defined, imagine that the error is represented by a demon icon as depicted in Fig. 6. The demon sits at the j^{th} neuron in the l^{th} layer. The demon interferes with the neuron's operation when the input to the neuron comes in. It adds a little change (i.e., a noise) to the neuron's weighted input, so that instead of generating only the expected output, the neuron instead outputs the expected output along with the noise. This change propagates through the subsequent layers in the DBN, finally causing the overall cost to change by a substantial amount [210]. Consequently, to minimize the value of the cost function, we need to adjust the values of the weights. The mathematical procedure to conduct this is to utilize the derivative of the cost function to update the weight $w_{ji}^{(l)}$ as shown in Fig. 6. To sufficiently adjust the weight, we need to use the backpropagation method several times until the value of the cost function reaches the requirement. Using Table I, we can find the values of the weights and the Means Square Errors (MSE) which are used to measure the value of the cost function. It can be clearly seen, from Table I, that the value of MSE decreases with the number of backpropagation steps.

After the training phase, every router obtains the values of the weights and biases of the DBNs that predict the next nodes for edge routers from itself. The next step is that every router forwards these values to the edge routers. Therefore, once every edge router gets the traffic patterns of all routers, it can utilize the DBNs to construct the whole paths to other edge routers.

Next, based on the simulation topology in [107], the performance of the deep learning based routing is evaluated. Since the computations of all the routers were outsourced to a single machine, the evaluation was restricted to a medium scale wireless mesh backbone network comprising 16 routers shown in Fig. 5 rather than a full-scale backbone network topology.

TABLE I
THE VALUES OF WEIGHTS AND MEAN SQUARE ERROR (MSE) FOR DIFFERENT NUMBERS OF THE BACKPROPAGATION STEPS

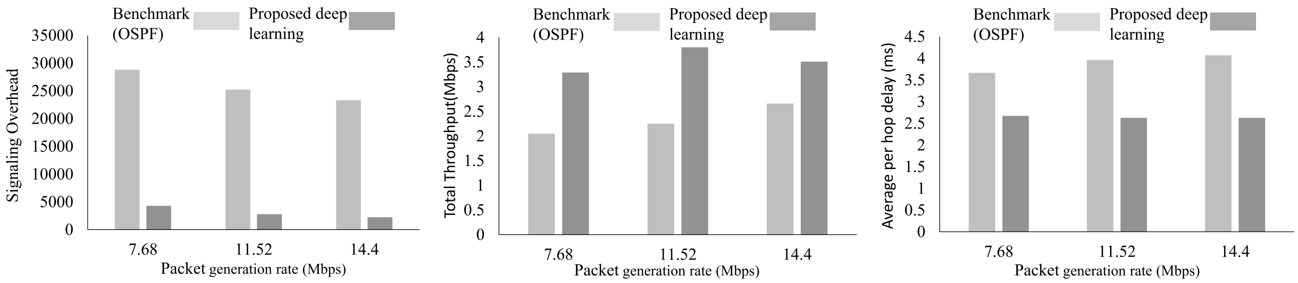
Step	Weight	Value	MSE
100	w_{00}^1	0.825148	2.918418×10^{-5}
	w_{10}^1	-0.600210	
	...	...	
	$w_{11,15}^3$	-0.074882	
	$w_{12,15}^3$	-0.052039	
200	w_{00}^1	0.825954	6.719093×10^{-6}
	w_{10}^1	-0.598363	
	...	...	
	$w_{11,15}^3$	-0.143467	
	$w_{12,15}^3$	-0.121055	
...	...	...	...
500	w_{00}^1	0.825623	9.843218×10^{-7}
	w_{10}^1	-0.597710	
	...	...	
	$w_{11,15}^3$	-0.231773	
	$w_{12,15}^3$	-0.209574	
...	...	...	...
1000	w_{00}^1	0.825115	4.882770×10^{-7}
	w_{10}^1	-0.597787	
	...	...	
	$w_{11,15}^3$	-0.263871	
	$w_{12,15}^3$	-0.241689	

TABLE II
COMPARISON OF THE LEARNING STRUCTURES FOR THE CONSIDERED NETWORK

$MSE(10^{-5})$ layers / nodes	4	5	6
14	2.157	2.224	2.232
16	2.15	2.223	2.229
18	2.15	2.218	2.224
20	2.159	2.212	2.225

Note that this scale of simulation is sufficient as long as it demonstrates that the proposed deep learning system outperforms the conventional routing strategies such as OSPF. The data and control packets sizes are both set to 1Kb. The link bandwidths are set to 8Mbps which is reasonable for this scale of wireless mesh backbone [211]. Every node is assumed to have an unlimited buffer. The overall data packet generating rate in the considered network is varied between 7.68Mbps to 14.4Mbps. For comparison of the adopted deep learning system, OSPF is used as the benchmark method. In the conducted simulations, the time-interval of each path updating phase is set to 0.25s during which signaling is exchanged once.

To decide the number of layers and the number of units in every hidden layer, different deep learning structures are compared first as shown in Table II. In the training process of the deep learning system, MSE is often used as the stopping condition of the training. Here, we use this value to measure the performance of different deep learning structures. For simplicity, 12 structures in which the numbers of layers are varied from 4 to 6 while the number of nodes in every hidden layer is varied in the range of {14, 16, 18, and 20}. It can be noticed that when the number of nodes in the hidden layers is fixed, the value of MSE grows bigger as the number of layers grows. This indicates that 4 layers are sufficient for our training data. In other words, more layers will cause



(a) Comparison of signaling overhead. (b) Comparison of aggregate throughput. (c) Comparison of average per-hop delay.

Fig. 7. Comparison of signaling overhead, throughput, and average per-hop delay between the benchmark method (OSPF) and the new, proof-of-concept, deep learning system with different overall data packet generate rates.

the problem of over-fitting for the considered scenario. On the other hand, the changing trend of MSE with the number of nodes in the hidden layers is not the same for different numbers of layers. For instance, for the deep learning systems comprising 4 layers, MSE reaches the minimum value when the number of nodes in every hidden layer is 16 and 18, which means that these two types of structures can achieve similar performance. However, the training will be more complex with the increase of nodes in the hidden layers. Therefore, we choose the deep learning structure of 4 layers and 16 nodes in every hidden layer for the evaluation of network performance metrics shown in Figs. 7(a), 7(b), and 7(c). As shown in these figures, three network performance metrics are evaluated between the conventional OSPF and deep learning. First, the signaling overhead is compared with the benchmark method (i.e., OSPF) in Fig. 7(a). The results show that the signaling overhead of OSPF is much higher compared to that of the deep learning system no matter how the generation rate is changed. The lower signaling achieved by our proposal can be explained as follows. While the conventional OSPF method needs all the routers to frequently flood their respective neighbors with the routing information, in the deep learning method, only the edge routers' traffic patterns are sufficient to compute the whole paths with an accuracy as high as 95%. Therefore, in the deep learning method, only the edge routers need to exchange traffic patterns among themselves and the traffic patterns of the inner routers are arbitrarily set in the running phase. Then, in Fig. 7(b), the throughput of the two methods are compared. It can be observed that the throughput achieved by the deep learning system is almost close to the average data generation rate of the routers since it avoids congestion and packet drops by evaluating routes to the destination much faster than the conventional OSPF. Finally, Fig. 7(c) demonstrates that the deep learning method finds the appropriate routes quickly enough so that the average per-hop delay of the adopted deep learning system is significantly lower than that of the benchmark method.

In this section, we highlighted the deep learning application on routing in a backbone network, and provided the detailed description of how to consider an appropriate deep learning system, how to characterize the inputs and outputs of the system, and how to train the system. In addition, simulation results were provided to demonstrate how the deep learning system outperforms the conventional routing strategy. However, it is worth noting that the deep-learning application for the network traffic control system is still at a preliminary stage of implementation, and therefore, various issues need to be considered as future research issues [212].

VI. Open Research Issues

A number of research challenges are anticipated to emanate from the deep learning applications for network traffic control systems. In this section, we discuss these open research issues.

