Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges

MUHAMMAD USAMA[1], JUNAID QADIR[1], AUNN RAZA[2],HUNAIN ARIF[2], KOK-LIM ALVIN YAU[3],YEHIA ELKHATIB[4], AMIR HUSSAIN[5],and ALA-AL-FUQAHA.[6]

[1]Information Technology University (ITU)-Punjab, Lahore, Pakistan
[2]National University of Science and Technology (NUST), Pakistan
[3]Sunway University, Malaysia
[4]MetaLab, School of Computing and Communication, Lancaster University, UK
[5]Edinburgh Napier University, UK
[6]Hamad Bin Khalifa University, Qatar

Corresponding author: Muhammad Usama (e-mail: muhammad.usama@itu.edu.pk).

ABSTRACT While machine learning and artificial intelligence have long been applied in networking research, the bulk of such works has focused on supervised learning. Recently, there has been a rising trend of employing unsupervised machine learning using unstructured raw network data to improve network performance and provide services such as traffic engineering, anomaly detection, Internet traffic classification, and quality of service optimization. The interest in applying unsupervised learning techniques in networking emerges from their great success in other fields such as computer vision, natural language processing, speech recognition, and optimal control (e.g., for developing autonomous self-driving cars). Unsupervised learning is interesting since it can unconstrain us from the need for labeled data and manual handcrafted feature engineering thereby facilitating flexible, general, and automated methods of machine learning. The focus of this survey paper is to provide an overview of the applications of unsupervised learning in the domain of networking. We provide a comprehensive survey highlighting the recent advancements in unsupervised learning techniques and describe their applications in various learning tasks in the context of networking. We also provide a discussion on future directions and open research issues, while also identifying potential pitfalls. While a few survey papers focusing on the applications of machine learning in networking have previously been published, a survey of similar scope and breadth is missing in the literature. Through this paper, we advance the state of knowledge by carefully synthesizing the insights from these survey papers while also providing contemporary coverage of recent advances.

INDEX TERMS *Machine Learning*, *Deep Learning*, *Unsupervised Learning*, *Computer Networks*

I. INTRODUCTION

Networks—such as the Internet and mobile telecom networks—serve the function of the central hub of modern human societies, which the various threads of modern life weave around. With networks becoming increasingly dynamic, heterogeneous, and complex, the management of such networks has become less amenable to manual administration, and it can benefit from leveraging support from methods for optimization and automated decision-making from the fields of artificial intelligence (AI) and machine learning (ML). Such AI and ML techniques have already transformed multiple fields—e.g., computer vision, natural language processing (NLP), speech recognition, and optimal control (e.g., for developing autonomous self-driving vehicles)—with the success of these techniques mainly attributed to *firstly*, significant advances in unsupervised ML techniques such as deep learning, *secondly*, the ready availability of large amounts of unstructured raw data amenable to processing by unsupervised learning algorithms, and *finally*, advances in computing technologies through advances such as cloud computing, graphics processing unit (GPU) technology and other hardware enhancements. It is anticipated that AI and ML will also make a similar impact on the networking ecosystem and will help realize a future vision of *cognitive networks* [1] [2], in which networks will self-organize and will autonomously implement intelligent network-wide behavior to solve problems such as routing, scheduling, resource allocation, and anomaly detection. The initial attempts towards creating cognitive or intelligent networks have relied mostly on *supervised ML methods*, which are efficient and powerful but are limited

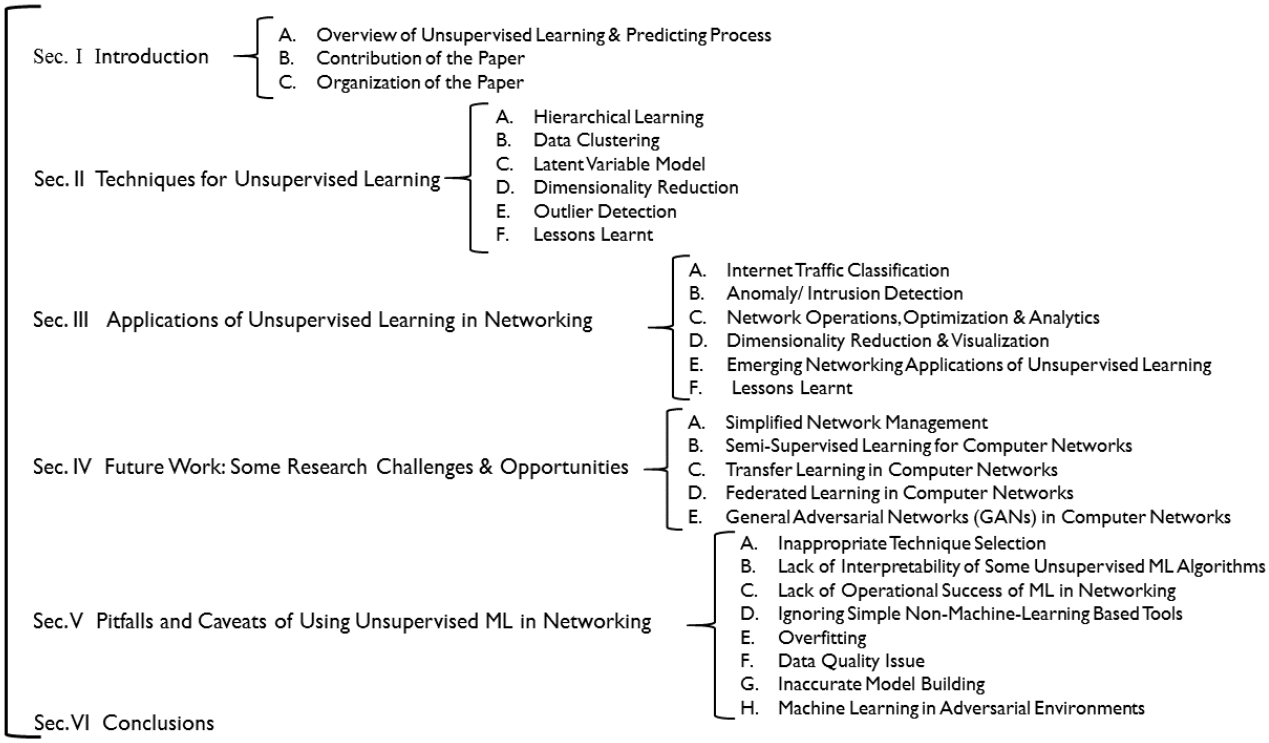

FIGURE 1. Outline of the paper

in scope by their need for labeled data. With network data becoming increasingly voluminous (with a disproportionate rise in unstructured unlabeled data), there is a groundswell of interest in leveraging *unsupervised ML methods* to utilize unlabeled data, in addition to labeled data where available, to optimize network performance [3]. The rising interest in applying unsupervised ML in networking applications also stems from the need to liberate ML applications from restrictive demands of supervised ML. Another reason of employing unsupervised ML in networking is the expensiveness of curating labeled network data at scale, since labeled data may be unavailable and manual annotation is prohibitively inconvenient, in addition, to be outdated quickly (due to the highly dynamic nature of computer networks) [4].

We are already witnessing the failure of human network administrators to manage and monitor all bits and pieces of network [5], and the problem will only exacerbate with further growth in the size of networks with paradigms such as becoming the Internet of things (IoT). An ML-based network management system (NMS) is desirable in such large networks so that faults/bottlenecks/anomalies may be predicted in advance with reasonable accuracy. In this regard, networks already have ample amount of untapped data, which can provide us with decision-making insights making networks more efficient and self-adapting. With unsupervised ML, the pipe dream is that every algorithm for adjusting network parameters (be it, TCP congestion window or rerouting network traffic during peak time) will optimize itself in a self-organizing fashion according to the environment and application, user, and network Quality of Service (QoS) requirements and constraints [6]. Unsupervised ML methods, in concert with existing supervised ML methods, can provide a more efficient method that lets a network manage, monitor, and optimize itself while keeping the human administrators in the loop with the provisioning of timely actionable information.

Next generation networks are expected to be self-driven, which means they have the ability to self configure, optimize, and heal [7]. All these self-driven properties can be achieved by building artificial intelligence in the system using ML techniques. Self-driven networks are supposed to utilize the network data to perform networking chores and most of the network data is imbalanced and unlabeled. In order to develop a reliable data-driven network, data quality must be taken care before subjecting it to an appropriate unsupervised ML [8]. Unsupervised ML techniques facilitate the analysis of raw datasets, thereby helping in generating analytic insights from unlabeled data. Recent advances in hierarchical learning, clustering algorithms, factor analysis, latent models, and outlier detection, have helped significantly advance the state of the art in unsupervised ML techniques. In particular, recent unsupervised ML advances—such as the development of "deep learning" techniques [22]—have however significantly advanced the ML state of the art by facilitating the processing of raw data without requiring careful engineering and domain expertise for feature crafting. Deep

TABLE 1. Comparison of our paper with existing survey and review papers. (Legend: $\surd$ means covered; $\times$ means not covered; $\approx$ means partially covered.)

Survey paper	Published In	Year	# References	Areas Focused	Unsupervised ML	Deep Learning	Pitfalls	Future Challenges
[9]	Elsevier Computer Networks	2007	100	ML for Network Intrusion Detection	$\approx$	$\times$	$\times$	$\surd$
[10]	IEEE COMST	2008	68	ML for Internet Traffic Classification	$\approx$	$\times$	$\times$	$\times$
[11]	IEEE COMST	2013	177	ML for Cognitive Radios	$\approx$	$\times$	$\times$	$\times$
[12]	IEEE COMST	2014	152	ML for WSNs	$\approx$	$\times$	$\times$	$\surd$
[13]	IEEE COMST	2016	113	ML for Cyber Security Intrusion Detection	$\approx$	$\times$	$\times$	$\surd$
[14]	IEEE COMST	2017	269	ML in SONs	$\approx$	$\times$	$\times$	$\surd$
[15]	Springer Book Chapter	2017	16	ML for Anomaly Detection in Industrial Networks	$\approx$	$\times$	$\times$	$\surd$
[16]	IEEE COMST	2017	260	ML for Network Traffic Control	$\approx$	$\surd$	$\times$	$\surd$
[17]	ArXiv	2017	154	ML for Network Intrusion Detection	$\approx$	$\surd$	$\times$	$\times$
[18]	Arxiv	2018	282	ML applications in the internet of things	$\approx$	$\approx$	$\surd$	$\surd$
[19]	Elsevier	2018	148	ML applications in the internet of things	$\approx$	$\approx$	$\times$	$\approx$
[20]	Springer	2018	501	ML in networking	$\surd$	$\approx$	$\times$	$\surd$
[21]	Springer	2018	79	ML applications in the internet of things	$\approx$	$\times$	$\times$	$\approx$
This Paper	-	2019	321	Unsupervised ML in Networking	$\surd$	$\surd$	$\surd$	$\surd$

learning is a class of machine learning, where hierarchical architectures are used for unsupervised feature learning and these learned features are then used for classification and other related tasks [23]. The versatility of deep learning and distributed ML can be seen in the diversity of their applications that range from self-driving cars to the reconstruction of brain circuits [22]. Unsupervised learning is also often used in conjunction with supervised learning in *semi-supervised learning* setting to preprocess the data before analysis and thereby help in crafting a good feature representation and in finding patterns and structures in unlabeled data.

The rapid advances in deep neural networks, the democratization of enormous computing capabilities through cloud computing and distributed computing, and the ability to store and process large swathes of data have motivated a surging interest in applying unsupervised ML techniques in the networking field. The field of networking also appears to be well suited to, and amenable to applications of unsupervised ML techniques, due to the largely distributed decision-making nature of its protocols, the availability of large amounts of network data, and the urgent need for *intelligent/cognitive* networking. Consider the case of routing in networks. Networks these days have evolved to be very complex, and they incorporate multiple physical paths for redundancy and utilize complex routing methodologies to direct the traffic. The application traffic does not always take the optimal path we would expect, leading to unexpected and inefficient routing performance. To tame such complexity, unsupervised ML techniques can autonomously self-organize the network taking into account a number of factors such as real-time network congestion statistics as well as application QoS requirements [24].

The purpose of this paper is to highlight the important advances in unsupervised learning, and after providing a tutorial introduction to these techniques, to review how such techniques have been, or could be, used for various tasks in modern next-generation networks comprising both computer networks as well as mobile telecom networks.

Contribution of the paper: To the best of our knowledge, there does not exist a survey that specifically focuses on the important applications of unsupervised ML techniques in networks, even though a number of surveys exist that focus on specific ML applications pertaining to networking—for instance, surveys on using ML for cognitive radios [11], traffic identification and classification [10], and anomaly detection [9] [15]. Previous survey papers have either focused on specific unsupervised learning techniques (e.g., [25] have provided a survey of the applications of neural networks in wireless networks) or on some specific applications of computer networking ([13] have provided a survey of the applications of ML in cyber intrusion detection). Our survey paper is timely since there is great interest in deploying automated and self-taught unsupervised learning models in the industry and academia. Due to relatively limited applications of unsupervised learning in networking—in particular, the deep learning trend has not yet impacted networking in a major way—unsupervised learning techniques hold a lot of promises for advancing the state of the art in networking in terms of adaptability, flexibility, and efficiency. The novelty of this survey is that it covers many different important appli-

cations of unsupervised ML techniques in computer networks and provides readers with a comprehensive discussion of the unsupervised ML trends, as well as the suitability of various unsupervised ML techniques. A tabulated comparison of our paper with other existing survey and review articles is presented in Table 1.

Organization of the paper: The organization of this paper is depicted in Figure 1. Section II provides a discussion on various unsupervised ML *techniques* (namely, hierarchical learning, data clustering, latent variable models, and outlier detection). Section III presents a survey of the *applications* of unsupervised ML specifically in the domain of computer networks. Section IV describes future work and opportunities with respect to the use of unsupervised ML in future networking. Section V discusses a few major pitfalls of the unsupervised ML approach and its models. Finally, Section VI concludes this paper. For the reader's facilitation, Table 2 shows all the acronyms used in this survey for convenient referencing.

II. TECHNIQUES FOR UNSUPERVISED LEARNING

In this section, we will introduce some widely used unsupervised learning techniques and their applications in computer networks. We have divided unsupervised learning techniques into six major categories: hierarchical learning, data clustering, latent variable models, dimensionality reduction, and outlier detection. Figure 2 depicts a taxonomy of unsupervised learning techniques and also the relevant sections in which these techniques are discussed. To provide a better understanding of the application of unsupervised ML techniques in networking, we have added few subsections highlighting significant applications of unsupervised ML techniques in networking domain.

A. HIERARCHICAL LEARNING

Hierarchical learning is defined as learning simple and complex features from a hierarchy of multiple linear and nonlinear activations. In learning models, a feature is a measurable property of the input data. Desired features are ideally informative, discriminative, and independent. In statistics, features are also known as explanatory (or independent) variables [26]. Feature learning (also known as data representation learning) is a set of techniques that can learn one or more features from input data [27]. It involves the transformation of raw data into a quantifiable and comparable representation, which is specific to the property of the input but general enough for comparison to similar inputs. Conventionally, features are handcrafted specific to the application on hand. It relies on domain knowledge but even then they do not generalize well to the variation of real-world data, which gives rise to automated learning of generalized features from the underlying structure of the input data. Like other learning algorithms, feature learning is also divided among domains of supervised and unsupervised learning depending on the type of available data. Almost all unsupervised learning algorithms undergo a stage of feature extraction in order to learn data representation from unlabeled data and generate a feature vector on the basis of which further tasks are performed.

Hierarchical learning is intimately related to how deep learning is performed in modern multi-layer neural networks. In particular, deep learning techniques benefits from the fundamental concept of artificial neural networks (ANNs), a deep structure consists of multiple hidden layers with multiple neurons in each layer, a nonlinear activation function, a cost function, and a back-propagation algorithm. Deep

TABLE 2. List of common acronyms used

ADS	Anomaly Detection System
A-NIDS	Anomaly & Network Intrusion Detection System
AI	Artificial Intelligence
ANN	Artificial Neural Network
ART	Adaptive Resonance Theory
BSS	Blind Signal Separation
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies
CDBN	Convolutional Deep Belief Network
CNN	Convolutional Neural Network
CRN	Cognitive Radio Network
DBN	Deep Belief Network
DDoS	Distributed Denial of Service
DNN	Deep Neural Network
DNS	Domain Name Service
DPI	Deep Packet Inspection
EM	Expectation-Maximization
GTM	Generative Topographic Model
GPU	Graphics Processing Unit
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
ICA	Independent Component Analysis
IDS	Intrusion Detection System
IoT	Internet of Things
LSTM	Long Short-Term Memory
LLE	Locally Linear Embedding
LRD	Low Range Dependencies
ML	Machine Learning
MLP	Multi-Layer Perceptron
MDS	Multi-Dimensional Scaling
MCA	Minor Component Analysis
NMF	Non-Negative Matrix Factorization
NMS	Network Management System
NN	Neural Network
NMDS	Nonlinear Multi-dimensional Scaling
OSPF	Open Shortest Path First
PU	Primary User
PCA	Principal Component Analysis
PGM	Probabilistic Graph Model
QoE	Quality of Experience
QoS	Quality of Service
RBM	Restricted Boltzmann Machine
RNN	Recurrent Neural Network
SDN	Software Defined Network
SOM	Self-Organizing Map
SON	Self-Organizing Network
SVM	Support Vector Machine
SON	Self Organizing Network
SSAE	Shrinking Sparse Autoencoder
TCP	Transmission Control Protocol
t-SNE	t-Distributed Stochastic Neighbor Embedding
TL	Transfer Learning
VoIP	Voice over IP
VoQS	Variation of Quality Signature
VAE	Variational Autoencoder
WSN	Wireless Sensor Network

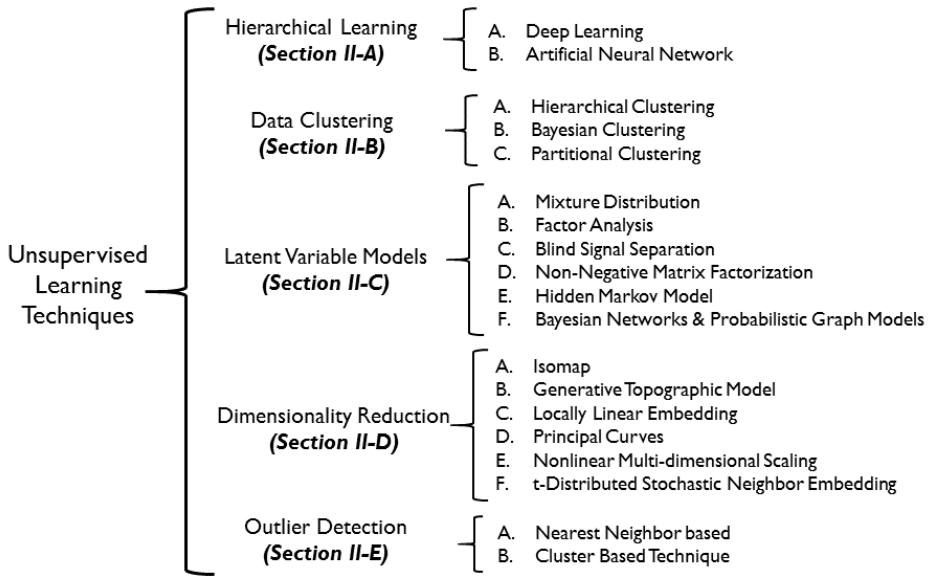

FIGURE 2. Taxonomy of unsupervised learning techniques

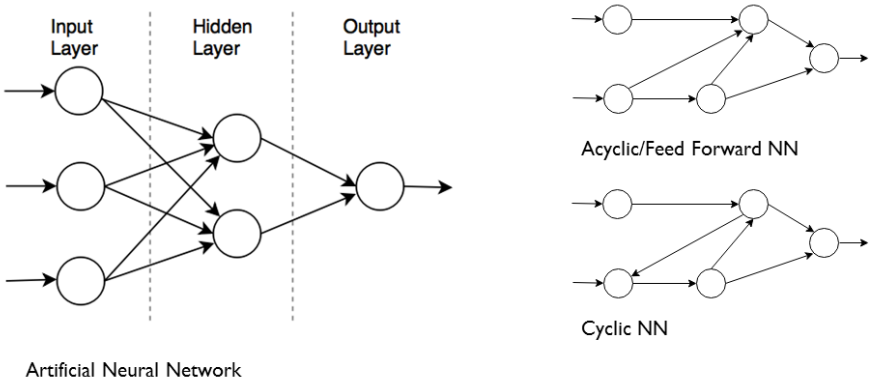

FIGURE 3. Illustration of an ANN (left); Different types of ANN topologies (right)

learning [40] is a hierarchical technique that models high-level abstraction in data using many layers of linear and nonlinear transformations. With deep enough stack of these transformation layers, a machine can self-learn a very complex model or representation of data. Learning takes place in hidden layers and the optimal weights and biases of the neurons are updated in two passes, namely, the forward pass and backward pass. A typical ANN and typical cyclic and acyclic topologies of interconnection between neurons are shown in Figure 3. A brief taxonomy of Unsupervised NNs is presented in Figure 4.

An ANN has three types of layers (namely input, hidden and output, each having different activation parameters). *Learning* is the process of assigning optimal activation parameters enabling ANN to perform input to output mapping. For a given problem, an ANN may require multiple hidden layers involving a long chain of computations, i.e., its *depth* [41]. Deep learning has revolutionized ML and is now increasingly being used in diverse settings—e.g., object identification in images, speech transcription into text, matching user's interests with items (such as news items, movies, products) and making recommendations, etc. But until 2006, relatively few people were interested in deep learning due to the high computational cost of deep learning procedures. It was widely believed that training deep learning architectures in an unsupervised manner was intractable, and supervised the training of deep NNs (DNN) also showed poor performance with large generalization errors [42]. However, recent advances [43]–[45] have shown that deep learning can be performed efficiently by separate unsupervised pre-training of each layer with the results revolutionizing the field of ML. Starting from the input (observation) layer, which acts as an input to the subsequent layers, pre-training tends to learn data distributions while the usual supervised stage performs a local search for fine-tuning.

TABLE 3. Applications of hierarchical learning/ deep learning in networking applications

Reference	Technique	Brief Summary
Internet Traffic Classification		
[28]	SAE & CNN	SAE and CNN are used for feature extraction from the Internet traffic data for classification and characterizing purpose.
[29]	CNN	CNN is used to extract features from the Internet traffic where traffic is considered as an image for malware detection.
[30]	Autoencoder	Autoencoder is used as a generative model to learn the latent feature representation of network traffic vector, for cyber attack detection and classification.
Anomaly/Intrusion Detection		
[31]	Denoising Autoencoder	Stochastically Improved autoencoder and denosing autoencoder are used to learn feature for zero day anomaly detection in Internet traffic.
[32]	RNN	Gated recurrent unit and random forest techniques are used for feature extraction and anomaly detection in IoT data.
[33]	RNN	RNN and DNN are employed to extract feature from raw data which then used for threat assessment and insider threat detection in data streams.
Network Operations, Optimization and Analytics		
[34]	Random Neural Network	Random neural network are used for extracting the quality behavior of multimedia application for improving the QoE of multimedia applications in wireless mesh network.
[35]	Random Neural Network	Random neural network are used for learning the mapping between QoE score and technical parameters so that it can give QoE score in real-time for multimedia applications in IEEE 802.11 wireless networks.
Emerging Networking Application of Unsupervised Learning		
[36]	DNN & CNN	Hierarchical learning is used for feature extraction from spectrogram snap shots of signal for modulation detection in communication system based on software defined radio.
[37]	CNN	Convolutional filters are used for feature extraction from cognitive radio waveforms for automatic recognition.
[38]	ANN	ANN is recommended to learn the hierarchy of the output, which is later used in SON.
[39]	RNN	RNN variant LSTM is used for learning memory based hierarchy of time interval based IoT sensor data, from smart cities datasets.

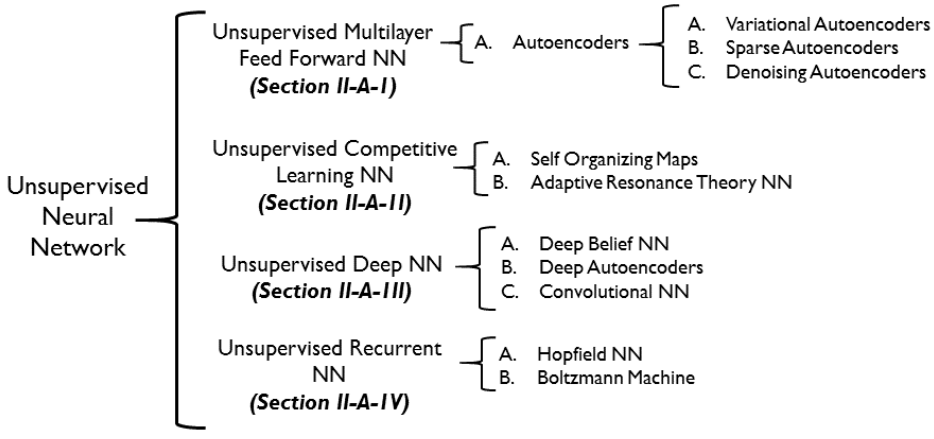

FIGURE 4. Taxonomy of unsupervised neural networks

1) Unsupervised Multilayer Feed Forward NN

Unsupervised multilayer feedforward NN, with reference to graph theory, has a directed graph topology as shown in Figure 3. It consists of no cycles, i.e., does not have a feedback path in input propagation through NN. Such kind of NN is often used to approximate a nonlinear mapping between inputs and required outputs. Autoencoders are the prime examples of unsupervised multilayer feedforward NNs.

a: Autoencoders

An autoencoder is an unsupervised learning algorithm for ANN used to learn compressed and encoded representation of data, mostly for dimensionality reduction and for unsupervised pre-training of feedforward NNs. Autoencoders are generally designed using approximation function and trained using backpropagation and stochastic gradient descent (SGD) techniques. Autoencoders are the first of their kind to use the back-propagation algorithm to train with unlabeled data. Autoencoders aim to learn a compact representation of the function of input using the same number of input and output units with usually less hidden units to encode a feature vector. They learn the input data function by recreating the input at the output, which is called encoding/decoding, to learn at the time of training NN. In short, a simple autoencoder learns a low-dimensional representation of the input data by exploiting similar recurring patterns.

Autoencoders have different variants [46] such as variational autoencoders, sparse autoencoders, and denoising autoencoders. *Variational autoencoder* is an unsupervised learning technique used clustering, dimensionality reduction, and visualization, and for learning complex distributions [47]. In a *sparse autoencoder*, a sparse penalty on the latent layer is applied for extracting a unique statistical feature from unlabeled data. Finally, *denoising autoencoders* are used to learn the mapping of a corrupted data point to its original location in the data space in an unsupervised manner for manifold learning and reconstruction distribution learning.

2) Unsupervised Competitive Learning NN

Unsupervised competitive learning NNs is a winner-take-all neuron scheme, where each neuron competes for the right of the response to a subset of the input data. This scheme is used to remove the redundancies from the unstructured data. Two major techniques of unsupervised competitive learning NNs are self-organizing maps and adaptive resonance theory NNs.

Self-Organizing/ Kohonen Maps: Self-Organizing Maps (SOM), also known as Kohonen's maps [48] [49], are a special class of NNs that uses the concept of *competitive learning*, in which output neurons compete amongst themselves to be activated in a real-valued output, results having only single neuron (or group of neurons), called *winning neuron*. This is achieved by creating lateral inhibition connections (negative feedback paths) between neurons [50]. In this orientation, the network determines the winning neuron within several iterations; subsequently, it is forced to reorganize itself based on the input data distribution (hence they are called Self-Organizing Maps). They were initially inspired by the human brain, which has specialized regions in which different sensory inputs are represented/processed by topologically ordered computational maps. In SOM, neurons are arranged on vertices of a lattice (commonly one or two dimensions). The network is forced to represent higher-dimensional data in lower-dimensional representation by preserving the topological properties of input data by using neighborhood function while transforming the input into a topological space in which neuron positions in the space are representatives of intrinsic statistical features that tell us about the inherently nonlinear nature of SOMs.

Training a network comprising SOM is essentially a three-stage process after random initialization of weighted connections. The three stages are as follow [51].

- *Competition:* Each neuron in the network computes its value using a discriminant function, which provides the basis of competition among the neurons. Neuron with the largest discriminant value in the competition group is declared the winner.
- *Cooperation:* The winner neuron then locates the center of the topological neighborhood of excited neurons in the previous stage, providing a basis for cooperation among excited neighboring neurons.
- *Adaption:* The excited neurons in the neighborhood increase/decrease their individual values of the discriminant function in regard to input data distribution through subtle adjustments such that the response of the winning neuron is enhanced for similar subsequent input. Adaption stage is distinguishable into two sub-stages: (1) the *ordering or self-organizing phase*, in which weight vectors are reordered according to topological space; and (2) the *convergence phase*, in which the map is fine-tuned and declared accurate to provide statistical quantification of the input space. This is the phase in which the map is declared to be converged and hence trained.

One essential requirement in training a SOM is the redundancy of the input data to learn about the underlying structure of neuron activation patterns. Moreover, sufficient quantity of data is required for creating distinguishable clusters; withstanding enough data for classification problem, there exist a problem of gray area between clusters and creation of infinitely small clusters where input data has minimal patterns.

Adaptive Resonance Theory: Adaptive Resonance Theory (ART) is another different category of NN models that is based on the theory of human cognitive information processing. It can be explained as an algorithm of incremental clustering which aims at forming multi-dimensional clusters, automatically discriminating and creating new categories based on input data. Primarily, ART models are classified as an unsupervised learning model; however, there exist ART variants that employ supervised and semi-supervised learning approaches as well. The main setback of most NN mod-

els is that they lose old information (updating/diminishing weights) as new information arrives, therefore an ideal model should be flexible enough to accommodate new information without losing the old one, and this is called the *plasticity-stability* problem. ART models provide a solution to this problem by self-organizing in real time and creating a competitive environment for neurons, automatically discriminating/creating new clusters among neurons to accommodate any new information.

ART model resonates around (top-down) observer expectations and (bottom-up) sensory information while keeping their difference within the threshold limits of vigilance parameter, which in result is considered as the member of the expected class of neurons [52]. Learning of an ART model primarily consists of a comparison field, recognition field, vigilance (threshold) parameter, and a reset module. The comparison field takes an input vector, which in result is passed, to best match in the recognition field; the best match is the current winning neuron. Each neuron in the recognition field passes a negative output in proportion to the quality of the match, which inhibits other outputs, therefore, exhibiting lateral inhibitions (competitions). Once the winning neuron is selected after a competition with the best match to the input vector, the reset module compares the quality of the match to the vigilance threshold. If the winning neuron is within the threshold, it is selected as the output, else the winning neuron is reset and the process is started again to find the next best match to the input vector. In case where no neuron is capable to pass the threshold test, a search procedure begins in which the reset module disables recognition neurons one at a time to find a correct match whose weight can be adjusted to accommodate the new match, therefore ART models are called self-organizing and can deal with the plasticity/stability dilemma.

3) Unsupervised Deep NN

In recent years unsupervised deep NN has become the most successful unsupervised structure due to its application in many benchmarking problems and applications [53]. Three major types of unsupervised deep NNs are deep belief NNs, deep autoencoders, and convolutional NNs.

Deep Belief NN: Deep Belief Neural Network or simply Deep Belief Networks (DBN) is a probability-based generative graph model that is composed of hierarchical layers of stochastic latent variables having binary valued activations, which are referred as hidden units or feature detectors. The top layers in DBNs have undirected, symmetric connections between them forming an associative memory. DBNs provide a breakthrough in unsupervised learning paradigm. In the learning stage, DBN learns to reconstruct its input, each layer acting as feature detectors. DBN can be trained by greedy layer-wise training starting from the top layer with raw input, subsequent layers are trained with the input data from the previously visible layer [43]. Once the network is trained in an unsupervised manner and learned the distribution of the data, it can be fine-tuned using supervised learning methods, or supervised layers can be concatenated in order to achieve the desired task (for instance, classification).

Deep Autoencoder: Another famous type of DBN is the *deep autoencoder*, which is composed of two symmetric DBNs—the first of which is used to encode the input vector, while the second decodes. By the end of the training of the deep autoencoder, it tends to reconstruct the input vector at the output neurons, and therefore the central layer between both DBNs is the actual compressed feature vector.

Convolutional NN: Convolutional NN (CNN) are feed forward NN in which neurons are adapted to respond to overlapping regions in two-dimensional input fields such as visual or audio input. It is commonly achieved by local sparse connections among successive layers and tied shared weights followed by rectifying and pooling layers which results in transformation invariant feature extraction. Another advantage of CNN over simple multilayer NN is that it is comparatively easier to train due to sparsely connected layers with the same number of hidden units. CNN represents the most significant type of architecture for computer vision as they solve two challenges with the conventional NNs: 1) scalable and computationally tractable algorithms are needed for processing high-dimensional images; and 2) algorithms should be transformation invariant since objects in an image can occur at an arbitrary position. However, most CNN's are composed of supervised feature detectors in the lower and middle hidden layers. In order to extract features in an unsupervised manner, a hybrid of CNN and DBN, called Convolutional Deep Belief Network (CDBN), is proposed in [54]. Making probabilistic max-pooling[1] to cover larger input area and convolution as an inference algorithm makes this model scalable with higher dimensional input. Learning is processed in an unsupervised manner as proposed in [44], i.e., greedy layer-wise (lower to higher) training with unlabeled data.

CDBN is a promising scalable generative model for learning translation invariant hierarchical representation from any high-dimensional unlabeled data in an unsupervised manner taking advantage of both worlds, i.e., DBN and CNN. CNN, being widely employed for computer vision applications, can be employed in computer networks for optimization of Quality of Experience (QoE) and Quality of Service (QoS) of multimedia content delivery over networks, which is an open research problem for next-generation computer networks [55].

4) Unsupervised Recurrent NN

Recurrent NN (RNN) is the most complex type of NN, and hence the nearest match to an actual human brain that processes sequential inputs. It can learn temporal behaviors of a given training data. RNN employs an internal memory per neuron to process such sequential inputs in order to

[1]Max-pooling is an algorithm of selecting the most responsive receptive field of a given interest region.

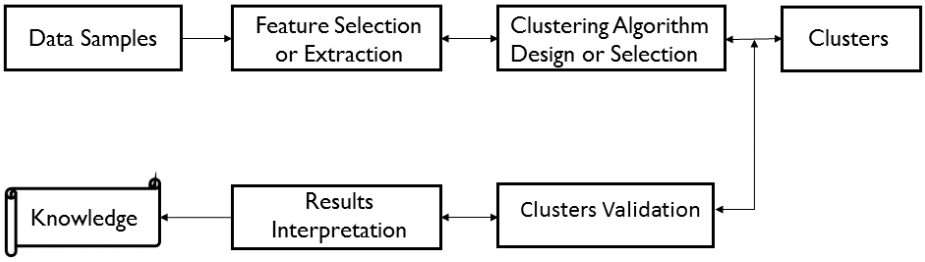

FIGURE 5. Clustering process

exhibit the effect of the previous event on the next. Compared to feed forward NNs, RNN is a stateful network. It may contain computational cycles among states and uses time as the parameter in the transition function from one unit to another. Being complex and recently developed, it is an open research problem to create domain-specific RNN models and train them with sequential data. Specifically, there are two perspectives of RNN to be discussed in the scope of this survey, namely, the depth of the architecture and the training of the network. The depth, in the case of a simple artificial NN, is the presence of hierarchical nonlinear intermediate layers between the input and output signals. In the case of an RNN, there are different hypotheses explaining the concept of depth. One hypothesis suggests that RNNs are inherently deep in nature when expanded with respect to sequential input; there are a series of nonlinear computations between the input at time $t(i)$ and the output at time $t(i+k)$.

However, at an individual discrete time step, certain transitions are neither deep nor nonlinear. There exist input-to-hidden, hidden-to-hidden, and hidden-to-output transitions, which are shallow in the sense that there are no intermediate nonlinear layers at discrete time step. In this regard, different deep architectures are proposed in [56] that introduce intermediate nonlinear transitional layers in between the input, hidden and output layers. Another novel approach is also proposed by stacking hidden units to create a hierarchical representation of hidden units, which mimic the deep nature of standard deep NNs.

Due to the inherently complex nature of RNN, to the best of our knowledge, there is no widely adopted approach for training RNNs and many novel methods (both supervised and unsupervised) are introduced to train RNNs. Considering unsupervised learning of RNN in the scope of this paper, [57] employ Long Short-term Memory (LSTM) RNN to be trained in an unsupervised manner using unsupervised learning algorithms, namely Binary Information Gain Optimization and non parametric Entropy Optimization, in order to make a network to discriminate between a set of temporal sequences and cluster them into groups. Results have shown remarkable ability of RNNs for learning temporal sequences and clustering them based on a variety of features. Two major types of unsupervised recurrent NN are Hopfield NN and Boltzmann machine.

Hopfield NN: Hopfield NN is a cyclic recurrent NN where each node is connected to others. Hopfield NN provides an abstraction of circular shift register memory with nonlinear activation functions to form a global energy function with guaranteed convergence to local minima. Hopfield NNs are used for finding clusters in the data without a supervisor.

Boltzmann Machine: The Boltzmann machine is a stochastic symmetric recurrent NN that is used for search and learning problems. Due to binary vector based simple learning algorithm of Boltzmann machine, very interesting features representing the complex unstructured data can be learned [58]. Since the Boltzmann machine uses multiple hidden layers as feature detectors, the learning algorithm becomes very slow. To avoid slow learning and to achieve faster feature detection instead of Boltzmann machine, a faster version, namely the restricted Boltzmann machine (RBM), is used for practical problems [59]. Restricted Boltzmann machine learns a probability distribution over its input data but since it is restricted in its layer to layer connectivity RBM loses its property of recurrence. It is faster than a Boltzmann machine because it only uses one hidden layer as a feature detector layer. RBM is used for dimensionality reduction, clustering and feature learning in computer networks.

5) Significant Applications of Hierarchical Learning in Networks

ANNs/DNNs are the most researched topic when creating intelligent systems in computer vision and natural language processing whereas their application in computer networks are very limited, they are employed in different networking applications such as classification of traffic, anomaly/intrusion detection, detecting Distributed Denial of Service (DDoS) attacks, and resource management in cognitive radios [60]. The motivation of using DNN for learning and predicting in networks is the unsupervised training that detects hidden patterns in ample amount of data that is near to impossible for a human to handcraft features catering for all scenarios. Moreover, many new research shows that a single model is not enough for the need of some applications, so developing a hybrid NN architecture having pros and cons of different models creates a new efficient NN which provides even better results. Such an approach is used in [61], in

which a hybrid model of ART and RNN is employed to learn and predict traffic volume in a computer network in real time. Real-time prediction is essential to adaptive flow control, which is achieved by using hybrid techniques so that ART can learn new input patterns without re-training the entire network and can predict accurately in the time series of RNN. Furthermore, DNNs are also being used in resource allocation and QoE/QoS optimizations. Using NN for optimization, efficient resource allocation without affecting the user experience can be crucial in the time when resources are scarce. Authors of [62], [63] propose a simple DBN for optimizing multimedia content delivery over wireless networks by keeping QoE optimal for end users. Table 3 also provides a tabulated description of hierarchical learning in networking applications. However, these are just a few notable examples of deep learning and neural networks in networks, refer to Section III for more applications and detailed discussion on deep learning and neural networks in computer networks.