A. Training Data Processing

Deep learning is a widely used technique in many areas, and big data are often served as its input for training. Such big data are collected from the real world, and processed as training sets. However, can such raw data sets be directly used in deep learning? In most learning systems, we require the training sets to be without redundancy while having accuracy and balance. However, in the real world, the available data sets are not always so perfect. For example, the data may be represented in some counterpart classes much more than those in some other passive classes [213]. Furthermore, they may contain many redundant and mislabeled data in it [214]. Large training sets require large memory and enormous amounts of training time and contribute to large training costs. Decreasing redundant (i.e., useless) training data is essential to improve the training effect. Besides normal training data processing, different deep learning application areas still have their special demands. For instance, consider the intelligence translation area, which confronts the challenge in determining how to select non-domain-specific language model training data [215]. In our prior work [107], in order to train network intelligent routing, we need large amounts of training data of existing routing strategies. Because of the difference of data between different routing strategies, we only used OSPF-based routing data as our training data. However, some portions of such datasets may still have imbalance, which cannot be directly used in the training phase. Hence, more accurate data sets to characterize the routing path is required. This is why, in Section V, we described the need to use many characterization strategies to pre-process and filter data. Indeed, there is still room for further improvement of dataset selection. For example, using the real routing data instance of multiple routing protocols, using more than 3-dimensional input data to characterize the routing path, and so forth need to be considered in the future. All in all, how to select and process training data to suit the training system and improve the quality of the training data, still, pose a significant research challenge.

B. Insights on the Deep Learning Problem

Due to its profound impact in the AI area, deep learning is dramatically changing the researchers' way of the thinking and interpreting the representation of problems, which are currently being solved with analytics. The deep learning technique moves away from instructing the computer how to computationally solve a problem to training the computer to solve the problem by itself. However, researchers must gain the insight on how the deep learning actually works. In other words, researchers should not take the deep learning technique for granted just by arbitrarily applying it to solve large computational problems. Instead, how the deep learning is working through the training process should be carefully and deeply understood. The deep neural network contains layers of nodes to represent features of objects. Can we have a method to analyze how those neural nodes construct such features of the neural network? If this is possible, the structure of deep learning system may be improved easily, and even newer, more sophisticated deep learning structures can be invented.

C. Optimize GPU-Based Deep Learning for Software Defined Routers

Software Defined Routers or SDRs are becoming an attractive platform for flexible packet processing. Furthermore, the provision of GPU-based deep learning in SDRs is both an opportunity and a challenge for researchers. Particularly, the CUDA streams provide an interesting way to optimize GPU-based applications, particularly for SDRs. The CUDA streams can increase the parallelism and throughput in a significant way. However, more research is needed in this aspect to encourage researchers in realizing that the future of networking heavily hinges upon SDNs. Therefore, more investigation is required on the GPU-based SDRs such as how to further improve the packet generation rate, how to improve resource management, and so forth to make an optimized execution of the deep learning algorithms. Also, on-chip deep learning can be another area where researchers should devote a lot of attention. This is because while the SDRs can be a proof-of-concept of the ability in deep learning algorithms for performing effective packets routing, the hardware implementation of deep learning can be much more efficient. With the current state-of-the-art hardware, a tempting idea could be to explore the use of integrated graphics processors instead of using dedicated discrete GPUs. By doing so, the GPU could be co-located with the CPU on the Quick-Path-Interconnect (QPI). In other words, moving the location from the typical PCI-express bus to QPI can offer more bus bandwidth to memory, and theoretically integrated graphics processors may be able to achieve even higher throughput as indicated in [257].

D. Scalability of Deep Learning Applications for Network Traffic Control Systems

Specific deep learning algorithms may work well for specific applications. In addition, the time versus space tradeoff could present a significant challenge for using deep learning algorithms for network traffic control systems. This may lead to scalability issues of using deep learning applications for large network traffic control systems, such as the Internet. For instance, in the case of the SDR exploiting deep learning for packets forwarding/routing scenario, the SDR not only requires a highly parallelized packets processing methodology but also needs to have a significantly huge storage. Furthermore, fast storage such as Ternary Content Addressable Memory (TCAM), Solid State Disk (SSD), and so forth are expensive and may not be scalable without adequate encoding schemes for deep learning inputs and outputs. Therefore, how to practically expand the storage for the SDRs to contain the training data for the deep learning system is a formidable research challenge. Using high-speed Storage Area Network (SAN) may be a viable approach that can provide access to the relevant dataset for training the deep learning system while moving the unwanted dataset to a backup archive storage. In the future, researchers need to adequately address the aforementioned scalability issue and develop appropriate solutions to deal with the same.

E. Deep Learning in the Internet-of-Things

Recently, the Internet-of-Things (IoT) emerged as a hot research area that aims to connect billions of things (e.g., sensors, objects, machines, devices, and so forth) in order to collect and process detailed information about events and environments to solve various challenges [2], [258]. As described in Section II, Google, Microsoft, Amazon, and other technology giants are heavily investing in deep learning techniques. These deep learning techniques can lead to the so-called deep linking of the IoT. Deep linking refers to a unified protocol or interface to allow applications to trigger and communicate with one another behind the scenes. For instance, the calendar application collaborates with the navigation application in a smart phone to indicate when the user should leave work to avoid the rush-hour traffic, and arrive at the favorite restaurant right on schedule. The IoT prototype called GreenIQ (a smart irrigation system for gardens) may know when it is about to rain, and avoid activating the garden sprinklers. The deep learning algorithms can be the behind-the-scene enablers to this deep linking interface in the IoT applications. In other words, the state-of-the-art IoT paradigm is restricted to connectivity. The IoT systems need to step beyond connectivity into the realm of intelligence empowered by deep learning. For instance, the smart home is a frequently referred to the use case of a typical IoT system that exhibits connectivity of devices, gadgets, wearable things, and so forth in the user home. However, the intelligent aspect of the smart home requires deep learning, which is becoming prevalent in more and more devices. The networks that connect the things need to be somehow augmented with the deep learning systems. Future research works need to deal with how to realize such an exciting harmony of the IoT and deep learning with the deep linking interface.

F. Deep Learning Approaches to Mobile Edge Computing

In the next generation networks, the various analytics on the massive IoT data and big data are expected to pose a significant challenge for conventional data-driven and rule-based expert systems. While they are easy to implement, such systems are usually slower in adapting to new data types. As a consequence, deep learning techniques are being considered for analytics in data centers as well as in the network edge. At the first glance, the deep learning approaches to the data analytics may not appear to be intimately connected with the network traffic control systems. However, the fusion of these various disciplines, i.e., communication networks, data analytics and computing, and deep learning, is imminent. In such an interdisciplinary paradigm, where should the deep learning algorithms be implemented? Data centers, which typically comprise processing and large memory resources, robust networks, and huge storage supporting massive datasets, may be able to make the best use of the deep learning algorithms for the state-of-the-art Web analytics. On the other hand, for the analytics tasks with immediate or near-immediate response time (e.g., the real-time IoT analytics), recent researchers are showing a growing interest in analytics at the network edge, referred to as the Mobile Edge Computing. This paradigm also referred to as the edge computing or edge analytics, may present a challenge for the deep learning algorithms simply because the equipment available in the network edge may not have sufficient resources to execute a robust deep learning system. In other words, whether a sophisticated deep learning system that consists of a complex computational circuit with millions of parameters can be effectively used for real-time the edge analytics can be genuinely a daunting challenge in the future. Currently, deep learning libraries such as TensorFlow, which require less memory and storage, while supporting heterogeneous distributed systems, may be effective for real-time analytics tasks exploiting the mobile edge nodes. However, this requires more investigation. Furthermore, the current deep learning architectures are drawing on only a tiny fraction of what is known about real neurons and brains. Researchers are working on developing further variations of deep/extreme learning machines, which may change the analytics and big data mining research area completely. For instance, recently, Intel announced that an autonomous vehicle may generate up to 4000GB of data every day [259]. On board deep learning algorithms are required to absorb this huge amount of data from a plethora of senors in the vehicle. Accordingly, the deep learning system has to decide how much data should be processed locally and how much information should be uploaded to the cloud while ensuring that the driving decisions are made at real-time to ensure the safety of the driver. Again, this creates an intricately complex inter-discipline of communication networks, deep learning, data mining, and perhaps even robotics. This is just an example to indicate that in the future, there may be unique areas to apply the deep learning techniques where network traffic control systems will be subject to a much more complex inter-disciplinary trend.

G. Deep Learning for Network Security

Network security is another area, which may initially appear as unrelated to the network traffic control systems. However, the security is a critical aspect of any network traffic control system. In the network security research area, a plethora of ML-based approaches exist in [177] and [178]. Most of these approaches deal with identifying anomalous traffic patterns as potential risks to the networks. Deep learning can offer a more accurate classification of the network threats, from malware detection to Distributed Denial of Service (DDoS) attacks. Nevertheless, deep learning algorithms still need to be implemented in the intrusion detection systems, deployment of which has been a key challenge in the literature. Therefore, how to enable deep learning applications to detect potential threats on the user-scale may be explored. Deep learning may have a huge impact on cyber security, particularly in terms of detecting zero-day malware, new malware, and sophisticated Advanced Persistent Threats (APTs). The state-of-the-art deep learning methods can be expected to exhibit superior performance compared to the traditional ML techniques. This is because the deep learning paradigm can provide an accurate information on suspicious (i.e., anomalous) activity without intervention or supervised training from human analysts. In addition, the deep learning approaches are much more robust to significantly large sets of encrypted data compared to traditional ML-based intrusion detection systems. Hence, the deep learning methodology, theoretically, should be adopted by IT organizations. However, the organizational policy may be a major obstacle against adopting deep learning systems to combat malicious threats that the researchers have to appropriately address in the future research works.