B. DATA CLUSTERING

Clustering is an unsupervised learning task that aims to find hidden patterns in unlabeled input data in the form of clusters [64]. Simply put, it encompasses the arrangement of data in meaningful natural groupings on the basis of the similarity between different *features* (as illustrated in Figure 5) to learn about its structure. Clustering involves the organization of data in such a way that there are high intra-cluster and low inter-cluster similarity. The resulting structured data is termed as *data-concept* [65]. Clustering is used in numerous applications from the fields of ML, data mining, network analysis, pattern recognition, and computer vision. The various techniques used for data clustering are described in more detail later in Section II-B. In networking, clustering techniques are widely deployed for applications such as traffic analysis and anomaly detection in all kinds of networks (e.g., wireless sensor networks and mobile ad-hoc networks), with anomaly detection [66].

Clustering improves performance in various applications. McGregor et al. [67] propose an efficient packet tracing approach using the Expectation-Maximization (EM) probabilistic clustering algorithm, which groups flows (packets) into a small number of clusters, where the goal is to analyze network traffic using a set of representative clusters.

A brief overview of different types of clustering methods and their relationships can be seen in Figure 6. Clustering can be divided into three main types [68], namely *hierarchical clustering*, *Bayesian clustering*, and *partitional clustering*. Hierarchical clustering creates a hierarchical decomposition of data, whereas Bayesian clustering forms a probabilistic model of the data that decides the fate of a new test point probabilistically. In contrast, partitional clustering constructs multiple partitions and evaluates them on the basis of certain criterion or characteristic such as the Euclidean distance.

Before delving into the general sub-types of clustering, there are two unique clustering techniques, which need to be discussed, namely *density-based clustering* and *grid-based clustering*. In some cases, density-based clustering is classified as a partitional clustering technique; however, we have kept it separate considering its applications in networking. Density-based models target the most densely populated area of data space and separate it from areas having low densities, thus forming clusters [69]. [70] use density-based clustering to cluster data stream in real time, which is important in many applications (e.g., intrusion detection in networks). Another technique is grid-based clustering, which divides the data space into cells to form a grid-like structure; subsequently, all clustering actions are performed on this grid [71]. [71] also present a novel approach that uses a customized grid-based clustering algorithm to detect anomalies in networks. [72] proposed a novel method for clustering the time series data, this scheme was based on a distance measure between temporal features of the time series.

We move on next to describe three major types of data clustering approaches as per the taxonomy is shown in Figure 6.

1) Hierarchical Clustering

Hierarchical clustering is a well-known strategy in data mining and statistical analysis in which data is clustered into a hierarchy of clusters using an agglomerative (bottom-up) or a divisive (top-down) approach. Almost all hierarchical clustering algorithms are unsupervised and deterministic. The primary advantage of hierarchical clustering over unsupervised K-means and EM algorithms is that it does not require the number of clusters to be specified beforehand. However, this advantage comes at the cost of computational efficiency. Common hierarchical clustering algorithms have at least quadratic computational complexity compared to the linear complexity of K-means and EM algorithms. Hierarchical clustering methods have a pitfall: these methods fail to accurately classify messy high-dimensional data as its heuristic may fail due to the structural imperfections of empirical data. Furthermore, the computational complexity of the common agglomerative hierarchical algorithms is NP-hard. SOM, as discussed in Section II-A2, is a modern approach that can overcome the shortcomings of hierarchical models [73].

2) Bayesian Clustering

Bayesian clustering is a probabilistic clustering strategy where the posterior distribution of the data is learned on the basis of a prior probability distribution. Bayesian clustering is divided into two major categories, namely parametric and non-parametric [74]. The major difference between parametric and non-parametric techniques is the dimensionality of parameter space: if there are finite dimensions in the parameter space, the underlying technique is called Bayesian parametric; otherwise, the underlying technique is called Bayesian non-parametric. A major pitfall with the Bayesian clustering approach is that the choice of the wrong prior probability distributions can distort the projection of the

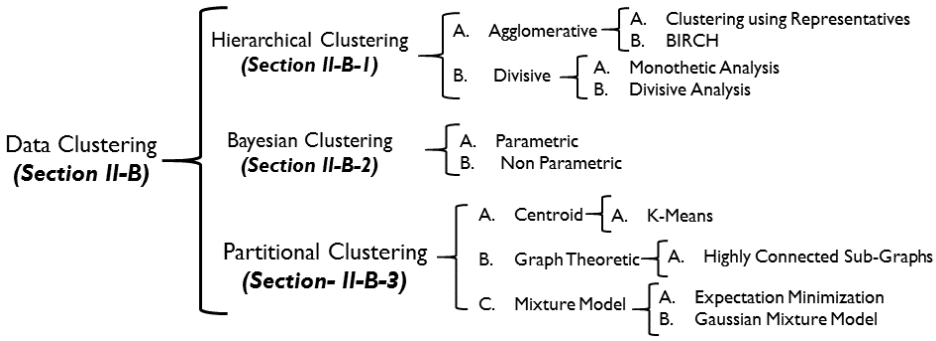

FIGURE 6. Clustering taxonomy

data. [75] performed Bayesian non-parametric clustering of network traffic data to determine the network application type.

3) Partitional Clustering

Partitional clustering corresponds to a special class of clustering algorithms that decomposes data into a set of disjoint clusters. Given n observations, the clustering algorithm partitions a data into $k < n$ clusters [76]. Partitional clustering is further classified into K-means clustering and mixture models.

a: K-Means Clustering

K-means clustering is a simple, yet widely used approach for classification. It takes a statistical vector as an input to deduce classification models or classifiers. K-means clustering tends to distribute m observations into n clusters where each observation belongs to the nearest cluster. The membership of observation to a cluster is determined using the cluster mean. K-means clustering is used in numerous applications in the domains of network analysis and traffic classification. [77] used K-means clustering in conjunction with supervised ID3 decision tree learning models to detect anomalies in a network. An ID3 decision tree is an iterative supervised decision tree algorithm based on the concept learning system. K-means clustering provided excellent results when used in traffic classification. [78] showed that K-means clustering performs well in traffic classification with an accuracy of 90%.

K-means clustering is also used in the domain of network security and intrusion detection. [79] proposed a K-means algorithm for intrusion detection. Experimental results on a subset of KDD-99 dataset shows that the detection rate stays above 96% while the false alarm rate stays below 4%. Results and analysis of experiments on K-means algorithm have demonstrated a better ability to search clusters globally.

Another variation of K-means is known as K-medoids, in which rather than taking the mean of the clusters, the most centrally located data point of a cluster is considered as the reference point of the corresponding cluster [80]. Few of the applications of K-medoids in the spectrum of anomaly detection can be seen here [80] [81].

b: Mixture Models

Mixture models are powerful probabilistic models for univariate and multivariate data. Mixture models are used to make statistical inferences and deductions about the properties of the sub-populations given only observations on the pooled population. They have also used to statistically model data in the domains of pattern recognition, computer vision, ML, etc. Finite mixtures, which are a basic type of mixture model, naturally model observations that are produced by a set of alternative random sources. Inferring and deducing different parameters from these sources based on their respective observations lead to clustering of the set of observations. This approach to clustering tackles drawbacks of heuristic-based clustering methods, and hence it is proven to be an efficient method for node classification in any large-scale network and has shown to yield effective results compared to techniques commonly used. For instance, K-means and hierarchical agglomerative methods rely on supervised design decisions, such as the number of clusters or validity of models [82]. Moreover, combining the EM algorithm with mixture models produces remarkable results in deciphering the structure and topology of the vertices connected through a multi-dimensional network [83]. [84] used Gaussian mixture model (GMM) to outperform signature based anomaly detection in network traffic data.

4) Significant Applications of Clustering in Networks

Clustering can be found in mostly all unsupervised learning problems, and there are diverse applications of clustering in the domain of computer networks. Two major networking applications where significant use of clustering can be seen are intrusion detection and Internet traffic classification. One novel way to detect anomaly is proposed in [85], this approach preprocesses the data using Genetic Algorithm (GA) combined with hierarchical clustering approach called Balanced Iterative Reducing using Clustering Hierarchies (BIRCH) to provide an efficient classifier based on Sup-

port Vector Machine (SVM). This hierarchical clustering approach stores abstracted data points instead of the whole dataset, thus giving more accurate and quick classification compared to all past methods, producing better results in detecting anomalies. Another approach [71] discusses the use of grid-based and density-based clustering for anomaly and intrusion detection using unsupervised learning. [86] used k-shape clustering scheme for analyzing spatiotemporal heterogeneity in mobile usage. Basically, a scalable parallel framework for clustering large datasets with high dimensions is proposed and then improved by inculcating frequency pattern trees. Table 4 also provides a tabulated description of data clustering applications in networks. These are just a few notable examples of clustering approaches in networks: refer to Section III for the detailed discussion on some salient clustering applications in the context of networks.

C. LATENT VARIABLE MODELS

A latent variable model is a statistical model that relates the manifest variables with a set of latent or hidden variables. Latent variable model allows us to express relatively complex distributions in terms of tractable joint distributions over an expanded variable space [95]. Underlying variables of a process are represented in higher dimensional space using a fixed transformation, and stochastic variations are known as latent variable models where the distribution in higher dimension is due to small number of hidden variables acting in a combination [96]. These models are used for data visualization, dimensionality reduction, optimization, distribution learning, blind signal separation and factor analysis. Next we will begin our discussion on various latent variable models, namely *mixture distribution*, *factor analysis*, *blind signal separation*, *non-negative matrix factorization*, *Bayesian networks & probabilistic graph models (PGM)*, *hidden Markov model (HMM)*, and *nonlinear dimensionality reduction techniques* (which further includes *generative topographic mapping*, *multi-dimensional scaling*, *principal curves*, *Isomap*, *locallly linear embedding*, and *t-distributed stochastic neighbor embedding*).

1) Mixture Distribution

Mixture distribution is an important latent variable model that is used for estimating the underlying density function. Mixture distribution provides a general framework for density estimation by using the simpler parametric distributions. Expectation maximization (EM) algorithm is used for estimating the mixture distribution model [97], through maximization of the log-likelihood of the mixture distribution model.

2) Factor Analysis

Another important type of latent variable model is factor analysis, which is a density estimation model. It has been used quite often in collaborative filtering and dimensionality reduction. It is different from other latent variable models in terms of the allowed variance for different dimensions as most latent variable models for dimensionality reduction in conventional settings use a fixed variance Gaussian noise model. In the factor analysis model, latent variables have diagonal covariance rather than isotropic covariance.

3) Blind Signal Separation

Blind Signal Separation (BSS), also referred to as Blind Source Separation, is the identification and separation of independent source signals from mixed input signals without or very little information about the mixing process. Figure 7 depicts the basic BSS process in which source signals are extracted from a mixture of signals. It is a fundamental and challenging problem in the domain of signal processing although the concept is extensively used in all types of multi-dimensional data processing. Most common techniques employed for BSS are principal component analysis (PCA) and independent component analysis (ICA).

a) Principal Component Analysis (PCA) is a statistical procedure that utilizes orthogonal transformation on the data to convert n number of possibly correlated variables into lesser k number of uncorrelated variables named principal components. Principal components are arranged in the descending order of their variability, first one catering for the most variable and the last one for the least. Being a primary technique for exploratory data analysis, PCA takes a cloud of data in n dimensions and rotates it such that maximum variability in the data is visible. Using this technique, it brings out the strong patterns in the dataset so that these patterns are more recognizable thereby making the data easier to explore and visualize.

PCA has primarily been used for dimensionality reduction in which input data of n dimensions is reduced to k dimensions without losing critical information in the data. The choice of the number of principal components is a question of the design decision. Much research has been conducted on selecting the number of components such as cross-validation approximations [98]. Optimally, k is chosen such that the ratio of the average squared projection error to the total variation in the data is less than or equal to 1% by which 99% of the variance is retained in the k principal components. But, depending on the application domain, different designs can increase/decrease the ratio while maximizing the required output. Commonly, many features of a dataset are often highly correlated; hence, PCA results in retaining 99% of the variance while significantly reducing the data dimensions.

b) Independent Component Analysis (ICA) is another technique for BSS that focuses on separating multivariate input data into additive components with the underlying assumption that the components are non-Gaussian and statistically independent. The most common example to understand ICA is the *cocktail party problem* in which there are n people talking simultaneously in a room and one tries to listen to a single voice. ICA actually separates source signals from input mixed signal by either minimizing the statistical de-

TABLE 4. Applications of data clustering in networking applications

Reference	Technique	Brief Summary
Internet Traffic Classification		
[87]	K-means & EM	A comparative analysis of Network traffic fault classification is performed between K-means and EM techniques.
[88]	K-means & Dissimilarity-based clustering	Semi supervised approach for Internet traffic classification benefits from K-means and dissimilarity-based clustering as a first step for the Internet traffic classification.
[89]	K-means	A novel variant of K-means clustering namely recursive time continuity constrained K-Means clustering, is proposed and used for real-time In-App activity analysis of encrypted traffic streams. Extracted feature vector of cluster centers are fed to random forest for further classification.
Anomaly/Intrusion Detection		
[90]	K-means & Hierarchical Clustering	K-means and hierarchical clustering is used to detect anomalies in call detail records of mobile wireless networks data.
[91]	GMM	GMM is used for detecting the anomalies that are affecting resources in cloud data centers.
[92]	K-means	K-means clustering is used for clustering the input data traffic for load balancing for network security.
Dimensionality Reduction and Visualization		
[93]	Fuzzy Feature Clustering	A new feature clustering based approach for dimensionality reduction of Internet traffic for intrusion detection is presented.
[94]	Fuzzy C-mean clustering & PCA	This works combines data clustering technique combined with PCA is used for dimensionality reduction and classification of the Internet traffic.

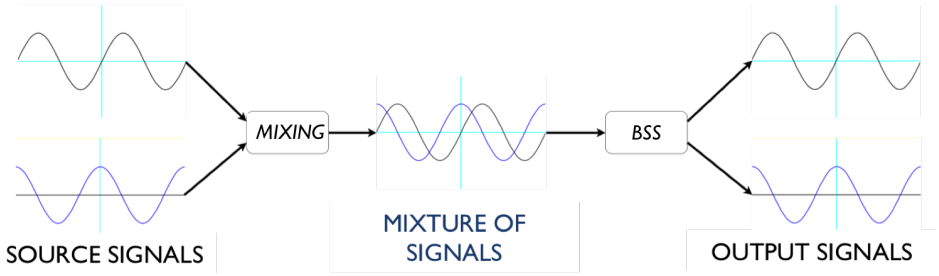

FIGURE 7. Blind signal separation (BSS): A mixed signal composed of various input signals mixed by some mixing process is blindly processed (i.e., with no or minimal information about the mixing process) to show the original signals.

pendence or maximizing the non-Gaussian property among the components in the input signals by keeping the underlying assumptions valid. Statistically, ICA can be seen as the extension of PCA, while PCA tries to maximize the second moment (variance) of data, hence relying heavily on Gaussian features; on the other hand, ICA exploits inherently non-Gaussian features of the data and tries to maximize the fourth moment of linear combination of inputs to extract non-normal source components in the data [99].

4) Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) is a technique to factorize a large matrix into two or more smaller matrices with no negative values, that is when multiplied, it reconstructs the approximate original matrix. NMF is a novel method in decomposing multivariate data making it easy and straightforward for exploratory analysis. By NMF, hidden patterns and intrinsic features within the data can be identified by decomposing them into smaller chunks, enhancing the interpretability of data for analysis, with positivity constraints. However, there exist many classes of algorithms [100] for NMF having different generalization properties, for example, two of them are analyzed in [101], one of which minimizes the least square error and while the other focuses on the Kullback-Leibler divergence keeping algorithm convergence intact.

5) Hidden Markov Model

Hidden Markov Models (HMM) are stochastic models of great utility, especially in domains where we wish to analyze temporal or dynamic processes such as speech recognition, primary users (PU) arrival pattern in cognitive radio networks (CRNs), etc. HMMs are highly relevant to CRNs since many environmental parameters in CRNs are not directly observable. An HMM-based approach can analytically model a Markovian stochastic process in which we do not have access to the actual states, which are assumed to be unobserved or hidden; instead, we can observe a state that is stochastically dependent on the hidden state. It is for this reason that an HMM is defined to be a doubly stochastic process.

6) Bayesian Networks & Probabilistic Graph Models (PGM)

In Bayesian learning we try to find the posterior probability distributions for all parameter settings, in this setup, we ensure that we have a posterior probability for every possible parameter setting. It is computationally expensive but we can use complicated models with a small dataset and still avoid overfitting. Posterior probabilities are calculated by dividing the product of sampling distribution and prior distribution by marginal likelihood; in simple words, posterior probabilities are calculated using Bayes theorem. The basis of reinforcement learning was also derived by using Bayes theorem [102]. Since Bayesian learning is computationally expensive a new research trend is approximate Bayesian learning [103]. Authors in [104] have given a comprehensive survey of different approximate Bayesian inference algorithms. With the emergence of Bayesian deep learning framework the deployment of Bayes learning based solution is increasing rapidly.

Probabilistic graph modeling is a concept associated with Bayesian learning. A model representing the probabilistic relationship between random variables through a graph is known as a probabilistic graph model (PGM). Nodes and edges in the graph represent a random variable and their probabilistic dependence, respectively. PGM are of two types: directed PGM and undirected PGM. Bayes networks also fall in the regime of directed PGM. PGM is used in many important areas such as computer vision, speech processing, and communication systems. Bayesian learning combined with PGM and latent variable models forms a probabilistic framework where deep learning is used as a substrate for making improved learning architecture for recommender systems, topic modeling, and control systems [105].

7) Significant Applications of Latent Variable Models in Networks

In [106], authors have applied latent structure on email corpus to find interpretable latent structure as well as evaluating its predictive accuracy on missing data task. A dynamic latent model for a social network is represented in [107]. Characterization of the end-to-end delay using a Weibull mixture model is discussed in [108]. Mixture models for end host traffic analysis have been explored in [109]. BSS is a set of statistical algorithms that are widely used in different application domains to perform different tasks such as dimensionality reduction, correlating and mapping features, etc. [110] employed PCA for Internet traffic classification in order to separate different types of flows in a network packet stream. Similarly, authors of [111] used a semi-supervised approach, where PCA is used for feature learning and an SVM classifier for intrusion detection in an autonomous network system. Another approach for detecting anomalies and intrusions proposed in [112] uses NMF to factorize different flow features and cluster them accordingly. Furthermore, ICA has been widely used in telecommunication networks to separate mixed and noisy source signals for efficient service. For example, [113] extends a variant of ICA called Efficient Fast ICA (EF-ICA) for detecting and estimating the symbol signals from the mixed CDMA signals received from the source endpoint.

In other literature, PCA uses a probabilistic approach to find the degree of confidence in detecting an anomaly in wireless networks [114]. Furthermore, PCA is also chosen as a method of clustering and designing Wireless Sensor Networks (WSNs) with multiple sink nodes [115]. However, these are just a few notable examples of BSS in networks, refer to Section III for more applications and detailed discussion on BSS techniques in the networking domain.

Bayesian learning has been applied for classifying Internet traffic, where Internet traffic is classified based on the posterior probability distributions. For early traffic identification in campus network real discretized conditional probability has been used to construct a Bayesian classifier [116]. Host-level intrusion detection using Bayesian networks is proposed in [117]. Authors in [118] purposed a Bayesian learning based feature vector selection for anomalies classification in BGP. Port scan attacks prevention scheme using a Bayesian learning approach is discussed in [119]. Internet threat detection estimation system is presented in [120]. A new approach towards outlier detection using Bayesian belief networks is described in [121]. Application of Bayesian networks in MIMO systems has been explored in [122]. Location estimation using Bayesian network in LAN is discussed in [123]. Similarly, Bayes theory and PGM are both used in Low-Density Parity Check (LDPC) and Turbo codes, which are the fundamental components of information coding theory. Table 5 also provides a tabulated description of latent variable models applications in networking.

D. DIMENSIONALITY REDUCTION

Representing data in fewer dimensions is another well-established task of unsupervised learning. Real world data often have high dimensions—in many datasets, these dimensions can run into thousands, even millions, of potentially correlated dimensions [133]. However, it is observed that the intrinsic dimensionality (governing parameters) of the data is less than the total number of dimensions. In order to find the essential pattern of the underlying data by extracting intrinsic dimensions, it is necessary that the real essence is not lost; e.g., it may be the case that a phenomenon is observable only in higher-dimensional data and is suppressed in lower dimensions, these phenomena are said to suffer from the curse of dimensionality [134]. While *dimensionality reduction* is sometimes used interchangeably with *feature selection* [135] [136], a subtle difference exists between the two [137]. Feature selection is traditionally performed as a supervised task with a domain expert helping in handcrafting a set of critical features of the data. Such an approach generally can perform well but is not scalable and prone to judgment bias. Dimensionality reduction, on the other hand, is more generally an unsupervised task, where instead of choosing a subset of features, it creates new features (dimensions) as a function of all features. Said differently, feature selection

TABLE 5. Applications of latent variable models in networking applications

Reference	Technique	Brief Summary
Internet Traffic Classification		
[124]	Mixture Distribution	An improved EM algorithm is proposed which derives a better GMM and used for the Internet traffic classification.
[125]	PCA	PCA based feature selection approach is used for the Internet traffic classification. Where PCA is employed for feature selection and irrelevant feature removal.
[126]	NMF	NMF based models are applied on the data streams to find the traffic patterns which frequently occurs in network for identification and classification of tidal traffic patterns in metro area mobile network traffic.
Anomaly/Intrusion Detection		
[127]	Bayesian Networks	Bayesian networks are employed for anomaly and intrusion detection such as DDoS attacks in cloud computing networks.
[128]	Hidden Semi-Markov Model	Hidden semi-Markov model is used to detect LTE signalling attack.
Network Operations, Optimization and Analytics		
[129]	Bayesian Networks	Scale-able Bayesian network models are used for data flow monitoring and analysis.
[130]	HMM	HMM and statistical analytic techniques combined with semantic analysis are used to propose a network management tool.
Dimensionality Reduction and Visualization		
[131]	PCA & Factor Analysis	PCA and factor analysis are used for dimensionality reduction and latent correlation identification in mobile traffic demand data.
[132]	PCA	PCA is used for dimensionality reduction and orthogonal coordinates of the social media profiles in ranking the social media profiles.

considers supervised data labels, while dimensionality reduction focuses on the data points and their distributions in an N-dimensional space.

There exist different techniques for reducing data dimensions [138] including projection of higher dimensional points onto lower dimensions, independent representation, and sparse representation, which should be capable of reconstructing the approximate data. Dimensionality reduction is useful for data modeling, compression, and visualization. By creating representative functional dimensions of the data and eliminating redundant ones, it becomes easier to visualize and form a learning model. Independent representation tries to disconnect the source of variation underlying the data distribution such that the dimensions of the representation are statistically independent [40]. Sparse representation technique represents the data vectors in linear combinations of small basis vectors.

It is worth noting here that many of the latent variable models (e.g., PCA, ICA, factor analysis) also function as techniques for dimensionality reduction. In addition to techniques such as PCA, ICA—which infer the latent inherent structure of the data through a linear projection of the data—a number of nonlinear dimensionality reduction techniques have also been developed and will be focused upon in this section to avoid repetition of linear dimensionality reduction techniques that have already been covered as part of the previous subsection. Linear dimensionality reduction techniques are useful in many settings but these methods may miss important nonlinear structure in the data due to their subspace assumption, which posits that the high-dimensional data points lie on a linear subspace (for example, on a 2-D or 3D plane). Such an assumption fails in high dimensions when data points are random but highly correlated with neighbors. In such environments nonlinear dimensionality reductions through *manifold learning* techniques—which can be construed as an attempt to generalize linear frameworks like PCA so that nonlinear structure in data can also be recognized—become desirable. Even though some supervised variants also exist, manifold learning is mostly performed in an unsupervised fashion using the nonlinear manifold substructure learned from the high-dimensional structure of the data from the data itself without the use of any predetermined classifier or labeled data. Some nonlinear dimensionality reduction (manifold learning) techniques are described below:

1) Isomap

Isomap is a nonlinear dimensionality reduction technique that finds the underlying low dimensional geometric information about a dataset. Algorithmic features of PCA and MDS are combined to learn the low dimensional nonlinear manifold structure in the data [139]. Isomap uses geodesic distance along the shortest path to calculate the low dimension representation shortest path, which can be computed using Dijkstra's algorithm.

2) Generative Topographic Model

Generative topographic mapping (GTM) represents the nonlinear latent variable mapping from continuous low dimen-

sional distributions embedded in high dimensional spaces [140]. Data space in GTM is represented as reference vectors and these vectors are a projection of latent points in data space. It is a probabilistic variant of SOM and works by calculating the Euclidean distance between data points. GTM optimizes the log-likelihood function, and the resulting probability defines the density in data space.

3) Locally Linear Embedding

Locally linear embedding (LLE) [133] is an unsupervised nonlinear dimensionality reduction algorithm. LLE represents data in lower dimensions yet preserving the higher dimensional embedding. LLE depicts data in a single global coordinate of lower dimensional mapping of input data. LLE is used to visualize multi-dimensional dimensional manifolds and feature extraction.

4) Principal Curves

The principal curve is a nonlinear dataset summarizing technique where non-parametric curves pass through the middle of multi-dimensional dataset providing the summary of the dataset [141]. These smooth curves minimize the average squared orthogonal distance between data points, this process also resembles the maximum likelihood for nonlinear regression in the presence of Gaussian noise [142].

5) Nonlinear Multi-dimensional Scaling

Nonlinear multi-dimensional scaling (NMDS) [143] is a nonlinear latent variable representation scheme. It works as an alternative scheme for factor analysis. In factor analysis, a multivariate normal distribution is assumed and similarities between different objects are expressed as a correlation matrix. Whereas NMDS does not impose such a condition, and it is designed to reach the optimal low dimensional configuration where similarities and dissimilarities among matrices can be observed. NMDS is also used in data visualization and mining tools for depicting the multi-dimensional data in 3 dimensions based on the similarities in the distance matrix.

6) t-Distributed Stochastic Neighbor Embedding

t-distributed stochastic neighbor embedding (t-SNE) is another nonlinear dimensionality reduction scheme. It is used to represent high dimensional data in 2 or 3 dimensions. t-SNE constructs a probability distribution in high dimensional space and constructs a similar distribution in lower dimensions and minimizes the KullbackâĂŞLeibler (KL) divergence between two distributions (which is a useful way to measure the difference between two probability distributions) [144].

Table 6 also provides a tabulated description of dimensionality reduction applications in networking. The applications of nonlinear dimensionality reduction methods are later described in detail in Section III-D.

E. OUTLIER DETECTION

Outlier detection is an important application of unsupervised learning. A sample point that is distant from other samples is called an outlier. An outlier may occur due to noise, measurement error, heavy tail distributions and a mixture of two distributions. There are two popular underlying techniques for unsupervised outlier detection upon which many algorithms are designed, namely the nearest neighbor based technique and clustering based method.

1) Nearest Neighbor Based Outlier Detection

The nearest neighbor method works on estimating the Euclidean distances or average distance of every sample from all other samples in the dataset. There are many algorithms based on nearest neighbor based techniques, with the most famous extension of the nearest neighbor being a k-nearest neighbor technique in which only k nearest neighbors participate in the outlier detection [154]. Local outlier factor is another outlier detection algorithm, which works as an extension of the k-nearest neighbor algorithm. Connectivity-based outlier factors [155], influenced outlierness [156], and local outlier probability models [157] are few famous examples of the nearest neighbor based techniques.

2) Cluster Based Outlier Detection

Clustering based methods use the conventional K-means clustering technique to find dense locations in the data and then perform density estimation on those clusters. After density estimation, a heuristic is used to classify the formed cluster according to the cluster size. Anomaly score is computed by calculating the distance between every point and its cluster head. Local density cluster based outlier factor [158], clustering based multivariate Gaussian outlier score [159] [160] and histogram based outlier score [161] are the famous cluster based outlier detection models in literature. SVM and PCA are also suggested for outlier detection in literature.

3) Significant Applications of Outlier Detection in Networks

Outlier detection algorithms are used in many different applications such as intrusion detection, fraud detection, data leakage prevention, surveillance, energy consumption anomalies, forensic analysis, critical state detection in designs, electrocardiogram and computed tomography scan for tumor detection. Unsupervised anomaly detection is performed by estimating the distances and densities of the provided non-annotated data [162]. More applications of outlier detection schemes will be discussed in Section III

F. LESSONS LEARNT

Key lessons drawn from the review of unsupervised learning techniques are summarized below.

1) Hierarchical learning techniques are the most popular schemes in literature for feature detection and extraction.

TABLE 6. Applications of dimensionality reduction in networking applications

Reference	Technique	Brief Summary
Internet Traffic Classification		
[145]	PCA & SVM	Internet traffic classification model is proposed based on PCA and SVM, where PCA is employed for dimensionality reduction and SVM for classification.
[146]	SOM & Probabilistic NN	Probabilistic neural network is used for dimensionality reduction and SOM for network traffic classification.
Anomaly/Intrusion Detection		
[147]	DBN	Dimensionality reduction of high dimensional feature set is performed by training a DBN as nonlinear dimensionality reduction tool for human activity recognition using smart phones.
[148]	Autoencoders	Latent representation learnt by using autoencoder is used for anomaly detection in network traffic, which is performed by using single Gaussian and full kernel density estimation.
[149]	PCA & SVM	A hybrid approach for intrusion detection is described, where PCA is used to perform dimensionality reduction operation on network data and SVM is used to detect intrusion in that low dimensional data.
Network Operations, Optimization and Analytics		
[150]	PCA	PCA is used for low dimensional feature extraction in a mobile network planning tool based on data analytic.
[151]	PCA & Simple Embedding	PCA combined with simple embedding from deep learning is used for dimensionality reduction which reduces the communication overhead between client and server.
Dimensionality Reduction and Visualization		
[152]	t-SNE & LSTM	LSTM is applied for modulation recognition in wireless data. t-SNE is used to perform dimensionality reduction and visualization of the wireless dataset's FFT response.
[153]	t-SNE & K-means	t-SNE is used for visualizing a high dimensional Wi-Fi mobility data in 3D.

2) Learning the joint distribution of a complex distribution over an expanded variable space is a difficult task. Latent variable models have been the recommended and well-established schemes in literature for this problem. These models are also used for dimensionality reduction and better representation of data.
3) Visualization of unlabeled multidimensional data is another unsupervised task. In this research, we have explored the dimensionality reduction as an underlying scheme for developing better multidimensional data visualization tools.

III. APPLICATIONS OF UNSUPERVISED LEARNING IN NETWORKING

In this section, we will introduce some significant applications of the unsupervised learning techniques that have been discussed in Section II in the context of computer networks. We highlight the broad spectrum of applications in networking and emphasize the importance of ML-based techniques, rather than classical hard-coded statistical methods, for achieving more efficiency, adaptability, and performance enhancement.

A. INTERNET TRAFFIC CLASSIFICATION

Internet traffic classification is of prime importance in networking as it provides a way to understand, develop and measure the Internet. Internet traffic classification is an important component for service providers to understand the characteristics of the service such as quality of service, quality of experience, user behavior, network security and many other key factors related to the overall structure of a network [163]. In this subsection, we will survey the unsupervised learning applications in network traffic classification.

As networks evolve at a rapid pace, malicious intruders are also evolving their strategies. Numerous novel hacking and intrusion techniques are being regularly introduced causing severe financial jolts to companies and headaches to their administrators. Tackling these unknown intrusions through accurate traffic classification on the network edge, therefore, becomes a critical challenge and an important component of the network security domain. Initially, when networks used to be small, simple port-based classification technique that tried to identify the associated application with the corresponding packet based on its port number was used. However, this approach is now obsolete because recent malicious software uses a dynamic port-negotiation mechanism to bypass firewalls and security applications. A number of contrasting Internet traffic classification techniques have been proposed since then, and some important ones are discussed next.

Most of the modern traffic classification methods use different ML and clustering techniques to produce accurate clusters of packets depending on their applications, thus producing efficient packet classification [10]. The main purpose of classifying network's traffic is to recognize the destination application of the corresponding packet and to control the flow of the traffic when needed such as prioritizing one flow over others. Another important aspect of traffic classification is to detect intrusions and malicious attacks or screen out forbidden applications (packets).

The first step in classifying Internet traffic is selecting accurate features, which is an extremely important, yet complex task. Accurate feature selection helps ML algorithms to avoid problems like class imbalance, low efficiency, and low classification rate. There are three major feature selection methods in Internet traffic for classification: the filter method, the wrapper based method, and the embedded method. These methods are based on different ML and genetic learning

algorithms [164]. Two major concerns in feature selection for Internet traffic classification are the large size of data and imbalanced traffic classes. To deal with these issues and to ensure accurate feature selection, a min-max ensemble feature selection scheme is proposed in [165]. A new information-theoretic approach for feature selection for skewed datasets is described in [166]. This algorithm has resolved the multi-class imbalance issue but it does not resolve the issues of feature selection. In 2017, an unsupervised autoencoder based scheme has outperformed previous feature learning schemes, autoencoders were used as a generative model and were trained in a way that the bottleneck layer learned a latent representation of the feature set; these features were then used for malware classification and anomaly detection to produce results that improved the state of the art in feature selection [30].

Much work has been done on classifying traffic based on supervised ML techniques. Initially, in 2004, the concept of clustering bi-directional flows of packets came out with the use of EM probabilistic clustering algorithm, which clusters the flows depending on various attributes such as packet size statistics, inter-arrival statistics, byte counts, and connection duration, etc. [67]. Furthermore, clustering is combined with the above model [172]; this strategy uses Naïve Bayes clustering to classify traffic in an automated fashion. Recently, unsupervised ML techniques have also been introduced in the domain of network security for classifying traffic. Major developments include a hybrid model to classify traffic in more unsupervised manner [173], which uses both labeled and unlabeled data to train the classifier making it more durable and efficient. However, later on, completely unsupervised methods for traffic classification have been proposed, and still, much work is going on in this area. Initially, a completely unsupervised approach for traffic classification was employed using the K-means clustering algorithm combined with log transformation to classify data into corresponding clusters. Then, [78] highlighted that using K-means and this method for traffic classification can improve accuracy by 10% to achieve an overall 90% accuracy.

Another improved and faster approach was proposed in 2006 [174], which examines the size of the first five packets and determines the application correctly using unsupervised learning techniques. This approach has shown to produce better results than the state-of-the-art traffic classifier, and also has removed its drawbacks (such as dealing with outliers or unknown packets, etc.). Another similar automated traffic classifier and application identifier can be seen in [175], and they use the auto-class unsupervised Bayesian classifier, which automatically learns the inherent natural classes in a dataset.

In 2013, another novel strategy for traffic classification known as *network traffic classification using correlation* was proposed [167], which uses non-parametric NN combined with statistical measurement of correlation within data to efficiently classify traffic. The presented approach addressed the three major drawbacks of supervised and unsupervised learning classification models: *firstly*, they are inappropriate for sparse complex networks as labeling of training data takes too much computation and time; *secondly*, many supervised schemes such as SVM are not robust to training data size; and *lastly*, and most importantly, all supervised and unsupervised algorithms perform poorly if there are few training samples. Thus, classifying the traffic using correlations appears to be more efficient and adapting. [176] compared four ANN approaches for computer network traffic, and modeled the Internet traffic like a time series and used mathematical methods to predict the time series. A greedy layer-wise training for unsupervised stacked autoencoder produced excellent classification results, but at the cost of significant system complexity. Genetic algorithm combined with constraint clustering process is used for Internet traffic data characterization [177]. In another work, a two-phased ML approach for Internet traffic classification using K-means and C5.0 decision tree is presented in [178] where the average accuracy of classification was 92.37%.

A new approach for Internet traffic classification has been introduced in 2017 by [88] in which unidirectional and bidirectional information is extracted from the collected traffic, and K-means clustering is performed on the basis of statistical properties of the extracted flows. A supervised classifier then classifies these clusters. Another unsupervised learning based algorithm for Internet traffic detection is described in [179] where a restricted Boltzmann machine based SVM is proposed for traffic detection, this paper model the detection as a classification problem. Results were compared with ANN and decision tree algorithms on the basis of precision and recall. Application of deep learning algorithms in Internet traffic classification has been discussed in [16], with this work also outlining the open research challenges in applying deep learning for Internet traffic classification. These problems are related to training the models for big data since Internet data for deep learning falls in big data regime, optimization issues of the designed models given the uncertainty in Internet traffic and scalability of deep learning architectures in Internet traffic classification. To cope with the challenges of developing a flexible high-performance platform that can capture data from a high-speed network operating at more than 60 Gbps, [180] have introduced a platform for high-speed packet to tuple sequence conversion which can significantly advance the state of the art in real-time network traffic classification. In another work, [181] used stacked autoencoders for Internet traffic classification and produced more than 90% accurate results for the two classes in KDD 99 dataset.

Deep belief network combined with Gaussian model employed for Internet traffic prediction in wireless mesh backbone network has been shown to outperform the previous maximum likelihood estimation technique for traffic prediction [182]. Given the uncertainty of WLAN channel traffic classification is very tricky, [169] proposed a new variant of Gaussian mixture model by incorporating universal background model and used it for the first time to classify the

TABLE 7. Internet traffic classification with respect to unsupervised learning techniques and tasks

Reference	Technique	Task	Brief Summary
[167]	Non Parametric NN	Hierarchical Representations/ Deep Learning	Applied statistical correlation with non parametric NN to produce efficient and adaptive results in traffic classification.
[67]	EM-based clustering	Data clustering	Applied EM probabilistic algorithm to cluster flows based on various attributes such as byte counts, inter-arrival statistics, etc. in flow classification.
[168]	EM-based clustering	Data clustering	Applied EM-based clustering approach to yield 9% better results compared to supervised Naïve Bayes based approach in traffic classification.
[78]	K-Means	Data clustering	Applied K-means clustering algorithm to produce an overall 90% accuracy in Internet traffic classification in a completely unsupervised manner.
[169]	GMM	Data Clustering	Applied GMM with universal background model for encrypted WLAN traffic classification.
[170]	GMM	Data Clustering	GMM and Kerner's traffic theory based ML model is used to evaluate real-time Internet traffic performance.
[171]	K-Means, DBSCAN	Data clustering	Applied cluster analysis to effectively identify similar traffic using transport layer statistics to overcome the problem of dynamic port allocation in port based classification.
[172]	Naïve Bayes clustering	Data clustering	Applied Naïve Bayes clustering algorithm in traffic classification.
[110]	PCA	Blind Signal Separation	Applied PCA and fast correlation based filter algorithm that yields more accurate and stable experimental results in Internet traffic flow classification.

WLAN traffic. A brief overview of the different Internet traffic classification systems, classified on the basis of unsupervised technique and tasks discussed earlier, is presented in the Table 7.