Also, for a huge data set containing both benign and malicious data, the higher number of dimensionality means that the intrusion detection method needs to deal with dimensionality reduction technique to estimate the presence of malicious data. Therefore, the underlying deep learning algorithm of the intrusion detection system needs to be able to perform dimensionality reduction in an efficient manner. In this vein, Auto-Encoders may be a viable candidate. However, for the online detection case in a substantially large network, this may be particularly challenging since the deep learning algorithm needs to ingest a significantly large training set almost at real-time. In addition, normality poisoning poses another challenge to the unsupervised deep learning based intrusion detection methodology (e.g., based on DBNs) that may change the sanctity of data used for detecting malicious activities. In other words, the deep learning algorithm needs to take into account the likelihood that anomalous data may be cleverly hidden within normal information in the network, thereby making the whole detection process self-defeatist [260].

VII. Conclusion

Deep learning is a new breed of Machine Intelligence technique, which is gaining much popularity and wide use in various computer science fields, such as object recognition, speech recognition, signal processing, robotics, AI gaming, and so forth. However, the application of deep learning in network systems just started to receive research attention. In this survey paper, we discussed the state-of-the-art machine learning and new deep learning researches in the network related areas such as WSN and social networks, network traffic classification, network flow prediction, mobility prediction, and so forth. Furthermore, we provided a comprehensive guide on how deep learning applications can stimulate a new area of research involving smart network traffic control systems. In particular, we focused on the newly emerging deep learning based routing. We provided a step-by-step description of the deep learning technique used for intelligent network routing. In addition, simulation results were provided to demonstrate the superior performance of the deep learning based routing method compared to the conventional routing strategy. Furthermore, we discussed a number of open research issues, and indicated how deep learning, networking, and computing are heading toward an intricate yet imminent inter-disciplinary area, which the future researchers need to embrace.

References

[1] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with China perspective," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 349–359, Aug. 2014.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[3] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Chicago, IL, USA, Aug. 2013, pp. 1436–1444.

[4] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, "Application of sampling methodologies to network traffic characterization," in *Proc. ACM SIGCOMM*, San Francisco, CA, USA, 1993, pp. 194–203.

[5] J. S. Marcus, *The Economic Impact of Internet Traffic Growth on Network Operators 2014*, wIK-Consult Google Inc., Bad Honnef, Germany, Oct. 2014. [Online]. Available: http://www.wik.org/uploads/media/Google_Two-Sided_Mkts.pdf

[6] M. Barabas, G. Boanea, and V. Dobrota, "Multipath routing management using neural networks-based traffic prediction," in *Proc. 3rd Int. Conf. Emerg. Netw. Intell.*, Lisbon, Portugal, Nov. 2011, pp. 118–124.

[7] L. Zhang and S. C. A. Thomopoulos, "Neural network implementation of the shortest path algorithm for traffic routing in communication networks," in *Proc. Int. Joint Conf. Neural Netw.*, San Diego, CA, USA, Aug. 1989, p. 591.

[8] J. Barbancho, C. León, F. J. Molina, and A. Barbancho, "A new QoS routing algorithm based on self-organizing maps for wireless sensor networks," *Telecommun. Syst.*, vol. 36, nos. 1–3, pp. 73–83, Nov. 2007.

[9] M. K. M. Ali and F. Kamoun, "Neural networks for shortest path computation and routing in computer networks," *IEEE Trans. Neural Netw.*, vol. 4, no. 6, pp. 941–954, Nov. 1993.

[10] G. Boanea, M. Barabas, A. B. Rus, V. Dobrota, and J. Domingo-Pascual, "Performance evaluation of a situation aware multipath routing solution," in *Proc. RoEduNet Int. Conf. 10th Edition Netw. Educ. Res.*, Iaşi, Romania, Jun. 2011, pp. 1–6.

[11] F. Liu, B. Liu, C. Sun, M. Liu, and X. Wang, *Deep Learning Approaches for Link Prediction in Social Network Services*. Heidelberg, Germany: Springer, 2013, pp. 425–432.

[12] Y. L. Gwon and H. T. Kung, "Inferring origin flow patterns in Wi-Fi with deep learning," in *Proc. 11th Int. Conf. Auton. Comput. (ICAC)*, vols. 18–20, Philadelphia, PA, USA, Jun. 2014, pp. 73–83.

[13] J. Ngiam *et al.*, "Multimodal deep learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Bellevue, WA, USA, Jun. 2011, pp. 689–696.

[14] Z. Li and R. Wang, "A multipath routing algorithm based on traffic prediction in wireless mesh networks," *Commun. Netw.*, vol. 1, no. 2, pp. 82–90, Aug. 2009.

[15] S. Chabaa, A. Zeroual, and J. Antari, "Identification and prediction of Internet traffic using artificial neural networks," *J. Intell. Learn. Syst. Appl.*, vol. 2, no. 3, pp. 147–155, Jul. 2010.

[16] H. Goh, N. Thome, M. Cord, and J.-H. Lim, "Top-down regularization of deep belief networks," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, 2013, pp. 1878–1886.

[17] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Assoc., 2012, pp. 2222–2230.

[18] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann machines," in *Proc. AISTATS*, Fort Lauderdale, FL, USA, Apr. 2009, pp. 448–455.

[19] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep Boltzmann machines," *Neural Comput.*, vol. 24, no. 8, pp. 1967–2006, Aug. 2012.

[20] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Trans. Signal Inf. Process.*, vol. 3, no. 2, pp. 1–29, Jan. 2014.

[21] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[22] P. E. Utgoff and D. J. Stracuzzi, "Many-layered learning," *Neural Comput.*, vol. 14, no. 10, pp. 2497–2529, Nov. 2002.

[23] Y. Bengio and Y. Lecun, *Scaling Learning Algorithms Towards AI*. Cambridge, MA, USA: MIT Press, Aug. 2007.

[24] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global optimization of a neural network-hidden Markov model hybrid," *IEEE Trans. Neural Netw.*, vol. 3, no. 2, pp. 252–259, Mar. 1992.

[25] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Norwell, MA, USA: Kluwer Acad., 1993.

[26] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 7–13, Jan. 2012.

[27] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[28] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[29] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[30] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19. Vancouver, BC, Canada, Aug. 2007, pp. 153–160.

[31] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009, pp. 41–48.

[32] A. Dosovitskiy, J. T. Springenberg, M. A. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," *CoRR*, vol. abs/1406, no. 6909, pp. 1–9, Apr. 2014.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Assoc., Dec. 2012, pp. 1097–1105.

[34] K. Kavukcuoglu *et al.*, "Learning convolutional feature hierarchies for visual recognition," in *Advances in Neural Information Processing Systems*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Vancouver, BC, Canada: Curran Assoc., Dec. 2010, pp. 1090–1098.

[35] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[36] I. Sutskever, J. Martens, and G. Hinton, "Generating text with recurrent neural networks," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, Bellevue, WA, USA, Jun. 2011, pp. 1017–1024.

[37] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.

[38] T. Mikolov, M. Karafiát, L. Burget, J. Černock, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. INTERSPEECH*, Chiba, Japan, Sep. 2010, pp. 1045–1048.

[39] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 8624–8628.

[40] I. Sutskever, "Training recurrent neural networks," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2013.

[41] C.-Y. Liou, J.-C. Huang, and W.-C. Yang, "Modeling word perception using the Elman network," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3150–3157, Oct. 2008.

[42] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biol. Cybern.*, vol. 59, nos. 4–5, pp. 291–294, Sep. 1988.

[43] R. Socher, Y. Bengio, and C. D. Manning, "Deep learning for NLP (without magic)," in *Proc. Tutorial Abstracts ACL*, Jul. 2012, p. 5.

[44] Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using two layer networks," Dept. Comput. Inf. Sci., Univ. California at Santa Cruz, Santa Cruz, CA, USA, Tech. Rep. UCSC-CRL-9I-20, Jun. 1994.

[45] M. A. Côté and H. Larochelle, "An infinite restricted Boltzmann machine," *Neural Comput.*, vol. 28, no. 7, pp. 1265–1288, Jul. 2016.

[46] T. Oohori, H. Naganuma, and K. Watanabe, "A new backpropagation learning algorithm for layered neural networks with nondifferentiable units," *Neural Comput.*, vol. 19, no. 5, pp. 1422–1435, May 2007.