B. ANOMALY/INTRUSION DETECTION

The increasing use of networks in every domain has increased the risk of network intrusions, which makes user privacy and the security of critical data vulnerable to attacks. According to the annual computer crime and security survey 2005 [199], conducted by the combined teams of CSI (Computer Security Institute) and FBI (Federal Bureau of Investigation), total financial losses faced by companies due to the security attacks and network intrusions were estimated as US $130 million. Moreover, according to the Symantec Internet Security Threat Report [200], approximately 5000 new vulnerabilities were identified in the year 2015. In addition, more than 400 million new variants of malware programs and 9 major breaches were detected exposing 10 million identities. Therefore, insecurity in today's networking environment has given rise to the ever-evolving domain of network security and intrusion/anomaly detection [200].

In general, Intrusion Detection Systems (IDS) recognize or identify any act of security breach within a computer or a network; specifically, all requests which could compromise the confidentiality and availability of data or resources of a system or a particular network. Generally, intrusion detection systems can be categorized into three types: (1) signature-based intrusion detection systems; (2) anomaly detection systems; and (3) compound/hybrid detection systems, which include selective attributes of both preceding systems.

Signature detection, also known as misuse detection, is a technique that was initially used for tracing and identifying misuses of user's important data, computer resources, and intrusions in the network based on the previously collected or stored signatures of intrusion attempts. The most important benefit of a signature-based system is that a computer administrator can exactly identify the type of attack a computer is currently experiencing based on the sequence of the packets defined by stored signatures. However, it is nearly impossible to maintain the signature database of all evolving possible attacks, thus this pitfall of the signature-based technique has given rise to anomaly detection systems.

Anomaly Detection System (ADS) is a modern intrusion and anomaly detection system. Initially, it creates a baseline image of a system profile, its network and user program activity. Then, on the basis of this baseline image, ADS classifies any activity deviating from this behavior as an intrusion. Few benefits of this technique are: firstly, they are capable of detecting insider attacks such as using system resources through another user profile; secondly, each ADS is based on a customized user profile which makes it very difficult for attackers to ascertain which types of attacks would not set an alarm; and lastly, it detects unknown behavior in a computer system rather than detecting intrusions, thus it is capable of detecting any unknown sophisticated attack which is different from the users' usual behavior. However, these benefits come with a trade-off, in which the process of training a system on a user's 'normal' profile and maintaining those profiles is a time consuming and challenging task. If an inappropriate user profile is created, it can result in poor performance. Since ADS detects any behavior that does not align with a user's normal profile, its false alarm rate can be high. Lastly, another pitfall of ADS is that a malicious user can train ADS gradually to accept inappropriate traffic as normal.

As anomaly and intrusion detection have been a popular research area since the origin of networking and Internet, numerous supervised as well as unsupervised [201] learning techniques have been applied to efficiently detect intrusions and malicious activities. However, latest research focuses on the application of unsupervised learning techniques in this area due to the challenge and promise of using big data for

TABLE 8. Anomaly & network intrusion detection systems (A-NIDS) with respect to unsupervised learning techniques

Reference	Technique	Brief Summary
Hierarchical Representations/ Deep Learning		
[183]	Hierarchical NN	Applied radial basis function in a two layered hierarchical IDS to detect intruders in real time.
[184]	SOM	Advocated unsupervised NNs such as SOM to provide a powerful supplement to existing IDSs.
[185]	SOM	Overviewed the capabilities of SOM and its application in IDS.
[186]	SOM	Analyzed TCP data traffic patterns using SOM and detected anomalies based on abnormal behavior.
[187]	SOM	Applied SOM to host based intrusion detection.
[188]	SOM	Applied a hierarchical NN to detect intruders, emphasizing on the development of relational hierarchies and time representation.
[189]	SOM & ART	Applied SOM combined with ART networks in real-time IDS.
[190]	SOM & J.48 Decision Tree	Applied SOM combined with J.48 decision tree algorithm in IDS to detect anomaly and misuses intelligently.
[191]	Multi-Layer Perceptrons (MLP)	Presented a two-tier IDS architecture. PCA in the first tier reduces input dimensions, while MLP in the second tier detects and recognizes attacks with low detection time and high accuracy.
Data Clustering		
[71]	Density & Grid Based Clustering	Applied an unsupervised clustering strategy in density and grid based clustering algorithms to detect anomalies.
[85]	Fuzzy Rough Clustering	Applied the idea of Fuzzy set theory and fuzzy rough C-means clustering algorithms in IDS to detect abnormal behaviors in networks, producing excellent results.
[79]	K-Means	Applied K-means clustering in IDS to detect intrusions and anomalies.
[192]	K-Means with C4.5 Decision Trees	Applied K-means clustering combined with C4.5 decision tree models to detect intrusive and anomalous behavior in networks and systems.
[193]	Sub-space Clustering	Implemented a unique unsupervised outliers and anomaly detection approach using Sub-Space Clustering and Multiple Evidence Accumulation techniques to exactly identify different kinds of network intrusions and attacks such as DoS/DDoS, probing attacks, buffer overflows, etc.
[194]	Two-Tier Clustering	Applied a novel bi-layered clustering technique, in which the first layer constitutes of clustering of packets and the second layer is responsible for anomaly detection and time correlation, to detect intrusions.
[77]	K-Means & ID3 Decision Trees	Applied K-means clustering combined with ID3 decision tree models to detect intrusive and anomalous behavior in systems.
[195]	Centroid Based Clustering	Presented a survey on intrusion detection techniques based on centroid clustering as well as other popular unsupervised approaches.
[196]	Finite GMM	Applied an unsupervised greedy learning of finite GMM for anomaly detection in intrusion detection system.
Blind Signal Separation		
[111]	PCA	Applied PCA and SVM in IDS.
[197]	PCA	Applied a novel approach to translate each network connection into a data vector, and then applied PCA to reduce its dimensionality and detect anomalies.
[198]	PCA	Applied PCA and dimensionality reduction techniques in attack recognition and anomaly detection.
[112]	NMF	Applied NMF algorithms to capture intrusion and network anomalies.

optimizing networks.

Initial work focuses on the application of basic unsupervised clustering algorithms for detecting intrusions and anomalies. In 2005, an unsupervised approach was proposed based on density and grid-based clustering to accurately classify the high-dimensional dataset in a set of clusters; those points which do not fall in any cluster are marked as abnormal [71]. This approach has produced good results but the false positive rate was very high. In follow-up work, another improved approach that used fuzzy rough C-means clustering was introduced [85] [195]. K-means clustering is also another famous approach used for detecting anomalies which were later proposed in 2009 [79], which showed great accuracy and outperformed existing unsupervised methods. However, later in 2012, an improved method which used K-means clustering combined with the C4.5 decision tree algorithm was proposed [192] to produce more efficient results than prior approaches. [202] combines cluster centers and nearest neighbors for effective feature representation which ensures a better intrusion detection, however, a limitation with this approach is that it is not able to detect user to resource and remote to local attacks. Another scheme using unsupervised learning approach for anomaly detection is presented in [203]. The presented scheme combines subspace clustering and correlation analysis to detect anomalies and provide protection against unknown anomalies; this experiment used WIDE backbone networks data [204] spanning over six years and produced better results than previous K-means based techniques. Work presented in [205] shows that for different intrusions schemes, there are a small set of measurements required to differentiate between normal and anomalous traffic; the authors used two co-clustering schemes to perform clustering and to determine which measurement subset contributed the most towards accurate detection.

Another famous approach for increasing detection accuracy is ensemble learning, work presented in [206] employed

many hybrid incremental ML approaches with gradient boosting and ensemble learning to achieve better detection performance. Authors in [207] surveyed anomaly detection research from 2009 to 2014 and find out the unique algorithmic similarity for anomaly detection in Internet traffic: most of the algorithms studied have following similarities 1) Removal of redundant information in training phase to ensure better learning performance 2) Feature selection usually performed using unsupervised techniques and increases the accuracy of detection 3) Use ensembles classifiers or hybrid classifiers rather than baseline algorithms to get better results. Authors in [208] have developed an artificial immune system based intrusion detection system they have used density-based spatial clustering of applications with noise to develop an immune system against the network intrusion detection.

The application of unsupervised intrusion detection in cloud network is presented in [209] where authors have proposed a fuzzy clustering ANN to detect the less frequent attacks and improve the detection stability in cloud networks. Another application of unsupervised intrusion detection system for clouds is surveyed in [210], where fuzzy logic based intrusion detection system using supervised and unsupervised ANN is proposed for intrusion detection; this approach is used for DOS and DDoS attacks where the scale of the attack is very large. Network intrusion anomaly detection system (NIDS) based on K-means clustering are surveyed in [211]; this survey is unique as it provides distance and similarity measure of the intrusion detection and this perspective has not been studied before 2015. Unsupervised learning based applications of anomaly detection schemes for wireless personal area networks, wireless sensor networks, cyber-physical systems, and WLANs are surveyed in [212].

Another paper [213] reviewing anomaly detection has presented the application of unsupervised SVM and clustering based applications in network intrusion detection systems. Unsupervised discretization algorithm is used in Bayesian network classifier for intrusion detection, which is based on Bayesian model averaging [214]; the authors show that the proposed algorithm performs better than the Naïve Bayes classifier in terms of accuracy on the NSL-KDD intrusion detection dataset. Border gateway protocol (BGP)—the core Internet inter-autonomous systems (inter-AS) routing protocol—is also error prone to intrusions and anomalies. To detect these BGP anomalies, many supervised and unsupervised ML solutions (such as hidden Markov models and principal component analysis) have been proposed in literature [215]. Another problem for anomaly detection is low volume attacks, which have become a big challenge for network traffic anomaly detection. While long-range dependencies (LRD) are used to identify these low volume attacks, LRD usually works on aggregated traffic volume; but since the volume of traffic is low, the attacks can pass undetected. To accurately identify low volume abnormalities, [216] proposed the examination of LRD behavior of control plane and data plane separately to identify low volume attacks.

Other than clustering, another widely used unsupervised technique for detecting malicious and abnormal behavior in networks is SOMs. The specialty of SOMs is that they can automatically organize a variety of inputs and deduce patterns among themselves, and subsequently determine whether the new input fits in the deduced pattern or not, thus detecting abnormal inputs [184] [185]. SOMs have also been used in host-based intrusion detection systems in which intruders and abusers are identified at a host system through incoming data traffic [188], later on, a more robust and efficient technique was proposed to analyze data patterns in TCP traffic [186]. Furthermore, complex NNs have also been applied to solve the same problem and remarkable results have been produced. A few examples include the application of ART combined with SOM [189]. The use of PCA can also be seen in detecting intrusions [197]. NMF has also been used for detecting intruders and abusers [112], and lastly dimensionality reduction techniques have also been applied to eradicate intrusions and anomalies in the system [198]. For more applications, refer to Table 8, which classifies different network anomaly and intrusion detection systems on the basis of unsupervised learning techniques discussed earlier.

C. NETWORK OPERATIONS, OPTIMIZATIONS, AND ANALYTICS

Network management comprises of all the operations included in initializing, monitoring and managing of a computer network based on its network functions, which are the primary requirements of the network operations. The general purpose of network management and monitoring systems is to ensure that basic network functions are fulfilled, and if there is any malfunctioning in the network, it should be reported and addressed accordingly. Following is a summary of different network optimization tasks achieved through unsupervised learning models.

1) QoS/ QoE Optimization

QoS and QoE are measures of service performance and end-user experience, respectively. QoS mainly deals with the performance as seen by the user being measured quantitatively, while QoE is a qualitative measure of subjective metrics experienced by the user. QoS/QoE for Internet services (especially multimedia content delivery services) is crucial in order to maximize the user experience. With the dynamic and bursty nature of Internet traffic, computer networks should be able to adapt to these changes without compromising end-user experiences. As QoE is quite subjective, it heavily relies on the underlying QoS which is affected by different network parameters. [232] and [233] suggested different measurable factors to determine the overall approximation of QoS such as error rates, bit rate, throughput, transmission delay, availability, jitters, etc. Furthermore, these factors are used to correlate QoS with QoE in the perspective of video streaming where QoE is essential to end-users.

The dynamic nature of the Internet dictates network design for different applications to maximize QoS/QoE since there

TABLE 9. Unsupervised learning techniques employed for network operations, optimizations and analytics

Reference	Technique	Brief Summary	Network Type
Hierarchical Representations/ Deep Learning			
[217]	ART fuzzy	Applied ART NNs at clusterheads and sensor nodes to extract regular patterns, reducing data for lesser communication overhead.	WSN
[218]	ART	Applied ART at each network node for data aggregation.	WSN
[219]	DNN	Applied different DNN layers corresponding to WSN layers in order to compress data.	WSN
[220]	RNN	Applied RNN to achieve optimal QoS in cognitive packet networks.	Cognitive networks
[221]	SOM	Applied SOM to cluster nodes into categories based on node location, energy and concentration; some nodes becomes clusterheads.	WSN
[222]	SOM	Applied SOM to categorize and select nodes with higher energy levels to become clusterheads based on node energy levels.	WSN
[223]	SOM	Applied SOM followed by K-means to cluster and select clusterheads in WSNs.	WSN
[224]	SOM	Applied SOMs in clusterheads to find patterns in data.	WSN
[225]	DNN	Applied a competitive neural algorithm for condition monitoring and fault detection in 3G cellular networks.	Cellular networks
[226]	RNN	Applied RNN for fault detection. RNN, which is deployed in each sensor node, takes inputs from neighboring nodes, and generates outputs for comparison with the generated data; if the difference exceeds a certain threshold, the node is regarded as anomalous.	WSN
Data Clustering			
[227]	Fuzzy C-Means Clustering	Applied fuzzy C-means clustering technique to select nodes with the highest residual energy to gather data and send information using an energy-efficient routing in WSNs.	WSN
[228]	K-Means Clustering	Applied K-means clustering to design multiple sink nodes in WSNs.	WSN
[229]	K-Means Partitioning	Applied K-means clustering to identify compromised nodes and applied Kullback-Leibler (KL) distance to determine the trustworthiness (reputation) of each node in a trust-based WSN.	WSN
Blind Signal Separation			
[230]	PCA	Applied PCA to resolve the problem of cooperative spectrum sensing in cognitive radio networks.	Cognitive radio networks
[231]	ICA	Applied ICA based CDMA receivers to separate and identify mixed source signals.	CDMA
[114]	PCA	Applied PCA to evaluate the degree of confidence in detection probability provided by a WSN. The probabilistic approach is a deviation from the idealistic assumption of sensing coverage used in a binary detection model.	WSN
[115]	PCA	Applied PCA for hierarchical anomaly detection in a distributed WSN.	WSN

is no predefined adaptive algorithm that can be used to fulfill all the necessary requirements for prospective application. Due to this fact, ML approaches are employed in order to adapt to the real-time network conditions and take measures to stabilize/maximize the user experience. [234] employed a hybrid architecture having unsupervised feature learning with a supervised classification for QoE-based video admission control and resource management. Unsupervised feature learning in this system is carried out by using a fully connected NN comprising RBMs, which capture descriptive features of video that are later classified by using a supervised classifier. Similarly, [235] presents an algorithm to estimate the Mean Opinion Score, a metric for measuring QoE, for VoIP services by using SOM to map quality metrics to features.

Moreover, research has shown that QoE-driven content optimization leads to the optimal utilization of the network. [236] showed that 43% of the bit overhead on average can be reduced per image delivered on the web. This is achieved by using the quality metric VoQS (Variation of Quality Signature), which can arbitrarily compare two images in terms of web delivery performance. By applying this metric for unsupervised clustering of the large image dataset, multiple coherent groups are formed in device-targeted and content-dependent manner. In another study [237], deep learning is used to assess the QoE of 3D images that have yet to show good results compared with the other deterministic algorithms. The outcome is a Reduced Reference QoE assessment process for automatic image assessment, and it has a significant potential to be extended to work on 3D video assessment.

In [238], a unique technique of the model-based RL approach is applied to improve bandwidth availability, and hence throughput performance, of a network. The MRL

model is embedded in a node that creates a model of the operating environment and uses it to generate virtual states and rewards for the virtual actions taken. As the agent does not need to wait for the real states and rewards from the operating environment, it can explore various kinds of actions on the virtual operating environment within a short period of time which helps to expedite the learning process, and hence the convergence rate to the optimal action. In [239], a MARL approach is applied in which nodes exchange Q-values among themselves and select their respective next-hop nodes with the best possible channel conditions while forwarding packets towards the destination. This helps to improve throughput performance as nodes in a network ensure that packets are successfully sent to the destination in a collaborative manner.

2) TCP Optimization

Transmission Control Protocol (TCP) is the core end-to-end protocol in TCP/IP stack that provides reliable, ordered and error-free delivery of messages between two communicating hosts. Due to the fact that TCP provides reliable and in-order delivery, congestion control is one of the major concerns of this protocol, which is commonly dealt with the algorithms defined in *RFC 5681*. However, classical congestion control algorithms are sub-optimal in hybrid wired/wireless networks as they react to packet loss in the same manner in all network situations. In order to overcome this shortcoming of classical TCP congestion control algorithms, an ML-based approach is proposed in [240], which employs a supervised classifier based on features learned for classifying a packet loss due to congestion or link errors. Other approaches to this problem currently employed in literature include using RL that uses fuzzy logic based reward evaluator based on game theory [241]. Another promising approach, named *Remy* [242], uses a modified model of *Markov decision process* based on three factors: 1) prior knowledge about the network; 2) a traffic model based on user needs (i.e., throughput and delay); and 3) an objective function that is to be maximized. By this learning approach, a customized best-suited congestion control scheme is produced specifically for that part of the network, adapted to its unique requirements. However, classifying packet losses using unsupervised learning methods is still an open research problem and there is a need for real-time adaptive congestion control mechanism for multi-modal hybrid networks.

For more applications, refer to Table 9, which classifies different various network optimization and operation works on the basis of their network type and the unsupervised learning technique used.

D. DIMENSIONALITY REDUCTION & VISUALIZATION

Network data usually consists of multiple dimensions. To apply machine learning techniques effectively the number of variables is needed to be reduced. Dimensionality reduction schemes have a number of significant potential applications in networks. In particular, dimensionality reduction can be used to facilitate network operations (e.g., for anomaly/intrusion detection, reliability analysis, or for fault prediction) and network management (e.g., through visualization of high-dimensional networking data). A tabulated summary of various research works using dimensionality reduction techniques for various kinds of networking applications is provided in Table 10.

Dimensionality reduction techniques have been used to improve the effectiveness of the anomaly/intrusion detection system. [255] proposed a DDoS detection system in SDN where dimensionality reduction is used for feature extraction and reduction in an unsupervised manner using stacked sparse autoencoders. [256] proposed a flow-based anomaly intrusion detection using replicator neural network. Proposed network is based on an encoder and decoder where the hidden layer between encoder and decoder performs the dimensionality reduction in an unsupervised manner, this process also corresponds to PCA. Similarly, [257] have proposed another anomaly detection procedure where dimensionality reduction for feature extraction is performed using multi-scale PCA and then using wavelet analysis, so that the anomalous traffic is separated from the flow. Dimensionality reduction using robust PCA based on minimum covariance determinant estimator for anomaly detection is presented in [258]. [259] applied PCA for dimensionality reduction in network intrusion detection application. To improve the performance of intrusion detection scheme, another algorithm based on dimensionality reduction for new feature learning using PCA is presented in [260] [261]. [262] have reviewed the dimensionality reduction schemes for intrusion detection in multimedia traffic and proposed an unsupervised feature selection scheme based on the dimensionality-reduced multimedia data.

Dimensionality reduction using autoencoders performs a vital role in fault prediction and reliability analysis of the cellular networks, this work also recommends deep belief networks and autoencoders as logical fault prediction techniques for self-organizing networks [263]. Most of the Internet applications use encrypted traffic for communication, previously deep packet inspection (DPI) was considered a standard way of classifying network traffic but with the varying nature of the network application and randomization of port numbers and payload size DPI has lost its significance. Authors in [264] have proposed a hybrid scheme for network traffic classification. The proposed scheme uses extreme machine learning, genetic algorithms and dimensionality reduction for feature selection and traffic classification. [265] applied fuzzy set theoretic approach for dimensionality reduction along with fuzzy C-mean clustering algorithm for the quality of web usage. In another work, [266] used Shrinking Sparse AutoEncoders (SSAE) for representing high-dimensional data and utilized SSAE in compressive sensing settings.

Visualization of high dimensional data in lower dimension representation is another application of dimensionality reduc-

TABLE 10. Dimensionality reduction techniques employed for networking applications

Reference	Technique	Brief Summary	Network Type
[243]	Autoencoders	Applied autoencoders to design an end-to-end communication system that can jointly learn transmitter and receiver implementations as well as signal encodings in unsupervised manner.	MIMO
[244]	Autoencoders	New approach for designing and optimizing the physical layer is explored using autoencoders for dimensionality reduction.	MIMO
[245]	Convolutional Autoencoders	Applied autoencoders for representation learning of structured radio communication signals.	Software Radio/ Cognitive Radio
[246]	Multi-dimensional Scaling	Applied distance based subspace dimensionality reduction technique for anomaly detection in data traffic.	Internet Traffic
[247]	Multi-dimensional Scaling	Used MDS to preprocess a statistical dataset for cell outage detection in SON.	SON
[248]	Sparse Gaussian Method	Applied sparse Gaussian method for linear dimensionality reduction over noisy channels in wireless sensor networks.	WSN
[249]	PCA	Applied linear and nonlinear dimensionality reduction techniques along with support vector machine for cognitive radio.	Cognitive Radio
[250]	PCA	Applied L1 norm PCA for dimensionality reduction in network intrusion detection system.	Internet Traffic
[251]	PCA	Applied PCA for diemensionality reduction in anomaly detection for cyber security applications.	SMS
[252]	Manifold Learning	Proposed a manifold learning based visualization tool for network traffic visualization and anomaly detection.	Internet Traffic
[253]	Transfer Learning and t-SNE	Used transfer learning for multimedia web mining and t-SNE for dimensionality reduction and visualization of web mining resultant model.	Multimedia Web
[254]	Clustering and t-SNE	Proposed an early threat detection scheme using darknet data, where clustering is used for threat detection and dimensionality reduction for visualization is performed by using t-SNE.	Internet Traffic

tion. There are many relevant techniques such as PCA and t-SNE that can be used to extract the underlying structure of high-dimensional data, which can then be visualized to aid human insight seeking and decision making [144]. A number of researchers have proposed to utilize dimensionality reduction techniques to aid visualization of networking data. [252] proposed a manifold learning based visualization tool for network traffic visualization and anomaly detection. [267] proposed a PCA-based solution for the detection and visualization of networking attacks, in which PCA is used for the dimensionality reduction of the feature vector extracted from KDD network traffic dataset. [268] used t-SNE for depicting malware fingerprints in their proposed network intrusion detection system. [269] proposed a rectangular dualization scheme for visualizing the underlying network topology. [270] used dimensionality reduction and t-SNE of clustering and visualization of botnet traffic. Finally, a lightweight platform for home Internet monitoring is presented in [271] where PCA and t-SNE are used for dimensionality reduction and visualization of the network traffic. A number of tools are readily available—e.g., Divvy [272], Weka [273]—that implement dimensionality reduction and other unsupervised ML techniques (such as PCA and manifold learning) and allow exploratory data analysis and visualization of high-dimensional data.

Dimensionality reduction techniques and tools have been utilized in all kinds of networks and we present some recent examples related to self-organizing networks (SONs) and software-defined radios (SDRs). [274] proposed a semi-supervised learning scheme for anomaly detection in SON based on dimensionality reduction and fuzzy classification technique. [275] used minor component analysis (MCA) for dimensionality reduction as a preprocessing step for user-level statistical data in LTE-A networks to detect the cell outage. [247] used multi-dimensional scaling (MDS), a dimensionality reduction scheme, as part of the preprocessing step for cell outage detection in SON. Another data-driven approach by [276] also uses MDS for getting a low dimensional embedding of target key point indicator vector as a preprocessing step to automatically detect cell outage in SON. [277] used PCA for dimensionality reduction of drive test samples to detect cell outages autonomously in SON. Conventional routing schemes are not sufficient for the fifth generation of communication systems. [278] proposed a supervised deep learning based routing scheme for heterogeneous network traffic control. Although supervised approach performed well, gathering a lot of heterogeneous traffic with labels, and then processing them with a plain ANN is computationally extensive and prone to errors due to the imbalanced nature of the input data and the potential for overfitting. In 2017, [279] has presented a deep learning based approach for routing and cost-effective packet processing. The proposed model uses deep belief architecture and benefits from the dimensionality reduction property of the restricted Boltzmann machine. The proposed work also provides a novel Graphics Processing Unit (GPU) based router architecture. The detailed analysis shows that deep learning based SDR and routing technique can meet the changing network requirements and massive network traffic growth. The routing scheme proposed in [279] outperforms conventional

open shortest path first (OSPF) routing technique in terms of throughput and average delay per hop.

E. EMERGING NETWORKING APPLICATIONS OF UNSUPERVISED LEARNING

Next generation network architectures such as Software-defined Networks (SDN), Self Organizing Networks (SON), and the Internet of Things (IoT) are expected to be the basis of future intelligent, adaptive, and dynamic networks [280]. ML techniques will be at the center of this revolution providing the aforementioned properties. This subsection covers the recent applications of unsupervised ML techniques in SDNs, SONs, and IoTs.

1) Software Defined Networks

SDN is a disruptive new networking architecture that simplifies network operating and managing tasks and provides infrastructural support for novel innovations by making the network programmable [281]. In simple terms, the idea of programmable networks is to simply decouple the data forwarding plane and control/decision plane, which is rather tightly coupled in the current infrastructure. The use of SDN can also be seen in managing and optimizing networks as network operators go through a lot of hassle to implement high-level security policies in term of distributed low-level system configurations, thus SDN resolves this issue by decoupling the planes and giving network operators better control and visibility over network, enabling them to make frequent changes to network state and providing support for high-level specification language for network control [282]. SDN is applicable in a wide variety of areas ranging from enterprise networks, data centers, infrastructure based wireless access networks, optical networks to home and small businesses, each providing many future research opportunities [281].

Unsupervised ML techniques are seeing a surging interest in SDN community as can be seen by a spate of recent work. A popular application of unsupervised ML techniques in SDNs relates to the application of *intrusion detection and mitigation of security attacks* [283]. Another approach for detecting anomalies in a cloud environment using unsupervised learning model has been proposed by [284] that uses SOM to capture emergent system behavior and predict unknown and novel anomalies without any prior training or configuration. A DDoS detection system for SDN is presented in [255] where stacked autoencoders are used to detect DDoS attacks. A density peak based clustering algorithm for DDoS attack is proposed as a new method to review the potentials of using SDN to develop an efficient anomaly detection method [285]. [286] have recently presented an intelligent threat aware response system for SDN using reinforcement learning, this work also recommends using unsupervised feature learning to improve the threat detection process. Another framework for anomaly detection, classification, and mitigation for SDN is presented in [287] where unsupervised learning is used for *traffic feature analysis*. [288] have presented a forensic framework for SDN and recommended K-means clustering for anomaly detection in SDN. Another work [289] discusses the potential opportunities for using unsupervised learning for *traffic classification* in SDN. Moreover, deep learning and distributed processing can also be applied to such models in order to better adapt to evolving networks and contribute to the future of SDN infrastructure as a service.

2) Self Organizing Networks

SON is another new and popular research regime in networking, SON is inspired by the biological system which works in the self-organization and achieves the task by learning from the surrounding environment. As the connected network devices are growing exponentially, and the communication cell size has reduced to femtocells, the property of self-organization is becoming increasingly desirable [290]. Feasibility of SON application in the fifth generation (5G) of wireless communication is studied in [291] and the study shows that without (supervised as well as unsupervised) ML support, SON is not possible. Application of ML techniques in SON has become a very important research area as it involves learning from the surroundings for intelligent decision-making and reliable communication [2].

Application of different ML-based SON for heterogeneous networks is considered in [292], this paper also describes the unsupervised ANN and hidden Markov models techniques employed for better learning from the surroundings and adapting accordingly. PCA and clustering are the two most used unsupervised learning schemes utilized for parameter optimization and feature learning in SON [290]. These ML schemes are used in self-configuration, self-healing, and self-optimization schemes. Game theory is another unsupervised learning approach used for designing self-optimization and greedy self-configuration design of SON systems [293]. Authors in [294] proposed an unsupervised ANN for link quality estimation of SON which outperformed simple moving average and exponentially weighted moving averages.

3) Internet of Things

IoT is an emerging paradigm with a growing academic and industry interest. IoT is an abstraction of intelligent, physical and virtual devices with unique identities, connected together to form a cyber-physical framework. These devices collect, analyze and transmit data to public or private cloud for intelligent [295]. IoT is a new networking paradigm and it is expected to be deployed in health care, smart cities, home automation, agriculture, and industry. With such a vast plane of applications, IoT needs ML to collect and analyze data to make intelligent decisions. The key challenge that IoT must deal with is the extremely large scale (billions of devices) of future IoT deployments [296]. Designing, analyzing and predicting are the three major tasks and all involve ML, a few examples of unsupervised ML are shared next. [297] recommend using unsupervised ML techniques for feature extraction and supervised learning for classification and pre-

dictions. Given the scale of the IoT, a large amount of data is expected in the network and therefore requires a load balancing method, a load balancing algorithm based on a restricted Boltzmann machine is proposed in [298]. Online clustering scheme forms dynamic IoT data streams is described in [299]. Another work describing an ML application in IoT recommends a combination of PCA and regression for IoT to get better prediction [300]. Usage of clustering technique in embedded systems for IoT applications is presented in [301]. An application using denoising autoencoders for acoustic modeling in IoT is presented in [302].

F. LESSONS LEARNT

Key lessons drawn from the review of unsupervised learning in networking applications are summarized below:

1) A recommended and well-studied method for unsupervised Internet traffic classification in literature is data clustering combined with the latent representation learning on traffic feature set by using autoencoders. Min-max ensemble learning will help to increase the efficiency of unsupervised learning if required.
2) Semi-supervised learning is also an appropriate method for Internet traffic classification given some labeled traffic data and channel characteristics are available for initial model training.
3) Application of generative models and transfer learning for the Internet traffic classification has not been explored properly in literature and can be a potential research direction.
4) The overwhelming growth in network traffic and expected surge in traffic with the evolution of 5G and IoT also elevates the level of threat and anomalies in network traffic. To deal with these anomalies in Internet traffic, data clustering, PCA, SOM, and ART are well explored unsupervised learning techniques in the literature. Self-taught learning has also been explored as a potential solution for anomaly detection and remains a possible research direction for future research in anomaly detection in network traffic.
5) Current state of the art in dimensionality reduction in network traffic is based on PCA and multidimensional scaling. Autoencoders, t-SNE, and manifold learning are potential areas of research in terms of dimensionality reduction and visualization.

IV. FUTURE WORK: SOME RESEARCH CHALLENGES AND OPPORTUNITIES

This section provides a discussion on some open directions for future work and the relevant opportunities in applying unsupervised ML in the field of networking.

A. SIMPLIFIED NETWORK MANAGEMENT

While new network architectures such as SDN have been proposed in recent years to simplify network management, network operators are still expected to know too much, and to correlate between what they know about how their network is designed with the current network's condition through their monitoring sources. Operators who manage these requirements by wrestling with complexity manually will definitely welcome any respite that they can get from (semi-)automated unsupervised machine learning. As highlighted in by [303], for ML to become pervasive in networking, the "semantic gap"—which refers to the key challenge of transferring ML results into actionable insights and reports for the network operator—must be overcome. This can facilitate a shift from a reactive interaction style for network management, where the network manager is expected to check maps and graphs when things go wrong, to a proactive one, where automated reports and notifications are created for different services and network regions. Ideally, this would be abstract yet informative, such as Google Maps Directions, e.g. "there is heavier traffic than usual on your route" as well as suggestions about possible actions. This could be coupled with an automated correlation of different reports coming from different parts of the network. This will require a move beyond mere notifications and visualizations to more substantial synthesis through which potential sources of problems can be identified. Another example relates to making measurements more user-oriented. Most users would be more interested in QoE instead of QoS, i.e., how the current condition of the network affects their applications and services rather than just raw QoS metrics. The development of measurement objectives should be from a business-eyeball perspective—and not only through presenting statistics gathered through various tools and protocols such as traceroute, ping, BGP, etc. with the burden of putting the various pieces of knowledge together being on the user.

B. SEMI-SUPERVISED LEARNING FOR COMPUTER NETWORKS

Semi-supervised learning lies between supervised and unsupervised learning. The idea behind semi-supervised learning is to improve the learning ability by using unlabeled data incorporation with a small set of labeled examples. In computer networks, semi-supervised learning is partially used in anomaly detection and traffic classification and has great potential to be used with deep unsupervised learning architectures like generative adversarial networks for improving the state of the art in anomaly detection and traffic classification. Similarly, user behavior learning for cybersecurity can also be tackled in a semi-supervised fashion. A semi-supervised learning based anomaly detection approach is presented in [304]. The presented approach used large amounts of unlabeled samples together with labeled samples to build a better intrusion detection classifier. In particular, a single hidden layer feed-forward NN has trained to output a fuzzy membership vector. The results show that using unlabeled samples help significantly improve the classifier's performance. In another work, [305] have proposed semi-supervised learning with 97% accuracy to filter out non-malicious data in millions of queries that Domain Name Service (DNS) servers receive.

C. TRANSFER LEARNING IN COMPUTER NETWORKS

Transfer learning is an emerging ML technique in which knowledge learned from one problem is applied to a different but related problem [306]. Although it is often thought that for ML algorithms, the training and future data must be in the same feature space and must have the same distribution, this is not necessarily the case in many real-world applications. In such cases, it is desirable to have *transfer learning* or knowledge transfer between the different task domains. Transfer learning has been successfully applied in computer vision and NLP applications but its implementation for networking has not been witnessed—even though in principle, this can be useful in networking as well due to the similar nature of Internet traffic and enterprise network traffic in many respects. [307] used transfer learning based caching procedure for wireless networks providing backhaul offloading in 5G networks.

D. FEDERATED LEARNING IN COMPUTER NETWORKS

Federated learning is a collaborative ML technique, which does not make use of centralized training data, and works by distributing the processing on different machines. Federated learning is considered to be the next big thing in cloud networks as they ensure the privacy of the user data and less computation on the cloud to reduce the cost and energy [308]. System and method for network address management in the federated cloud are presented in [309] and the application of federated IoT and cloud computing for health care is presented in [310]. An end-to-end security architecture for federated cloud and IoT is presented in [311].

E. GENERATIVE ADVERSARIAL NETWORKS (GANS) IN COMPUTER NETWORKS

Adversarial networks—based on generative adversarial network (GAN) training originally proposed by Goodfellow and colleagues at the University of Montreal [312]—have recently emerged as a new technique using which machines can be trained to predict outcomes by only the observing the world (without necessarily being provided labeled data). An adversarial network has two NN models: a generator which is responsible for generating some type of data from some random input and a discriminator, which has the task of distinguishing between input from the generator or a real data set. The two NNs optimize themselves together resulting in a more realistic generation of data by the generator, and a better sense of what is plausible in the real world for the discriminator. [313] proposed a GAN for generating malware examples to attack a malware classifier and then proposes a defense against it. Another adversarial perturbation attack on malware classifier is proposed in [314]. The use of GANs for ML in networking can improve the performance of ML-based networking applications such as anomaly detection in which malicious users have an incentive to adversarial craft new attacks to avoid detection by network managers.

V. PITFALLS AND CAVEATS OF USING UNSUPERVISED ML IN NETWORKING

With the benefits and intriguing results of unsupervised learning, there also exist many shortcomings that are not addressed widely in the literature. Some potential pitfalls and caveats related to unsupervised learning are discussed next.

A. INAPPROPRIATE TECHNIQUE SELECTION

To start with, the first potential pitfall could be the selection of technique. Different unsupervised learning and predicting techniques may have excellent results on some applications while performing poorly on others—it is important to choose the best technique for the task at hand. Another reason could be a poor selection of features or parameters on which basis predictions are made—thus parameter optimization is also important for unsupervised algorithms.

B. LACK OF INTERPRETABILITY OF SOME UNSUPERVISED ML ALGORITHMS

Some unsupervised algorithms such as deep NNs operate as a black box, which makes it difficult to explain and interpret the working of such models. This makes the use of such techniques unsuitable for applications in which interpretability is important. As pointed out in [303], understandability of the semantics of the decisions made by ML is especially important for the operational success of ML in large-scale operational networks and its acceptance by operators, network managers, and users. But prediction accuracy and simplicity are often in conflict [315]. As an example, the greater accuracy of NNs accrues from its complex nature in which input variables are combined in a nonlinear fashion to build a complicated hard-to-explain model; with NNs it may not be possible to get interpretability as well since they make a tradeoff in which they sacrifice interpretability to achieve high accuracy. There are various ongoing research efforts that are focused on making techniques such as NNs less opaque [316]. Apart from the focus on NNs, there is a general interest in making AI and ML more explainable and interpretable—e.g., the Defense Advanced Research Projects Agency or DARPA's *explainable AI project*[2] is aiming to develop explainable AI models (leveraging various design options spanning the performance-vs-explainability trade-off space) that can explain the rationale of their decision-making so that users are able to appropriately trust these models particularly for new envisioned control applications in which optimization decisions are made autonomously by algorithms.

C. LACK OF OPERATIONAL SUCCESS OF ML IN NETWORKING

In literature, researchers have noted that despite substantial academic research, and practical applications of unsupervised learning in other fields, we see that there is a dearth of practical applications of ML solutions in operational

[2]https://www.darpa.mil/program/explainable-artificial-intelligence

networks—particular for applications such as network intrusion detection [303], which are challenging problems for a number of reasons including 1) the very high cost of errors; 2) the lack of training data; 3) the semantic gap between results and their operational interpretation; 4) enormous variability in input data; and finally, 5) fundamental difficulties in conducting sound performance evaluations. Even for other applications, the success of ML and its wide adoption in practical systems at scale lags the success of ML solutions in many other domains.

D. IGNORING SIMPLE NON-MACHINE-LEARNING BASED TOOLS

One should also keep in mind a common pitfall that academic researchers may suffer from which is not realizing that network operators may have simpler non-machine learning based solutions that may work as well as naïve ML-based solutions in practical settings. Failure to examine the ground realities of operational networks will undermine the effectiveness of ML-based solutions. We should expect ML-based solutions to augment and supplement rather than replace other non-machine-learning based solutions—at least for the foreseeable future.

E. OVERFITTING

Another potential issue with unsupervised models is overfitting; it corresponds to a model representing the noise or random error rather than learning the actual pattern in data. While commonly associated with supervised ML, the problem of overfitting lurks whenever we learn from data and thus is applicable to unsupervised ML as well. As illustrated in Figure 8, ideally speaking, we expect ML algorithms to provide improved performance with more data; but with increasing model complexity, performance starts to deteriorate after a certain point—although, it is possible to get poorer results empirically with increasing data when working with unoptimized out-of-the-box ML algorithms [317]. According to the Occam Razor principle, the model complexity should be commensurate with the amount of data available, and with overly complex models, the ability to predict and generalize diminishes. Two major reasons for overfitting could be the overly large size of the learning model and fewer sample data used for training purposes. Generally, data is divided into two portions (actual data and stochastic noise); due to the unavailability of labels or related information, unsupervised learning model can overfit the data, which causes issues in testing and deployment phase. Cross-validation, regularization, and Chi-squared testing are highly recommended for designing or tweaking an unsupervised learning algorithm to avoid overfitting [318].