[47] C. L. P. Chen, C.-Y. Zhang, L. Chen, and M. Gan, "Fuzzy restricted Boltzmann machine for the enhancement of deep learning," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 2163–2173, Dec. 2015.

[48] C. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, Oct. 1992.

[49] *Methods and Apparatus for Reinforcement Learning*. Accessed on Dec. 2016. [Online]. Available: https://www.google.com/patents/US20150100530

[50] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *CoRR*, vol. abs/1509, no. 06461, pp. 1–7, Nov. 2015.

[51] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *CoRR*, vol. abs/1511, no. 05952, pp. 1–21, Nov. 2015.

[52] Z. Wang, N. de Freitas, and M. Lanctot, "Dueling network architectures for deep reinforcement learning," *CoRR*, vol. abs/1511, no. 06581, pp. 1–15, Nov. 2015.

[53] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *CoRR*, vol. abs/1509, no. 02971, pp. 1–14, Sep. 2015.

[54] I. J. Goodfellow, A. Courville, and Y. Bengio, "Large-scale feature learning with spike-and-slab sparse coding," in *Proc. 29th Int. Conf. Mach. Learn.*, Edinburgh, U.K., Jun. 2012, pp. 1–8.

[55] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, "Better mixing via deep representations," *CoRR*, vol. abs/1207, no. 4404, pp. 1–13, Jun. 2012.

[56] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Stat. (AISTATS)*, Fort Lauderdale, FL, USA, Apr. 2011, pp. 315–323.

[57] J. Chen and X. Liu, "Transfer learning with one-class data," *Pattern Recognit. Lett.*, vol. 37, pp. 32–40, Feb. 2014.

[58] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*, G. Montavon, G. Orr, and K.-R. Muller, Eds. Berlin, Germany: Springer, May 2012.

[59] R. Collobert, "Deep learning for efficient discriminative parsing," in *Proc. 14th Int. Conf. Artif. Intell. Stat. (AISTATS)*, Fort Lauderdale, FL, USA, Apr. 2011, pp. 224–232.

[60] N. Kato, M. Suzuki, S. Omachi, H. Aso, and Y. Nemoto, "A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 3, pp. 258–262, Mar. 1999.

[61] K. Saruta, N. Kato, M. Abe, and Y. Nemoto, "High accuracy recognition of ETL9B using exclusive learning neural network-II: ELNET-II," *IEICE Trans. Inf. Syst.*, vol. 79, no. 5, pp. 516–522, May 1996.

[62] *MNIST Database*. Accessed Dec. 2016. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[64] S. Rifai, Y. Bengio, P. Vincent, and Y. N. Dauphin, "A generative process for sampling contractive auto-encoders," in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, Edinburgh, U.K., Jun. 2012, pp. 1855–1862.

[65] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *CoRR*, vol. abs/1202, no. 2745, pp. 1–8, Feb. 2012.

[66] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[67] Emerging Technology From the arXiv. *Google Unveils Neural Network With 'Superhuman' Ability to Determine the Location of Almost Any Image*. Accessed Dec. 2016. [Online]. Available: https://www.technologyreview.com/s/600889/google-unveils-neural-network-with-superhuman-ability-to-determine-the-location-of-almost/

[68] T. Landauer, C. Kamm, and S. Singhal, "Learning a minimally structured back propagation network to recognize speech," in *Proc. 9th Annu. Conf. Cogn. Sci. Soc.*, Seattle, WA, USA, 1987, pp. 531–536.

[69] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "Tandem connectionist feature extraction for conversational speech recognition," in *Machine Learning for Multimodal Interaction*, vol. 3361. Heidelberg, Germany: Springer, 2005, pp. 223–231.

[70] O. Abdel-Hamid *et al.*, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[71] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Waikoloa Village, HI, USA, Dec. 2011, pp. 24–29.

[72] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. 12th Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Florence, Italy, Aug. 2011, pp. 437–440.

[73] D. Yu, M. L. Seltzer, J. Li, J. Huang, and F. Seide, "Feature learning in deep neural networks—A study on speech recognition tasks," *CoRR*, vol. abs/1301.3605, pp. 1–9, Jan. 2013. [Online]. Available: http://arxiv.org/abs/1301.3605

[74] (2016). *Microsoft Audio Video Indexing Service (MAVIS)*. Accessed on Dec. 2016. [Online]. Available: https://www.microsoft.com/en-us/research/project/mavis/

[75] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, vol. 28. Atlanta, GA, USA, May 2013, pp. 1139–1147.

[76] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, vol. 1631. Seattle, WA, USA, Oct. 2013, pp. 1–12.

[77] R. Socher, E. H. Huang, J. Pennin, A. Y. Ng, and C. D. Manning, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Granada, Spain: Curran Assoc., Dec. 2011, pp. 801–809.

[78] *The Ibrain Is Here and It's Already Inside Your Phone*. Accessed Aug. 2016. [Online]. Available: https://backchannel.com/an-exclusive-look-at-how-ai-and-machine-learning-work-at-apple-8dbfb131932b#.43bf9cm00

[79] G. E. Hinton, "Learning distributed representations of concepts," in *Proc. 8th Annu. Conf. Cogn. Sci. Soc.*, 1986, pp. 1–12.

[80] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[81] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, Helsinki, Finland, 2008, pp. 160–167.

[82] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars," in *Proc. ACL*, Sofia, Bulgaria, Aug. 2013, pp. 455–465.

[83] A. Deoras, T. Mikolov, S. Kombrink, and K. Church, "Approximate inference: A sampling based modeling technique to capture complex dependencies in a language model," *Speech Commun.*, vol. 55, no. 1, pp. 162–177, Jan. 2013.

[84] *Wall Street Journal-Based Continuous Speech Recognition (CSR) Corpus*. Accessed Dec. 2016. [Online]. Available: http://catalog.ldc.upenn.edu/docs/LDC94S13A/wsj1.txt

[85] H. Schwenk, A. Rousseau, and M. Attik, "Large, pruned or continuous space language models on a GPU for statistical machine translation," in *Proc. NAACL Workshop Future Lang. Model.*, Montreal, QC, Canada, Jun. 2012, pp. 11–19.

[86] D. Castelvecchi. *Nature News, Deep Learning Boosts Google Translate Tool*. Accessed Dec. 2016. [Online]. Available: http://www.nature.com/news/deep-learning-boosts-google-translate-tool-1.20696

[87] T. Schmidt, R. Newcombe, and D. Fox, "Self-supervised visual descriptor learning for dense correspondence," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 420–427, Apr. 2017.

[88] G. Pasquale, C. Ciliberto, L. Rosasco, and L. Natale, "Object identification from few examples by improving the invariance of a deep convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 4904–4911.

[89] T. De Bruin, J. Kober, K. Tuyls, and R. Babuška, "Improved deep reinforcement learning for robotics through distribution-based experience retention," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 3947–3952.

[90] L. Porzi *et al.*, "Learning depth-aware deep representations for robotic perception," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 468–475, Apr. 2017.

[91] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Hamburg, Germany, Sep. 2015, pp. 4415–4420.

[92] J. Yu, K. Weng, G. Liang, and G. Xie, "A vision-based robotic grasping system using deep learning for 3D object recognition and pose estimation," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Shenzhen, China, Dec. 2013, pp. 1175–1180.

[93] A. S. Polydoros, L. Nalpantidis, and V. Krüger, "Real-time deep learning of robotic manipulator inverse dynamics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Hamburg, Germany, Sep. 2015, pp. 3442–3448.

[94] J. Li, P. Ozog, J. Abernethy, R. M. Eustice, and M. Johnson-Roberson, "Utilizing high-dimensional features for real-time robotic applications: Reducing the curse of dimensionality for recursive Bayesian estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 1230–1237.

[95] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "Fast robust monocular depth estimation for obstacle detection with fully convolutional networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 4296–4303.

[96] K. Charalampous, I. Kostavelis, and A. Gasteratos, "Context-dependent social mapping," in *Proc. IEEE Int. Conf. Imag. Syst. Tech. (IST)*, Chania, Greece, Oct. 2016, pp. 30–35.

[97] M. Giering, V. Venugopalan, and K. Reddy, "Multi-modal sensor registration for vehicle perception via deep neural networks," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Waltham, MA, USA, Sep. 2015, pp. 1–6.

[98] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition," in *Proc. ICASSP*, Vancouver, BC, Canada, May 2013, pp. 3687–3691.

[99] R. Qian, Y. Yue, F. Coenen, and B. Zhang, "Traffic sign recognition with convolutional neural network based on max pooling positions," in *Proc. 12th Int. Conf. Nat. Comput. Fuzzy Syst. Knowl. Disc. (ICNC-FSKD)*, Changsha, China, Aug. 2016, pp. 578–582.