F. DATA QUALITY ISSUES

It should be noted that all ML is data dependent, and the performance of ML algorithms is affected largely by the nature, volume, quality, and representation of data. In the case of unsupervised ML data quality issues must be carefully considered since any problem with the data quality will seriously mar the performance of ML algorithms. A potential problem is that dataset may be *imbalanced* if the samples size from one class is very much smaller or larger than the other classes [319]. In such imbalanced datasets, the algorithm must be careful not to ignore the rare class by assuming it to be noise. Although imbalanced datasets are more of a nuisance for supervised learning techniques, they may also pose problems for unsupervised and semi-supervised learning techniques.

G. INACCURATE MODEL BUILDING

It is difficult to build accurate and generic models since each model is optimized for certain kind of applications. Unsupervised ML models should be applied after carefully studying the application and the suitability of the algorithm in such settings [320]. For example, we highlight certain issues related to the unsupervised task of clustering: 1) random initialization in K-means is not recommended; 2) number of clusters is not known before the clustering operation as we do not have labels; 3) in the case of hierarchical clustering, we do not know when to stop and this can cause increase in the time complexity of the process, and 4) evaluating the clustering result is very tricky since the ground truth is mostly unknown.

H. MACHINE LEARNING IN ADVERSARIAL ENVIRONMENTS

Many networking problems, such as anomaly detection, are adversarial problems in which the malicious intruder is continually trying to outwit the network administrators (and the tools used by the network administrators). In such settings, machine learning that learns from historical data may not perform due to clever crafting of attacks specifically for circumventing any schemes based on previous data.

Due to these challenges, pitfalls, and weaknesses, due care must be exercised while using unsupervised and semi-supervised ML. These pitfalls can be avoided in part by using various best practices [321], such as end-to-end learning pipeline testing, visualization of the learning algorithm, regularization, proper feature engineering, dropout, sanity checks through human inspection—whichever is appropriate for the problem's context.

VI. CONCLUSIONS

We have provided a comprehensive survey of machine learning tasks, latest unsupervised learning techniques, and trends, along with a detailed discussion of the applications of these techniques in networking related tasks. Despite the recent wave of success of unsupervised learning, there is a scarcity of unsupervised learning literature for computer networking applications, which this survey aims to address. The few previously published survey papers differ from our work in their focus, scope, and breadth; we have written this paper in a manner that carefully synthesizes the insights from these

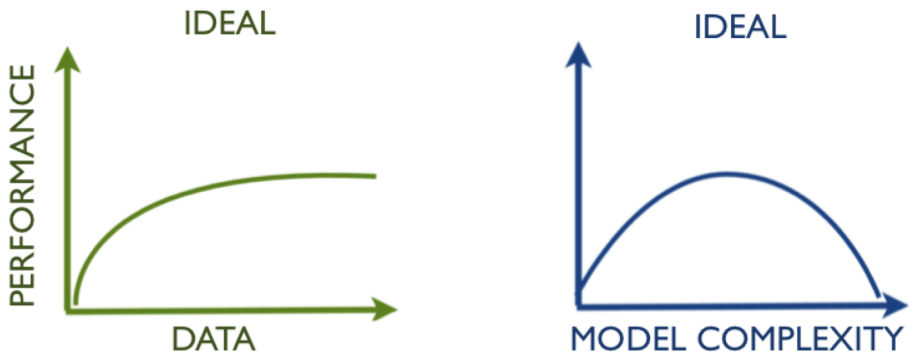

FIGURE 8. Intuitively, we expect the ML model's performance to improve with more data but to deteriorate in performance if the model becomes overly complex for the data. Figure adapted from [317].

survey papers while also providing contemporary coverage of recent advances. Due to the versatility and evolving nature of computer networks, it was impossible to cover each and every application; however, an attempt has been made to cover all the major networking applications of unsupervised learning and the relevant techniques. We have also presented concise future work and open research areas in the field of networking, which is related to unsupervised learning, coupled with a brief discussion of significant pitfalls and challenges in using unsupervised machine learning in networks.

VII. REFERENCES
REFERENCES

[1] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks," in Cognitive radio, software defined radio, and adaptive wireless systems, pp. 17–41, Springer, 2007.

[2] S. Latif, F. Pervez, M. Usama, and J. Qadir, "Artificial intelligence as an enabler for cognitive self-organizing future networks," arXiv preprint arXiv:1702.02823, 2017.

[3] J. Qadir, K.-L. A. Yau, M. A. Imran, Q. Ni, and A. V. Vasilakos, "IEEE access special section editorial: Artificial intelligence enabled networking," IEEE Access, vol. 3, pp. 3079–3082, 2015.

[4] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 4, pp. 70–73, 2014.

[5] S. Shenker, M. Casado, T. Koponen, N. McKeown, et al., "The future of networking, and the past of protocols," Open Networking Summit, vol. 20, pp. 1–30, 2011.

[6] A. Malik, J. Qadir, B. Ahmad, K.-L. A. Yau, and U. Ullah, "Qos in ieee 802.11-based wireless networks: a contemporary review," Journal of Network and Computer Applications, vol. 55, pp. 24–46, 2015.

[7] N. Feamster and J. Rexford, "Why (and how) networks should run themselves," arXiv preprint arXiv:1710.11583, 2017.

[8] J. Jiang, V. Sekar, I. Stoica, and H. Zhang, "Unleashing the potential of data-driven networking," in International Conference on Communication Systems and Networks, pp. 110–126, Springer, 2017.

[9] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Computer networks, vol. 51, no. 12, pp. 3448–3470, 2007.

[10] T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using Machine Learning," Communications Surveys & Tutorials, IEEE, vol. 10, no. 4, pp. 56–76, 2008.

[11] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," Communications Surveys & Tutorials, IEEE, vol. 15, no. 3, pp. 1136–1159, 2013.

[12] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1996–2018, 2014.

[13] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications Surveys Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.

[14] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self organizing cellular networks," IEEE Communications Surveys & Tutorials, 2017.

[15] A. Meshram and C. Haas, "Anomaly detection in industrial networks using machine learning: A roadmap," in Machine Learning for Cyber Physical Systems, pp. 65–72, Springer, 2017.

[16] Z. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrowâĂŹs intelligent network traffic control systems," IEEE Communications Surveys & Tutorials, 2017.

[17] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," arXiv preprint arXiv:1701.02145, 2017.

[18] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," arXiv preprint arXiv:1807.11023, 2018.

[19] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: A survey," Digital Communications and Networks, vol. 4, no. 3, pp. 161–175, 2018.

[20] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," Journal of Internet Services and Applications, vol. 9, no. 1, p. 16, 2018.

[21] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for internet of things," International Journal of Machine Learning and Cybernetics, vol. 9, no. 8, pp. 1399–1417, 2018.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[23] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," APSIPA Transactions on Signal and Information Processing, vol. 3, 2014.

[24] J. Qadir, "Artificial intelligence based cognitive routing for cognitive radio networks," Artificial Intelligence Review, vol. 45, no. 1, pp. 25–96, 2016.

[25] N. Ahad, J. Qadir, and N. Ahsan, "Neural networks in wireless networks: Techniques, applications and guidelines," Journal of Network and Computer Applications, vol. 68, pp. 1–27, 2016.

[26] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, Feature extraction: foundations and applications, vol. 207. Springer, 2008.

[27] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in International conference on artificial intelligence and statistics, pp. 215–223, 2011.

[28] M. J. S. M. S. Mohammad Lotfollahi, Ramin Shirali, "Deep packet: A novel approach for encrypted traffic classification using deep learning," 2017.

[29] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning,"

in Information Networking (ICOIN), 2017 International Conference on, pp. 712–717, IEEE, 2017.
[30] M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," in Neural Networks (IJCNN), 2017 International Joint Conference on, pp. 3854–3861, IEEE, 2017.
[31] R. C. Aygun and A. G. Yavuz, "Network anomaly detection with stochastically improved autoencoder based models," in Cyber Security and Cloud Computing (CSCloud), 2017 IEEE 4th International Conference on, pp. 193–198, IEEE, 2017.
[32] M. K. Putchala, Deep Learning Approach for Intrusion Detection System (IDS) in the Internet of Things (IoT) Network using Gated Recurrent Neural Networks (GRU). PhD thesis, Wright State University, 2017.
[33] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," 2017.
[34] E. Aguiar, A. Riker, M. Mu, and S. Zeadally, "Real-time qoe prediction for multimedia applications in wireless mesh networks," in Consumer Communications and Networking Conference (CCNC), 2012 IEEE, pp. 592–596, IEEE, 2012.
[35] K. Piamrat, A. Ksentini, C. Viho, and J.-M. Bonnin, "Qoe-aware admission control for multimedia applications in ieee 802.11 wireless networks," in Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th, pp. 1–5, IEEE, 2008.
[36] K. Karra, S. Kuzdeba, and J. Petersen, "Modulation recognition using hierarchical deep neural networks," in Dynamic Spectrum Access Networks (DySPAN), 2017 IEEE International Symposium on, pp. 1–3, IEEE, 2017.
[37] M. Zhang, M. Diao, and L. Guo, "Convolutional neural networks for automatic cognitive radio waveform recognition," IEEE Access, vol. 5, pp. 11074–11082, 2017.
[38] J. Moysen and L. Giupponi, "From 4g to 5g: Self-organized network management meets machine learning," arXiv preprint arXiv:1707.09300, 2017.
[39] X. Xie, D. Wu, S. Liu, and R. Li, "Iot data analytics using deep learning," arXiv preprint arXiv:1708.03854, 2017.
[40] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT Press, 2016.
[41] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, 2015.
[42] Y. Bengio, "Learning deep architectures for AI," Foundations and trends® in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009.
[43] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.
[44] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al., "Greedy layer-wise training of deep networks," Advances in neural information processing systems, vol. 19, p. 153, 2007.
[45] C. Poultney, S. Chopra, Y. L. Cun, et al., "Efficient learning of sparse representations with an energy-based model," in Advances in neural information processing systems, pp. 1137–1144, 2006.
[46] J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, Q. V. Le, and A. Y. Ng, "On optimization methods for deep learning," in Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 265–272, 2011.
[47] C. Doersch, "Tutorial on variational autoencoders," arXiv preprint arXiv:1606.05908, 2016.
[48] T. Kohonen, "The self-organizing map," Proceedings of the IEEE, vol. 78, no. 9, pp. 1464–1480, 1990.
[49] T. Kohonen, "The self-organizing map," Neurocomputing, vol. 21, no. 1, pp. 1–6, 1998.
[50] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," Psychological review, vol. 65, no. 6, p. 386, 1958.
[51] S. S. Haykin, Neural networks and learning machines, vol. 3. Pearson Education Upper Saddle River, 2009.
[52] G. A. Carpenter and S. Grossberg, Adaptive resonance theory. Springer, 2010.
[53] J. Karhunen, T. Raiko, and K. Cho, "Unsupervised deep learning: A short review," Advances in Independent Component Analysis and Learning Machines, p. 125, 2015.
[54] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in Proceedings of the 26th Annual International Conference on Machine Learning, pp. 609–616, ACM, 2009.
[55] S. Baraković and L. Skorin-Kapov, "Survey and challenges of QoE management issues in wireless networks," Journal of Computer Networks and Communications, vol. 2013, 2013.
[56] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," arXiv preprint arXiv:1312.6026, 2013.
[57] M. Klapper-Rybicka, N. N. Schraudolph, and J. Schmidhuber, "Unsupervised learning in LSTM recurrent neural networks," in Artificial Neural Networks ICANN 2001, pp. 684–691, Springer, 2001.
[58] G. E. Hinton, "Boltzmann machine," Scholarpedia, vol. 2, no. 5, p. 1668, 2007. revision #91075.
[59] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in Artificial Intelligence and Statistics, pp. 448–455, 2009.
[60] K. Tsagkaris, A. Katidiotis, and P. Demestichas, "Neural network-based learning schemes for cognitive radio systems," Computer Communications, vol. 31, no. 14, pp. 3394–3404, 2008.
[61] F. H. V. Teles and L. L. Lee, "A Neural Architecture Based on the Adaptive Resonant Theory and Recurrent Neural Networks," IJCSA, vol. 4, no. 3, pp. 45–56, 2007.
[62] D. Munaretto, D. Zucchetto, A. Zanella, and M. Zorzi, "Data-driven QoE optimization techniques for multi-user wireless networks," in Computing, Networking and Communications (ICNC), 2015 International Conference on, pp. 653–657, IEEE, 2015.
[63] L. Badia, D. Munaretto, A. Testolin, A. Zanella, and M. Zorzi, "Cognition-based networks: Applying cognitive science to multimedia wireless networking," in A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on, pp. 1–6, IEEE, 2014.
[64] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey," A Review of Machine Learning Techniques for Processing Multimedia Content, vol. 1, pp. 9–16, 2004.
[65] P. Berkhin, "A survey of clustering data mining techniques," in Grouping multidimensional data, pp. 25–71, Springer, 2006.
[66] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," Communications Surveys & Tutorials, IEEE, vol. 16, no. 1, pp. 303–336, 2014.
[67] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques," in Passive and Active Network Measurement, pp. 205–214, Springer, 2004.
[68] R. Xu and D. Wunsch, "Survey of clustering algorithms," IEEE Transactions on neural networks, vol. 16, no. 3, pp. 645–678, 2005.
[69] M. Rehman and S. A. Mehdi, "Comparison of density-based clustering algorithms," Lahore College for Women University, Lahore, Pakistan, University of Management and Technology, Lahore, Pakistan, 2005.
[70] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 133–142, ACM, 2007.
[71] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," in Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38, pp. 333–342, Australian Computer Society, Inc., 2005.
[72] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 1855–1870, ACM, 2015.
[73] P. Mangiameli, S. K. Chen, and D. West, "A comparison of SOM neural network and hierarchical clustering methods," European Journal of Operational Research, vol. 93, no. 2, pp. 402–417, 1996.
[74] P. Orbanz and Y. W. Teh, "Bayesian nonparametric models," in Encyclopedia of Machine Learning, pp. 81–89, Springer, 2011.
[75] B. Kurt, A. T. Cemgil, M. Mungan, and E. Saygun, "Bayesian nonparametric clustering of network traffic data,"
[76] X. Jin and J. Han, Encyclopedia of Machine Learning, ch. Partitional Clustering, pp. 766–766. Boston, MA: Springer US, 2010.
[77] S. R. Gaddam, V. V. Phoha, and K. S. Balagani, "K-means+ ID3: A novel method for supervised anomaly detection by cascading k-means clustering and ID3 decision tree learning methods," IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, pp. 345–354, 2007.
[78] L. Yingqiu, L. Wei, and L. Yunchun, "Network traffic classification using K-Means clustering," in Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS), 2007., pp. 360–365, IEEE, 2007.
[79] M. Jianliang, S. Haikun, and B. Ling, "The application on intrusion detection based on k-means cluster algorithm," in Information Technology and

Applications, 2009. IFITA'09. International Forum on, vol. 1, pp. 150–152, IEEE, 2009.

[80] R. Chitrakar and H. Chuanhe, "Anomaly detection using support vector machine classification with k-medoids clustering," in Internet (AH-ICI), 2012 Third Asian Himalayas International Conference on, pp. 1–5, IEEE, 2012.

[81] R. Chitrakar and H. Chuanhe, "Anomaly based intrusion detection using hybrid learning approach of combining k-medoids clustering and naive Bayes classification," in Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on, pp. 1–5, IEEE, 2012.

[82] M. A. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 24, no. 3, pp. 381–396, 2002.

[83] M. E. Newman and E. A. Leicht, "Mixture models and exploratory analysis in networks," Proceedings of the National Academy of Sciences, vol. 104, no. 23, pp. 9564–9569, 2007.

[84] M. Bahrololum and M. Khaleghi, "Anomaly intrusion detection system using gaussian mixture model," in Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on, vol. 1, pp. 1162–1167, IEEE, 2008.

[85] W. Chimphlee, A. H. Abdullah, M. N. M. Sap, S. Srinoy, and S. Chimphlee, "Anomaly-based intrusion detection using fuzzy rough clustering," in Hybrid Information Technology, 2006. ICHIT'06. International Conference on, vol. 1, pp. 329–334, IEEE, 2006.

[86] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, C. Ziemlicki, and Z. Smoreda, "Not all apps are created equal: Analysis of spatiotemporal heterogeneity in nationwide mobile service usage," 2017.

[87] M. Adda, K. Qader, and M. Al-Kasassbeh, "Comparative analysis of clustering techniques in network traffic faults classification," International Journal of Innovative Research in Computer and Communication Engineering, vol. 5, no. 4, pp. 6551–6563, 2017.

[88] A. Vlăduţu, D. Comăneci, and C. Dobre, "Internet traffic classification based on flows' statistical properties with machine learning," International Journal of Network Management, vol. 27, no. 3, 2017.

[89] J. Liu, Y. Fu, J. Ming, Y. Ren, L. Sun, and H. Xiong, "Effective and real-time in-app activity analysis in encrypted internet traffic streams," in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 335–344, ACM, 2017.

[90] M. S. Parwez, D. Rawat, and M. Garuba, "Big data analytics for user activity analysis and user anomaly detection in mobile wireless network," IEEE Transactions on Industrial Informatics, 2017.

[91] T. Lorido-Botran, S. Huerta, L. Tomás, J. Tordsson, and B. Sanz, "An unsupervised approach to online noisy-neighbor detection in cloud data centers," Expert Systems with Applications, vol. 89, pp. 188–204, 2017.

[92] G. Frishman, Y. Ben-Itzhak, and O. Margalit, "Cluster-based load balancing for better network security," in Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, pp. 7–12, ACM, 2017.

[93] G. R. Kumar, N. Mangathayaru, and G. Narsimha, "A feature clustering based dimensionality reduction for intrusion detection (fcbdr).," IADIS International Journal on Computer Science & Information Systems, vol. 12, no. 1, 2017.

[94] T. Wiradinata and A. S. Paramita, "Clustering and feature selection technique for improving internet traffic classification using k-nn," 2016.

[95] C. M. Bishop, "Latent variable models," in Learning in graphical models, pp. 371–403, Springer, 1998.

[96] A. Skrondal and S. RABE-HESKETH, "Latent variable modelling: a survey," Scandinavian Journal of Statistics, vol. 34, no. 4, pp. 712–745, 2007.

[97] C. M. Bishop, Neural networks for pattern recognition. Oxford university press, 1995.

[98] J. Josse and F. Husson, "Selecting the number of components in principal component analysis using cross-validation approximations," Computational Statistics & Data Analysis, vol. 56, no. 6, pp. 1869–1879, 2012.

[99] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," Neural networks, vol. 13, no. 4, pp. 411–430, 2000.

[100] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," IEEE Transactions on Knowledge and Data Engineering,, vol. 25, no. 6, pp. 1336–1353, 2013.

[101] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in Advances in neural information processing systems, pp. 556–562, 2001.

[102] M. O. Duff, Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes. PhD thesis, University of Massachusetts Amherst, 2002.

[103] M. J. Beal, Variational algorithms for approximate Bayesian inference. University of London United Kingdom, 2003.

[104] T. P. Minka, A family of algorithms for approximate Bayesian inference. PhD thesis, Massachusetts Institute of Technology, 2001.

[105] H. Wang and D.-Y. Yeung, "Towards Bayesian deep learning: A survey," arXiv preprint arXiv:1604.01662, 2016.

[106] C. DuBois, J. R. Foulds, and P. Smyth, "Latent set models for two-mode network data.," in ICWSM, 2011.

[107] J. R. Foulds, C. DuBois, A. U. Asuncion, C. T. Butts, and P. Smyth, "A dynamic relational infinite feature model for longitudinal social networks.," in AISTATS, vol. 11, pp. 287–295, 2011.