[100] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[101] *The Game Imitation: A Portable Deep Learning Model for Modern Gaming AI*. Accessed Dec. 2016. [Online]. Available: http://cs231n.stanford.edu/reports2016/113_Report.pdf

[102] A. Mestres *et al.*, "Knowledge-defined networking," *CoRR*, vol. abs/1606.06222, pp. 1–8, Nov. 2016.

[103] (2016). *Deep Learning Comp Sheet: Deeplearning4j vs. Torch vs. Theano vs. Caffe vs. Tensorflow*. Accessed Aug. 2016. [Online]. Available: http://deeplearning4j.org/compare-dl4j-torch7-pylearn.html

[104] T. White, *Hadoop: The Definitive Guide*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2009.

[105] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput. (HotCloud)*, Berkeley, CA, USA, 2010, p. 10.

[106] (2016). *Will API*. Accessed Aug. 2016. [Online]. Available: https://scarsty.gitbooks.io/will/content/

[107] N. Kato *et al.*, "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wireless Commun.*, to be published, doi: 10.1109/MWC.2016.1600317WC.

[108] S. Chintala. *Understanding Natural Language With Deep Neural Networks Using Torch*. Accessed Dec. 2016. [Online]. Available: https://devblogs.nvidia.com/parallelforall/understanding-natural-language-deep-neural-networks-using-torch/

[109] P. Roquero, J. Ramos, V. Moreno, I. González, and J. Aracil, "High-speed TCP flow record extraction using GPUs," *J. Supercomput.*, vol. 71, no. 10, pp. 3851–3876, Oct. 2015.

[110] A. E. A. A. Abdulla, H. Nishiyama, J. Yang, N. Ansari, and N. Kato, "HYMN: A novel hybrid multi-hop routing algorithm to improve the longevity of WSNs," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2531–2541, Jul. 2012.

[111] H. Nakayama, Z. M. Fadlullah, N. Ansari, and N. Kato, "A novel scheme for WSAN sink mobility based on clustering and set packing techniques," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2381–2389, Oct. 2011.

[112] K. Suto, H. Nishiyama, N. Kato, and C.-W. Huang, "An energy-efficient and delay-aware wireless computing system for industrial wireless sensor networks," *IEEE Access*, vol. 3, pp. 1026–1035, Jul. 2015.

[113] A. E. A. A. Abdulla, H. Nishiyama, and N. Kato, "Extending the lifetime of wireless sensor networks: A hybrid routing algorithm," *Comput. Commun. J.*, vol. 35, no. 9, pp. 1056–1063, May 2012.

[114] H. Nakayama, N. Ansari, A. Jamalipour, and N. Kato, "Fault-resilient sensing in wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 11–12, pp. 2375–2384, Sep. 2007.

[115] Y. Kawamoto, H. Nishiyama, Z. M. Fadlullah, and N. Kato, "Effective data collection via satellite-routed sensor system (SRSS) to realize global-scaled Internet of Things," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3645–3654, Oct. 2013.

[116] D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "Toward energy efficient big data gathering in densely distributed sensor networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 388–397, Sep. 2014.

[117] H. Nishiyama, T. Ngo, N. Ansari, and N. Kato, "On minimizing the impact of mobility on topology control in mobile ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 1158–1166, Mar. 2012.

[118] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *CoRR*, vol. abs/1405, no. 4463, pp. 1–23, May 2014.

[119] G. E. P. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis*, vol. 40. New York, NY, USA: Wiley, 2011.

[120] C. E. Rasmussen, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, Feb. 2006.

[121] A. Shareef, Y. Zhu, and M. Musavi, "Localization using neural networks in wireless sensor networks," in *Proc. 1st Int. Conf. Mobile Wireless Middleware Oper. Syst. Appl.*, Innsbruck, Austria, 2007, pp. 1–7.

[122] D. Fontanella, M. Nicoli, and L. Vandendorpe, "Bayesian localization in sensor networks: Distributed algorithm and fundamental limits," in *Proc. IEEE Int. Conf. Commun.*, Cape Town, South Africa, May 2010, pp. 1–5.

[123] C.-H. Lu and L.-C. Fu, "Robust location-aware activity recognition using wireless sensor network in an attentive home," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 4, pp. 598–609, Oct. 2009.

[124] R. V. Kulkarni and G. K. Venayagamoorthy, "Neural network based secure media access control protocol for wireless sensor networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Atlanta, GA, USA, Feb. 2009, pp. 1680–1687.

[125] M. H. Kim and M.-G. Park, "Bayesian statistical modeling of system energy saving effectiveness for MAC protocols of wireless sensor networks," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, vol. 209. Heidelberg, Germany: Springer, Jun. 2009, pp. 233–245.

[126] Y.-J. Shen and M.-S. Wang, "Broadcast scheduling in wireless sensor networks using fuzzy hopfield neural network," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 900–907, Feb. 2008.

[127] S. Kaplantzis, A. Shilton, N. Mani, and Y. Sekercioglu, "Detecting selective forwarding attacks in wireless sensor networks using support vector machines," in *Proc. 3rd Int. Conf. Intell. Sensors Sensor Netw. Inf.*, Melbourne, VIC, Australia, Dec. 2007, pp. 335–340.

[128] D. Janakiram, V. A. M. Reddy, and A. V. U. P. Kumar, "Outlier detection in wireless sensor networks using Bayesian belief networks," in *Proc. 1st Int. Conf. Commun. Syst. Softw. Middleware*, New Delhi, India, Jan. 2006, pp. 1–6.

[129] J. W. Branch, C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta, "In-network outlier detection in wireless sensor networks," *Knowl. Inf. Syst.*, vol. 34, no. 1, pp. 23–54, Jul. 2013.

[130] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *Proc. Int. Conf. Commun.*, Glasgow, U.K., Jun. 2007, pp. 3864–3869.

[131] P. P. Jayaraman, A. Zaslavsky, and J. Delsing, *Intelligent Processing of K-Nearest Neighbors Queries Using Mobile Data Collectors in a Location Aware 3D Wireless Sensor Network*. Córdoba, Spain: Springer, 2010.

[132] J. Winter, Y. Xu, and W.-C. Lee, "Energy efficient processing of K nearest neighbor queries in location-aware sensor networks," in *Proc. 2nd Int. Conf. Mobile Ubiquitous Syst. Netw. Services*, San Diego, CA, USA, Jul. 2005, pp. 281–292.

[133] L. Yu, N. Wang, and X. Meng, "Real-time forest fire detection with wireless sensor networks," in *Proc. Int. Conf. Wireless Commun. Netw. Mobile Comput.*, vol. 2. Wuhan, China, Sep. 2005, pp. 1214–1217.

[134] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 5, pp. 981–988, May 2008.

[135] A. Snow, P. Rastogi, and G. Weckman, "Assessing dependability of wireless networks using neural networks," in *Proc. Mil. Commun. Conf.*, vol. 5. Atlantic City, NJ, USA, Oct. 2005, pp. 2809–2815.

[136] Y. Wang, M. Martonosi, and L.-S. Peh, "Predicting link quality using supervised learning in wireless sensor networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 11, no. 3, pp. 71–83, Jul. 2007.

[137] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'nearest neighbor' meaningful?" in *Database Theory*. Heidelberg, Germany: Springer, 1999.

[138] T. O. Ayodele, *Types of Machine Learning Algorithms*. Portsmouth, U.K.: InTech, Feb. 2010.

[139] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 660–674, May/Jun. 1991.

[140] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. 4, no. 2, pp. 4–22, Apr. 1987.

[141] T. Kohonen, "Self-organizing maps," *Neurocomputing*, vol. 21, nos. 1–3, pp. 19–30, Mar. 1998.

[142] I. Steinwart and A. Christmann, *Support Vector Machines*. Berlin, Germany: Springer, 2008.

[143] B. Yang, J. Yang, J. Xu, and D. Yang, *Area Localization Algorithm for Mobile Nodes in Wireless Sensor Networks Based on Support Vector Machines* (Mobile Ad-Hoc and Sensor Networks). Berlin, Germany, Springer, 2007.

[144] W. Kim, J. Park, and H. J. Kim, "Target localization using ensemble support vector regression in wireless sensor networks," in *Proc. Wireless Commun. Netw. Conf.*, Sydney, NSW, Australia, Apr. 2010, pp. 1–5.

[145] D. A. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 7, pp. 981–994, Jul. 2008.

[146] Y. Chen, Y. Qin, Y. Xiang, J. Zhong, and X. Jiao, *Intrusion Detection System Based on Immune Algorithm and Support Vector Machine in Wireless Sensor Network* (Communications in Computer and Information Science), vol. 86. Heidelberg, Germany: Springer, May 2011, pp. 372–376.

[147] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1062–1074, May 2013.

[148] Z. Yang, N. Meratnia, and P. Havinga, "An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine," in *Proc. IEEE Int. Conf. Intell. Sensors Sensor Netw. Inf. Process.*, Sydney, NSW, Australia, Dec. 2008, pp. 151–156.

[149] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, Mar. 2002.

[150] S. V. Macua, P. Belanovic, and S. Zazo, "Consensus-based distributed principal component analysis in wireless sensor networks," in *Proc. 11th Int. Workshop Signal Process. Adv. Wireless Commun.*, Marrakesh, Morocco, Jun. 2010, pp. 1–5.

[151] Y.-C. Tseng, Y.-C. Wang, K.-Y. Cheng, and Y.-Y. Hsieh, "iMouse: An integrated mobile surveillance and wireless sensor system," *Computer*, vol. 40, no. 6, pp. 60–66, Jun. 2007.

[152] A. Rooshenas, H. R. Rabiee, A. Movaghar, and M. Y. Naderi, "Reducing the data transmission in wireless sensor networks using the principal component analysis," in *Proc. 6th Int. IEEE Conf. Intell. Sensors Sensor Netw. Inf. Process.*, Brisbane, QLD, Australia, Dec. 2010, pp. 133–138.

[153] R. Masiero *et al.*, "Data acquisition through joint compressive sensing and principal component analysis," in *Proc. IEEE Glob. Telecommun. Conf.*, Honolulu, HI, USA, Nov./Dec. 2009, pp. 1–6.

[154] R. Masiero, G. Quer, M. Rossi, and M. Zorzi, "A Bayesian analysis of compressive sensing data recovery in wireless sensor networks," in *Proc. Int. Conf. Ultra Mod. Telecommun. Workshops*, St. Petersburg, Russia, Oct. 2009, pp. 1–6.

[155] S. Lee and T. Chung, "Data aggregation for wireless sensor networks using self-organizing map," in *Artificial Intelligence and Simulation* (LNCS 3397). Heidelberg, Germany: Springer, Nov. 2005, pp. 508–517.

[156] T. Kanungo *et al.*, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[157] I. T. Jolliffe, *Principal Component Analysis*. Berlin, Germany: Springer-Verlag, 2002.

[158] R. Arroyo-Valles, R. Alaiz-Rodriguez, A. Guerrero-Curieses, and J. Cid-Sueiro, "Q-probabilistic routing in wireless sensor networks," in *Proc. 3rd Int. Conf. Intell. Sensors Sensor Netw. Inf.*, Melbourne VIC, Australia, Dec. 2007, pp. 1–6.

[159] S. Dong, P. Agrawal, and K. Sivalingam, "Reinforcement learning based geographic routing protocol for UWB wireless sensor network," in *Proc. IEEE Glob. Telecommun. Conf.*, Washington, DC, USA, Nov. 2007, pp. 652–656.

[160] A. Forster and A. L. Murphy, "FROMS: Feedback routing for optimizing multiple sinks in WSN with reinforcement learning," in *Proc. 3rd Int. Conf. Intell. Sensors Sensor Netw. Inf.*, Melbourne, VIC, Australia, Dec. 2007, pp. 371–376.

[161] R. Sun, S. Tatsumi, and G. Zhao, "Q-MAP: A novel multicast routing method in wireless ad hoc networks with multiagent reinforcement learning," in *Proc. Region 10 Conf. Comput. Commun. Control Power Eng.*, vol. 1. Beijing, China, Oct. 2002, pp. 667–670.

[162] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, Jun. 2013.

[163] T. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.

[164] L. Grimaudo, M. Mellia, E. Baralis, and R. Keralapura, "SeLeCT: Self-learning classifier for Internet traffic," *IEEE Trans. Netw. Service Manag.*, vol. 11, no. 2, pp. 144–157, Jun. 2014.

[165] R. Ettiane, A. Chaoub, and R. Elkouch, "Enhanced traffic classification design through a randomized approach for more secure 3G mobile networks," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Fes, Morocco, Oct. 2016, pp. 116–121.

[166] D. Zuev and A. W. Moore, "Traffic classification using a statistical approach," in *Proc. Int. Workshop Passive Active Netw. Meas.*, Boston, MA, USA, 2005, pp. 321–324.

[167] H.-J. Kang, M.-S. Kim, and J. W.-K. Hong, "A method on multimedia service traffic monitoring and analysis," in *Proc. Int. Workshop Distrib. Syst. Oper. Manag.*, 2003, pp. 93–105.

[168] J. van der Merwe, R. Cáceres, Y.-H. Chu, and C. Sreenan, "Mmdump: A tool for monitoring Internet multimedia traffic," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 5, pp. 48–59, Oct. 2000.

[169] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in *Proc. 13th Int. Conf. World Wide Web*, New York, NY, USA, May 2004, pp. 512–521.

[170] S. Zander, T. T. T. Nguyen, and G. J. Armitage, "Self-learning IP traffic classification based on statistical flow characteristics," in *Proc. Int. Workshop Passive Active Netw. Meas.*, Boston, MA, USA, 2005, pp. 325–328.

[171] S. Sen and W. Jia, "Analyzing peer-to-peer traffic across large networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 219–232, Apr. 2004.

[172] P. Wang, S.-C. Lin, and M. Luo, "A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, San Francisco, CA, USA, Jun. 2016, pp. 760–765.

[173] T. T. T. Nguyen and G. J. Armitage, "Training on multiple sub-flows to optimise the use of Machine Learning classifiers in real-world IP networks," in *Proc. 31st IEEE Conf. Local Comput. Netw.*, Tampa, FL, USA, Nov. 2006, pp. 369–376.

[174] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas.*, Taormina, Italy, Oct. 2004, pp. 135–148.

[175] A. W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 50–60, Jun. 2005.

[176] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proc. SIGCOMM Workshop Min. Netw. Data*, Pisa, Italy, Sep. 2006, pp. 281–286.

[177] K. M. C. Tan and B. S. Collie, "Detection and classification of TCP/IP network services," in *Proc. 13th Annu. Comput. Security Appl. Conf.*, San Diego, CA, USA, Dec. 1997, pp. 99–107.

[178] J. P. Early, C. E. Brodley, and C. Rosenberg, "Behavioral authentication of server flows," in *Proc. 19th Annu. Comput. Security Appl. Conf.*, Las Vegas, NV, USA, Jul. 2003, pp. 46–55.

[179] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 5–16, Oct. 2006.

[180] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/realtime traffic classification using semi-supervised learning," *Perform. Eval.*, vol. 64, no. 9, pp. 1194–1213, Oct. 2007.

[181] Z. Wang. *The Applications of Deep Learning on Traffic Identification*. Accessed Dec. 2016. [Online]. Available: https://www.blackhat.com/docs/us-15/materials/us-15-Wang-The-Applications-Of-Deep-Learning-On-Traffic-Identification-wp.pdf

[182] T. P. Oliveira, J. S. Barbar, and A. S. Soares, *Multilayer Perceptron and Stacked Autoencoder for Internet Traffic Prediction*. Heidelberg, Germany: Springer, Sep. 2014, pp. 61–71.

[183] S. Basu, A. Mukherjee, and S. Klivansky, "Time series models for Internet traffic," in *Proc. IEEE INFOCOM*, vol. 2. San Francisco, CA, USA, Mar. 1996, pp. 611–620.

[184] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003, pp. 325–336.

[185] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[186] M. Rresenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nat. Neurosci.*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999.

[187] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. ICML*, Corvallis, OR, USA, Jun. 2007, pp. 759–766.

[188] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: Global versus component-based approach," in *Proc. ICCV*, Vancouver, BC, Canada, Jul. 2001, pp. 688–694.

[189] A. Coates and A. Y. Ng, "Selecting receptive fields in deep networks," in *Proc. NIPS*, Granada, Spain, Dec. 2011, pp. 2528–2536.

[190] T. Lin, S. Liu, and H. Zha, "Incoherent dictionary learning for sparse representation," in *Proc. ICPR*, vol. 14. Tsukuba, Japan, 2012, pp. 1237–1240.

[191] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 3501–3508.

[192] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 2691–2698.

[193] R. Hyndman. *DataMarket: Data Library (TSDL)*. Accessed Dec. 2016. [Online]. Available: https://datamarket.com/data/list/?q= provider:tsdl

[194] Y. Jia *et al.*, "Fusing social networks with deep learning for volunteerism tendency prediction," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 165–171.

[195] S. Gao, H. Pang, P. Gallinari, J. Guo, and N. Kato, "A novel embedding method for information diffusion prediction in social network big data," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2017.2684160.

[196] C. Curme, T. Preis, H. E. Stanley, and H. S. Moat, "Quantifying the semantics of search behavior before stock market moves," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 32, pp. 11600–11605, Aug. 2014.

[197] K. Dembczynski, W. Kotlowski, and D. Weiss, "Predicting ads click-through rate with decision rules," in *Proc. Workshop Target. Ranking Online Advertising*, Beijing, China, Apr. 2008.

[198] J. B. Kim, P. Albuquerque, and B. J. Bronnenberg, "Online demand under limited consumer search," *Market. Sci.*, vol. 29, no. 6, pp. 1001–1023, Nov./Dec. 2010.

[199] A. Vieira, "Predicting online user behaviour using deep learning algorithms," *CoRR*, vol. abs/1511, no. 06247, pp. 1–21, Nov. 2015.

[200] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD)*, New York, NY, USA, Aug. 2014, pp. 701–710.

[201] L. Tang and H. Liu, "Leveraging social media networks for classification," *Data Min. Knowl. Disc.*, vol. 23, no. 3, pp. 447–478, Nov. 2011.

[202] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *Proc. 15th ACM SIGKDD (KDD)*, Paris, France, Jun./Jul. 2009, pp. 817–826.

[203] L. Tang and H. Liu, "Scalable learning of collective behavior based on sparse social dimensions," in *Proc. 18th ACM Conf. Inf. Knowl. Manag.*, Hong Kong, Nov. 2009, pp. 1107–1116.

[204] S. A. Macskassy and F. Provost, "A simple relational classifier," in *Proc. 2nd Workshop Multi Relational Data Min. (MRDM-KDD)*, New York, NY, USA, Dec. 2003, pp. 64–76.

[205] M. B. Lazreg, M. Goodwin, and O.-C. Granmo, "Deep learning for social media analysis in crises situations (position paper)," in *Proc. 29th Annu. Workshop Swedish Artif. Intell. Soc. (SAIS)*, Malmö, Sweden, Jun. 2016, pp. 31–36.

[206] M.-A. Russon. *Using Deep Learning Neural Networks to Predict Social Unrest Five Days Before it Happens*. Accessed Dec. 2016. [Online]. Available: http://www.ibtimes.co.uk/cia-using-deep-learning-neural-networks-predict-social-unrest-five-days-before-it-happens-1585115

[207] *The Zettabyte Erajtrends and Analysis*. Accessed Nov. 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

[208] J. Moy, *OSPF Version 2*, Ascend Commun., Alameda, CA, USA, Apr. 1998.

[209] G. Rétvári and T. Cinkler, "Practical OSPF traffic engineering," *IEEE Commun. Lett.*, vol. 8, no. 11, pp. 689–691, Nov. 2004.

[210] *The Zettabyte Era-Trends and Analysis*. Accessed Nov. 2016. [Online]. Available: http://neuralnetworksanddeeplearning.com/ chap2.html

[211] A. Raniwala and T.-C. Chiueh, "Evaluation of a wireless enterprise backbone network architecture," in *Proc. 12th Annu. IEEE Symp. High Perform. Interconnects*, Stanford, CA, USA, Aug. 2004, pp. 98–104.

[212] B. Mao *et al.*, "Routing or computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning," *IEEE Trans. Comput.*, to be published, doi: 10.1109/TC.2017.2709742.

[213] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 20–29, Jun. 2004.

[214] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *J. Artif. Intell. Res.*, vol. 11, no. 1, pp. 131–167, Jul. 1999.

[215] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proc. ACL Conf. Short Papers (ACLShort)*, Uppsala, Sweden, Apr. 2010, pp. 220–224.

[216] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD)*, San Diego, CA, USA, 2011, pp. 1100–1108.

[217] S. Jiang, J. Ferreira, and M. C. Gonzales, "Activity-based human mobility patterns inferred from mobile phone data: A case study of Singapore," *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 208–219, May 2017.

[218] S. Hasan, X. Zhan, and S. V. Ukkusuri, "Understanding urban human activity and mobility patterns using large-scale location-based data from online social media," in *Proc. 2nd ACM SIGKDD Int. Workshop Urban Comput. (UrbComp)*, Chicago, IL, USA, 2013, pp. 1–8.

[219] K. Zhao, S. Tarkoma, S. Liu, and H. Vo, "Urban human mobility data mining: An overview," in *Proc. IEEE Int. Conf. Big Data*, Washington, DC, USA, Dec. 2016, pp. 1911–1920.

[220] Y. Chon, E. Talipov, H. Shin, and H. Cha, "Mobility prediction-based smartphone energy optimization for everyday location monitoring," in *Proc. 9th ACM Conf. Embedded Netw. Sensor Syst.*, Seattle, WA, USA, 2011, pp. 82–95.

[221] V. Belik, T. Geisel, and D. Brockmann, "Natural human mobility patterns and spatial spread of infectious diseases," *Phys. Rev. X*, vol. 1, no. 1, pp. 1–11, Aug. 2011.

[222] S. Ni and W. Weng, "Impact of travel patterns on epidemic dynamics in heterogeneous spatial metapopulation networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 79, no. 1, Jan. 2009, Art. no. 016111.

[223] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and POIs," in *Proc. SIGKDD*, Beijing, China, Aug. 2012, pp. 186–194.

[224] G. Qi *et al.*, "Measuring social functions of city regions from large-scale taxi behaviors," in *Proc. IEEE PerCom*, Seattle, WA, USA, Mar. 2011, pp. 384–388.

[225] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proc. 13th ACM Int. Conf. Ubiquitous Comput.*, Beijing, China, Sep. 2011, pp. 89–98.

[226] S. Goh, K. Lee, J. S. Park, and M. Y. Choi, "Modification of the gravity model and application to the metropolitan Seoul subway system," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 86, Aug. 2012, Art. no. 026102.

[227] W.-S. Jung, F. Wang, and H. E. Stanley, "Gravity model in the Korean highway," *Europhys. Lett.*, vol. 81, no. 4, Jan. 2008, Art. no. 48005.

[228] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Orlando, FL, USA, Nov. 2013, pp. 344–353.

[229] W. Rao, K. Zhao, E. Lagerspetz, P. Hui, and S. Tarkoma, "Energy-aware keyword search on mobile phones," in *Proc. MCC SIGCOMM*, Helsinki, Finland, Aug. 2012, pp. 59–64.

[230] K. Zhao, M. P. Chinnasamy, and S. Tarkoma, "Automatic city region analysis for urban routing," in *Proc. IEEE ICDMW*, Atlantic City, NJ, USA, Nov. 2015, pp. 1136–1142.

[231] P. Sundsøy, J. Bjelland, B.-A. Reme, A. M. Iqbal, and E. Jahani, "Deep learning applied to mobile phone data for individual income classification," in *Proc. 5th Int. Conf. Artif. Intell. Technol. Appl. (ICAITA)*, Dubai, UAE, Nov. 2016, pp. 96–99.

[232] L. Ghouti, "Mobility prediction using fully-complex extreme learning machines," in *Proc. Eur. Symp. Artif. Neural Netw.*, Bruges, Belgium, Apr. 2014, pp. 607–612.

[233] G.-B. Huang, M.-B. Li, L. Chen, and C.-K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, nos. 4–6, pp. 576–583, Jan. 2008.

[234] X. Song, H. Kanasugi, and R. Shibasaki, "DeepTransport: Prediction and simulation of human mobility and transportation mode at a citywide level," in *Proc. IJCAI*, New York, NY, USA, Jul. 2016, pp. 2618–2624.

[235] X. Ouyang, C. Zhang, P. Zhou, and H. Jiang, "Deepspace: An online deep learning framework for mobile big data to understand human mobility patterns," *CoRR*, vol. abs/1610, no. 07009, pp. 1–11, Nov. 2016.

[236] G. Zhou, K. Sohn, and H. Lee, "Online incremental feature learning with denoising autoencoders," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, Apr. 2012, pp. 1453–1461.

[237] T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang, "Error-driven incremental learning in deep convolutional neural network for large-scale image classification," in *Proc. ACM Int. Conf. Multimedia (MM)*, Orlando, FL, USA, Nov. 2014, pp. 177–186.

[238] I. Konno, H. Nishiyama, and N. Kato, "An adaptive media access control mechanism for cognitive radio," *IEICE Trans. Commun.*, vol. J94-B, no. 2, pp. 253–263, Feb. 2011.

[239] Z. M. Fadlullah, H. Nishiyama, N. Kato, and M. M. Fouda, "Intrusion detection system (IDS) for combating attacks against cognitive radio networks," *IEEE Netw. Mag.*, vol. 27, no. 3, pp. 51–56, May/Jun. 2013.

[240] T. W. Rondeau, B. Le, C. J. Rieser, and C. W. Bostian, "Cognitive radios with genetic algorithms: Intelligent control of software defined radios," in *Proc. Softw. Defined Radio Forum Tech. Conf.*, Phoenix, AZ, USA, Nov. 2004, pp. C3–C8.

[241] T. W. Rondeau and C. W. Bostian, "Cognitive techniques: Physical and link layers," in *Cognitive Radio Technology*, B. A. Fette, Ed. Burlington, MA, USA: Newnes, Aug. 2006, pp. 219–268.

[242] *Working Group on Wireless Local Area Networks*, IEEE Standard 802.11, Dec. 2016. [Online]. Available: http://www.ieee802.org/11/

[243] C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of machine learning to cognitive radio networks," *IEEE Wireless Commun.*, vol. 14, no. 4, pp. 47–52, Aug. 2007.

[244] N. Abbas, Y. Nasser, and K. E. Ahmad, "Recent advances on artificial intelligence and learning techniques in cognitive radio networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, p. 174, Dec. 2015.

[245] M. Zorzi, A. Zanella, A. Testolin, M. D. F. D. Grazia, and M. Zorzi, "Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence," *IEEE Access*, vol. 3, pp. 1512–1530, Nov. 2015.

[246] T. J. O'Shea and J. Corgan, "Convolutional radio modulation recognition networks," *CoRR*, vol. abs/1602, no. 04105, pp. 1–15, Mar. 2016.

[247] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, Montreal, QC, Canada, 2009, pp. 873–880. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553486

[248] J. Dean *et al.*, "Large scale distributed deep networks," in *Proc. NIPS*, 2012, pp. 1223–1231.

[249] A. Taherpour, S. Gazor, and A. Taherpour, "Adaptive spectrum sensing and learning in cognitive radio networks," in *Proc. 18th Eur. Signal Process. Conf.*, Aalborg, Denmark, Aug. 2010, pp. 860–864.

[250] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 336–361, 1st Quart., 2013.

[251] N. Agoulmine, *Autonomic Network Management Principles: From Concepts to Applications*. Burlington, MA, USA: Academic, Oct. 2011.

[252] M. Dirani and Z. Altman, "A cooperative reinforcement learning approach for inter-cell interference coordination in OFDMA cellular networks," in *Proc. IEEE 8th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, Avignon, France, May 2010, pp. 170–176.

[253] R. Razavi, S. Klein, and H. Claussen, "Self-optimization of capacity and coverage in LTE networks using a fuzzy reinforcement learning approach," in *Proc. IEEE 21st Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Istanbul, Turkey, Sep. 2010, pp. 1865–1870.

[254] R. Razavi, S. Klein, and H. Claussen, "A fuzzy reinforcement learning approach for self-optimization of coverage in LTE networks," *Bell Labs Tech. J.*, vol. 15, no. 3, pp. 153–175, Dec. 2010.

[255] M. N. ul Islam and A. Mitschele-Thiel, "Reinforcement learning strategies for self-organized coverage and capacity optimization," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Paris, France, Apr. 2012, pp. 2818–2823.

[256] N. Morozs, T. Clarke, and D. Grace, "Heuristically accelerated reinforcement learning for dynamic secondary spectrum sharing," *IEEE Access*, vol. 3, pp. 2771–2783, Dec. 2015.

[257] S. Han, K. Jang, K. Park, and S. Moon, "Building a single-box 100 Gbps software router," in *Proc. 17th IEEE Workshop Local Metropolitan Area Netw. (LANMAN)*, Long Branch, NJ, USA, May 2010, pp. 1–4.

[258] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Seoul, South Korea, Mar. 2014, pp. 287–292.

[259] (2016). *Intel Autonomous Cars*. Accessed Dec. 2016. http://www.intel.eu/content/www/eu/en/it-managers/autonamous-cars.html

[260] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *Int. J. Security Appl.*, vol. 9, no. 5, pp. 205–216, 2015.

Zubair Md. Fadlullah (M'11–SM'13) received the B.Sc. (Hons.) degree in computer science and information technology from the Islamic University of Technology, Bangladesh, in 2003, and the M.Sc. and Ph.D. degrees in applied information science from Tohoku University in 2008 and 2011, respectively. He is currently an Associate Professor with the Graduate School of Information Sciences, Tohoku University, Japan. His research interests are in the areas of 5G, smart grid, network security, intrusion detection, game theory, quality of security service provisioning mechanism, and deep learning. He was a recipient of the Dean's Award and the President's Award from Tohoku University in 2011, the IEEE Asia Pacific Outstanding Researcher Award in 2015, the NEC Foundation Prize for research contributions in 2016, and several best paper awards in the Globecom, IC-NIDC, and IWCMC conferences.

Fengxiao Tang (S'15) received the B.E. degree in measurement and control technology and instrument from the Wuhan University of Technology, Wuhan, China, in 2012, and the M.S. degree in software engineering from the Central South University, Changsha, China, in 2015. He is currently pursuing the Ph.D. degree with the GSIS, Tohoku University, Japan. His research interests are unmanned aerial vehicles system, game theory optimization, and deep learning.

Bomin Mao (S'15) received the B.Sc. degree in telecommunications engineering and the M.S. degree in electronics and telecommunications engineering with Xidian University, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with GSIS, Tohoku University, Japan. His research interests are involving wireless networks, software defined networks, and quality of service, particularly with applications of machine intelligence and deep learning.

Nei Kato (F'13) is a Full Professor and the Director of Research Organization of Electrical Communication, Tohoku University, Japan. He has been engaged in research on computer networking, wireless mobile communications, satellite communications, ad hoc and sensor and mesh networks, smart grid, IoT, Big Data, and pattern recognition. He has published over 350 papers in prestigious peer-reviewed journals and conferences. Since 2015, he has been the Editor-in-Chief of *IEEE Network Magazine*, has been the Associate Editor-in-Chief of IEEE INTERNET OF THINGS JOURNAL since 2013, an Area Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY since 2014, and the Chair of IEEE Communications Society, Sendai Chapter. He served as a Member-at-Large on the Board of Governors, IEEE Communications Society from 2014 to 2016, a Vice Chair of Fellow Committee of IEEE Computer Society in 2016, a member of the IEEE Computer Society Award Committee from 2015 to 2016, and the IEEE Communications Society Award Committee from 2015 to 2017. He has also served as the Chair of Satellite and Space Communications Technical Committee from 2010 to 2012, and Ad Hoc and Sensor Networks Technical Committee of IEEE Communications Society from 2014 to 2015. He was a recipient of the Minoru Ishida Foundation Research Encouragement Prize in 2003, the Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, Satellite and Space Communications Technical Committee in 2005, the FUNAI information Science Award in 2007, the TELCOM System Technology Award from Foundation for Electrical Communications Diffusion in 2008, the IEICE Network System Research Award in 2009, the IEICE Satellite Communications Research Award in 2011, the KDDI Foundation Excellent Research Award in 2012, the IEICE Communications Society Distinguished Service Award in 2012, the IEICE Communications Society Best Paper Award in 2012, the Distinguished Contributions to Disaster-Resilient Networks Research and Development Award from Ministry of Internal Affairs and Communications, Japan in 2014, the Outstanding Service and Leadership Recognition Award 2016 from IEEE Communications Society Ad Hoc and Sensor Networks Technical Committee, the Radio Achievements Award from Ministry of Internal Affairs and Communications, Japan, in 2016, and the Best Paper Awards from IEEE ICC/GLOBECOM/WCNC/VTC. He is a Distinguished Lecturer of the IEEE Communications Society and Vehicular Technology Society. He is a fellow of the IEICE.

Osamu Akashi received the B.Sc. and M.Sc. degrees in information science, and the Ph.D. degree in mathematical and computing sciences from Tokyo Institute of Technology in 1987, 1989, and 2001, respectively. He joined the Nippon Telegraph and Telephone Corporation Software Laboratories in 1989, and is a Senior Research Scientist with the NTT Network Innovation Laboratories. His research interests are in the areas of distributed systems, multiagent systems, and network architectures. He is a member of ACM, IEICE, IPSJ, and JSSST.

Takeru Inoue received the B.E., M.E., and Ph.D. degrees from Kyoto University, Japan, in 1998, 2000, and 2006, respectively. He is a Distinguished Researcher with NTT Network Innovation Laboratories. He was an ERATO Researcher with the Japan Science and Technology Agency from 2011 to 2013. His research interests widely cover algorithmic approaches in computer networks.

Kimihiro Mizutani (M'11) received the M.S. degree in information system from the Nara Institute Science and Technology in 2010. He is a Researcher with NTT Network Innovation Laboratory. His research interest is future Internet architecture. He was a recipient of the Best Student Paper from International Conference on Communication Systems and Application in 2010, and research awards from IPSJ and IEICE in 2010 and 2013, respectively. He is a member of the IEICE and IEEE Communication Society.