[108] J.-A. Hernández and I. W. Phillips, "Weibull mixture model to characterise end-to-end internet delay at coarse time-scales," IEE Proceedings-Communications, vol. 153, no. 2, pp. 295–304, 2006.

[109] J. M. Agosta, J. Chandrashekar, M. Crovella, N. Taft, and D. Ting, "Mixture models of endhost network traffic," in INFOCOM, 2013 Proceedings IEEE, pp. 225–229, IEEE, 2013.

[110] R. Yan and R. Liu, "Principal component analysis based network traffic classification," Journal of computers, vol. 9, no. 5, pp. 1234–1240, 2014.

[111] X. Xu and X. Wang, "An adaptive network intrusion detection method based on PCA and support vector machines," in Advanced Data Mining and Applications, pp. 696–703, Springer, 2005.

[112] X. Guan, W. Wang, and X. Zhang, "Fast intrusion detection based on a non-negative matrix factorization model," Journal of Network and Computer Applications, vol. 32, no. 1, pp. 31–44, 2009.

[113] Z. Albatineh and F. Salem, "New blind multiuser detection in DS-CDMA based on extension of efficient fast independent component analysis (EF-ICA)," in 2013 4th International Conference on Intelligent Systems, Modelling and Simulation, pp. 543–548, IEEE, 2013.

[114] N. Ahmed, S. S. Kanhere, and S. Jha, "Probabilistic coverage in wireless sensor networks," in Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on, pp. 8–pp, IEEE, 2005.

[115] V. Chatzigiannakis, S. Papavassiliou, M. Grammatikou, and B. Maglaris, "Hierarchical anomaly detection in distributed large-scale sensor networks," in Computers and Communications, 2006. ISCC'06. Proceedings. 11th IEEE Symposium on, pp. 761–767, IEEE, 2006.

[116] R. Gu, H. Wang, and Y. Ji, "Early traffic identification using Bayesian networks," in Network Infrastructure and Digital Content, 2010 2nd IEEE International Conference on, pp. 564–568, IEEE, 2010.

[117] J. Xu and C. Shelton, "Continuous time Bayesian networks for host level network intrusion detection," Machine learning and knowledge discovery in databases, pp. 613–627, 2008.

[118] N. Al-Rousan, S. Haeri, and L. Trajković, "Feature selection for classification of BGP anomalies using Bayesian models," in Machine Learning and Cybernetics (ICMLC), 2012 International Conference on, vol. 1, pp. 140–147, IEEE, 2012.

[119] D.-p. Liu, M.-w. Zhang, and T. Li, "Network traffic analysis using refined Bayesian reasoning to detect flooding and port scan attacks," in Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on, pp. 1000–1004, IEEE, 2008.

[120] M. Ishiguro, H. Suzuki, I. Murase, and H. Ohno, "Internet threat detection system using Bayesian estimation," in Proc. The 16th Annual Computer Security Incident Handling Conference, 2004.

[121] D. Janakiram, V. Reddy, and A. P. Kumar, "Outlier detection in wireless sensor networks using Bayesian belief networks," in Communication System Software and Middleware, 2006. Comsware 2006. First International Conference on, pp. 1–6, IEEE, 2006.

[122] S. Haykin, K. Huber, and Z. Chen, "Bayesian sequential state estimation for MIMO wireless communications," Proceedings of the IEEE, vol. 92, no. 3, pp. 439–454, 2004.

[123] S. Ito and N. Kawaguchi, "Bayesian based location estimation system using wireless LAN," in Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on, pp. 273–278, IEEE, 2005.

[124] S. Liu, J. Hu, S. Hao, and T. Song, "Improved em method for internet traffic classification," in Knowledge and Smart Technology (KST), 2016 8th International Conference on, pp. 13–17, IEEE, 2016.

[125] H. Shi, H. Li, D. Zhang, C. Cheng, and W. Wu, "Efficient and robust feature extraction and selection for traffic classification," Computer Networks, vol. 119, pp. 1–16, 2017.

[126] S. Troia, G. Sheng, R. Alvizu, G. A. Maier, and A. Pattavina, "Identification of tidal-traffic patterns in metro-area mobile networks via matrix factorization based model," in Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on, pp. 297–301, IEEE, 2017.

[127] L. Nie, D. Jiang, and Z. Lv, "Modeling network traffic for traffic matrix estimation and anomaly detection based on bayesian network in cloud computing networks," Annals of Telecommunications, vol. 72, no. 5-6, pp. 297–305, 2017.

[128] J.-h. Bang, Y.-J. Cho, and K. Kang, "Anomaly detection of network-initiated lte signaling traffic in wireless sensor and actuator networks based on a hidden semi-markov model," Computers & Security, vol. 65, pp. 108–120, 2017.

[129] X. Chen, K. Irie, D. Banks, R. Haslinger, J. Thomas, and M. West, "Scalable bayesian modeling, monitoring and analysis of dynamic network flow data," Journal of the American Statistical Association, no. just-accepted, 2017.

[130] B. Mokhtar and M. Eltoweissy, "Big data and semantics management system for computer networks," Ad Hoc Networks, vol. 57, pp. 32–51, 2017.

[131] A. Furno, M. Fiore, and R. Stanica, "Joint spatial and temporal classification of mobile traffic demands," in INFOCOM–36th Annual IEEE International Conference on Computer Communications, 2017.

[132] M. Malli, N. Said, and A. Fadlallah, "A new model for rating usersâĂŹ profiles in online social networks," Computer and Information Science, vol. 10, no. 2, p. 39, 2017.

[133] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," Science, vol. 290, no. 5500, pp. 2323–2326, 2000.

[134] E. Keogh and A. Mueen, "Curse of dimensionality," in Encyclopedia of Machine Learning, pp. 257–258, Springer, 2011.

[135] P. Pudil and J. Novovičová, "Novel methods for feature subset selection with respect to problem knowledge," in Feature Extraction, Construction and Selection, pp. 101–116, Springer, 1998.

[136] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in International Conference on Machine Learning, vol. 3, pp. 856–863, 2003.

[137] W. M. Hartmann, "Dimension reduction vs. variable selection," in Applied Parallel Computing. State of the Art in Scientific Computing, pp. 931–938, Springer, 2006.

[138] I. K. Fodor, "A survey of dimension reduction techniques," Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory, 2002.

[139] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," science, vol. 290, no. 5500, pp. 2319–2323, 2000.

[140] C. Bishop, M. SvensÃĪn, and C. K. Williams, "Gtm: The generative topographic mapping," 1998.

[141] T. Hastie and W. Stuetzle, "Principal curves," Journal of the American Statistical Association, vol. 84, no. 406, pp. 502–516, 1989.

[142] D. Lee, "Estimations of principal curves," http://www.dgp.toronto.edu/~dwlee/pcurve/pcurve_csc2515.pdf, 2002.

[143] J. B. Kruskal, "Nonmetric multidimensional scaling: a numerical method," Psychometrika, vol. 29, no. 2, pp. 115–129, 1964.

[144] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, no. Nov, pp. 2579–2605, 2008.

[145] J. Cao, Z. Fang, G. Qu, H. Sun, and D. Zhang, "An accurate traffic classification model based on support vector machines," International Journal of Network Management, vol. 27, no. 1, 2017.

[146] W. Zhou, X. Zhou, S. Dong, and B. Lubomir, "A som and pnn model for network traffic classification," Boletín Técnico, vol. 55, no. 1, pp. 174–182, 2017.

[147] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," Pattern Recognition, vol. 58, pp. 121–134, 2016.

[148] M. Nicolau, J. McDermott, et al., "A hybrid autoencoder and density estimation model for anomaly detection," in International Conference on Parallel Problem Solving from Nature, pp. 717–726, Springer, 2016.

[149] S. T. Ikram and A. K. Cherukuri, "Improving accuracy of intrusion detection model using pca and optimized svm," Journal of computing and information technology, vol. 24, no. 2, pp. 133–148, 2016.

[150] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, "A mobile network planning tool based on data analytics," Mobile Information Systems, vol. 2017, 2017.

[151] S. A. Ossia, A. S. Shamsabadi, A. Taheri, H. R. Rabiee, N. Lane, and H. Haddadi, "A hybrid deep learning architecture for privacy-preserving mobile analytics," arXiv preprint arXiv:1703.02952, 2017.

[152] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Distributed deep learning models for wireless signal classification with low-cost spectrum sensors," arXiv preprint arXiv:1707.08908, 2017.

[153] M. H. Sarshar, Analyzing Large Scale Wi-Fi Data Using Supervised and Unsupervised Learning Techniques. PhD thesis, 2017.

[154] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in ACM Sigmod Record, vol. 29, pp. 427–438, ACM, 2000.

[155] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 535–548, Springer, 2002.

[156] W. Jin, A. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," Advances in Knowledge Discovery and Data Mining, pp. 577–593, 2006.

[157] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in Proceedings of the 18th ACM conference on Information and knowledge management, pp. 1649–1652, ACM, 2009.

[158] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," Pattern Recognition Letters, vol. 24, no. 9, pp. 1641–1650, 2003.

[159] M. Goldstein and S. Uchida, "Behavior analysis using unsupervised anomaly detection," in The 10th Joint Workshop on Machine Perception and Robotics (MPR 2014). Online, 2014.

[160] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," PloS one, vol. 11, no. 4, p. e0152173, 2016.

[161] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," KI-2012: Poster and Demo Track, pp. 59–63, 2012.

[162] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM computing surveys (CSUR), vol. 41, no. 3, p. 15, 2009.

[163] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network traffic classification techniques and comparative analysis using machine learning algorithms," in Computer and Communications (ICCC), 2016 2nd IEEE International Conference on, pp. 2451–2455, IEEE, 2016.

[164] Y. Dhote, S. Agrawal, and A. J. Deen, "A survey on feature selection techniques for internet traffic classification," in Computational Intelligence and Communication Networks (CICN), 2015 International Conference on, pp. 1375–1380, IEEE, 2015.

[165] Y. Huang, Y. Li, and B. Qiang, "Internet traffic classification based on min-max ensemble feature selection," in Neural Networks (IJCNN), 2016 International Joint Conference on, pp. 3485–3492, IEEE, 2016.

[166] L. Zhen and L. Qiong, "A new feature selection method for Internet traffic classification using ML," Physics Procedia, vol. 33, pp. 1338–1345, 2012.

[167] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," Parallel and Distributed Systems, IEEE Transactions on, vol. 24, no. 1, pp. 104–117, 2013.

[168] J. Erman, A. Mahanti, and M. Arlitt, "Qrp05-4: Internet traffic identification using machine learning," in Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE, pp. 1–6, IEEE, 2006.

[169] J. Kornycky, O. Abdul-Hameed, A. Kondoz, and B. C. Barber, "Radio frequency traffic classification over WLAN," IEEE/ACM Transactions on Networking, vol. 25, no. 1, pp. 56–68, 2017.

[170] X. Liu, L. Pan, and X. Sun, "Real-time traffic status classification based on gaussian mixture model," in Data Science in Cyberspace (DSC), IEEE International Conference on, pp. 573–578, IEEE, 2016.

[171] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in Proceedings of the 2006 SIGCOMM workshop on Mining network data, pp. 281–286, ACM, 2006.

[172] T. T. Nguyen and G. Armitage, "Clustering to assist supervised machine learning for real-time IP traffic classification," in Communications, 2008. ICC'08. IEEE International Conference on, pp. 5857–5862, IEEE, 2008.

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

[174] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," ACM SIGCOMM Computer Communication Review, vol. 36, no. 2, pp. 23–26, 2006.

[175] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l, pp. 250–257, IEEE, 2005.

[176] T. P. Oliveira, J. S. Barbar, and A. S. Soares, "Computer network traffic prediction: a comparison between traditional and deep learning neural networks," International Journal of Big Data Intelligence, vol. 3, no. 1, pp. 28–37, 2016.

[177] N. Shrivastava and A. Dubey, "Internet traffic data categorization using particle of swarm optimization algorithm," in Colossal Data Analysis and Networking (CDAN), Symposium on, pp. 1–8, IEEE, 2016.

[178] T. Bakhshi and B. Ghita, "On Internet traffic classification: A two-phased machine learning approach," Journal of Computer Networks and Communications, vol. 2016, 2016.

[179] J. Yang, J. Deng, S. Li, and Y. Hao, "Improved traffic detection with support vector machine based on restricted Boltzmann machine," Soft Computing, vol. 21, no. 11, pp. 3101–3112, 2017.

[180] R. Gonzalez, F. Manco, A. Garcia-Duran, J. Mendes, F. Huici, S. Niccolini, and M. Niepert, "Net2Vec: Deep learning for the network," arXiv preprint arXiv:1705.03881, 2017.

[181] M. E. Aminanto and K. Kim, "Deep learning-based feature selection for intrusion detection system in transport layer," http://caislab.kaist.ac.kr/publication/paper_files/2016/20160623_AM.pdf, 2016.

[182] L. Nie, D. Jiang, S. Yu, and H. Song, "Network traffic prediction based on deep belief network in wireless mesh backbone networks," in Wireless Communications and Networking Conference (WCNC), 2017 IEEE, pp. 1–5, IEEE, 2017.

[183] C. Zhang, J. Jiang, and M. Kamel, "Intrusion detection using hierarchical neural networks," Pattern Recognition Letters, vol. 26, no. 6, pp. 779–791, 2005.

[184] B. C. Rhodes, J. A. Mahaffey, and J. D. Cannady, "Multiple self-organizing maps for intrusion detection," in Proceedings of the 23rd national information systems security conference, pp. 16–19, 2000.

[185] H. G. Kayacik, M. Heywood, et al., "On the capability of an SOM based intrusion detection system," in Neural Networks, 2003. Proceedings of the International Joint Conference on, vol. 3, pp. 1808–1813, IEEE, 2003.

[186] S. Zanero, "Analyzing TCP traffic patterns using self organizing maps," in Image Analysis and Processing–ICIAP 2005, pp. 83–90, Springer, 2005.

[187] P. Lichodzijewski, A. N. Zincir-Heywood, and M. I. Heywood, "Host-based intrusion detection using self-organizing maps," in IEEE international joint conference on neural networks, pp. 1714–1719, 2002.

[188] P. Lichodzijewski, A. N. Zincir-Heywood, and M. I. Heywood, "Dynamic intrusion detection using self-organizing maps," in The 14th Annual Canadian Information Technology Security Symposium (CITSS), Citeseer, 2002.

[189] M. Amini, R. Jalili, and H. R. Shahriari, "RT-UNNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks," Computers & Security, vol. 25, no. 6, pp. 459–468, 2006.

[190] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," Expert systems with Applications, vol. 29, no. 4, pp. 713–722, 2005.

[191] V. Golovko and L. Vaitsekhovich, "Neural network techniques for intrusion detection," in Proc. Int. Conf. Neural Networks and Artificial Intelligence, pp. 65–69, 2006.

[192] A. P. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading k-means clustering and C4.5 decision tree algorithm," Procedia Engineering, vol. 30, pp. 174–182, 2012.

[193] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," Computer Communications, vol. 35, no. 7, pp. 772–783, 2012.

[194] S. Zanero and S. M. Savaresi, "Unsupervised learning techniques for an intrusion detection system," in Proceedings of the 2004 ACM symposium on Applied computing, pp. 412–419, ACM, 2004.

[195] S. Zhong, T. M. Khoshgoftaar, and N. Seliya, "Clustering-based network intrusion detection," International Journal of reliability, Quality and safety Engineering, vol. 14, no. 02, pp. 169–187, 2007.

[196] N. Greggio, "Anomaly detection in idss by means of unsupervised greedy learning of finite mixture models," Soft Computing, pp. 1–16, 2017.

[197] W. Wang and R. Battiti, "Identifying intrusions in computer networks with Principal Component Analysis," in Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on, pp. 8–pp, IEEE, 2006.

[198] V. Golovko, L. U. Vaitsekhovich, P. Kochurko, U. S. Rubanau, et al., "Dimensionality reduction and attack recognition using neural network approaches," in Neural Networks, 2007. IJCNN 2007. International Joint Conference on, pp. 2734–2739, IEEE, 2007.

[199] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson, "2005 CSI/FBI computer crime and security survey," Computer Security Journal, 2005.

[200] Symantec, "Internet security threat report." https://www.symantec.com/security-center/threat-report, 2016. Accessed: 2017-02-02.

[201] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," Expert Systems with Applications, vol. 36, no. 10, pp. 11994–12000, 2009.

[202] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," Knowledge-based systems, vol. 78, pp. 13–21, 2015.

[203] J. Mazel, P. Casas, R. Fontugne, K. Fukuda, and P. Owezarski, "Hunting attacks in the dark: clustering and correlation analysis for unsupervised anomaly detection," International Journal of Network Management, vol. 25, no. 5, pp. 283–305, 2015.

[204] C. Sony and K. Cho, "Traffic data repository at the WIDE project," in Proceedings of USENIX 2000 Annual Technical Conference: FREENIX Track, pp. 263–270, 2000.

[205] E. E. Papalexakis, A. Beutel, and P. Steenkiste, "Network anomaly detection using co-clustering," in Encyclopedia of Social Network Analysis and Mining, pp. 1054–1068, Springer, 2014.

[206] V. Mišković, M. Milosavljević, S. Adamović, and A. Jevremović, "Application of hybrid incremental machine learning methods to anomaly based intrusion detection," methods, vol. 5, p. 6, 2014.

[207] N. F. Haq, A. R. Onik, M. Avishek, K. Hridoy, M. Rafni, F. M. Shah, and D. M. Farid, "Application of machine learning approaches in intrusion detection system: a survey," International Journal of Advanced Research in Artificial Intelligence, 2015.

[208] T. Hämäläinen, "Artificial immune system based intrusion detection: innate immunity using an unsupervised learning approach," International Journal of Digital Content Technology and its Applications (JDCTA), 2014.

[209] G. K. Chaturvedi, A. K. Chaturvedi, and V. R. More, "A study of intrusion detection system for cloud network using FC-ANN algorithm," 2016.

[210] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42–57, 2013.

[211] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," IEEE Communications Surveys & Tutorials, vol. 17, no. 1, pp. 70–91, 2015.

[212] R. Mitchell and R. Chen, "A survey of intrusion detection in wireless network applications," Computer Communications, vol. 42, pp. 1–23, 2014.

[213] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp. 19–31, 2016.

[214] L. Xiao, Y. Chen, and C. K. Chang, "Bayesian model averaging of Bayesian network classifiers for intrusion detection," in Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International, pp. 128–133, IEEE, 2014.

[215] B. Al-Musawi, P. Branch, and G. Armitage, "BGP anomaly detection techniques: A survey," IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 377–396, 2017.

[216] B. AsSadhan, K. Zeb, J. Al-Muhtadi, and S. Alshebeili, "Anomaly detection based on LRD behavior analysis of decomposed control and data planes network traffic using soss and farima models," IEEE Access, 2017.

[217] A. Kulakov, D. Davcev, and G. Trajkovski, "Application of wavelet neural-networks in wireless sensor networks," in Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on, pp. 262–267, IEEE, 2005.

[218] S. G. Akojwar and R. M. Patrikar, "Improving life time of wireless sensor networks using neural network based classification techniques with coop-

erative routing," International Journal of Communications, vol. 2, no. 1, pp. 75–86, 2008.

[219] C. Li, X. Xie, Y. Huang, H. Wang, and C. Niu, "Distributed data mining based on deep neural network for wireless sensor network," International Journal of Distributed Sensor Networks, 2014.

[220] E. Gelenbe, R. Lent, A. Montuori, and Z. Xu, "Cognitive packet networks: QoS and performance," in Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002. Proceedings. 10th IEEE International Symposium on, pp. 3–9, IEEE, 2002.

[221] M. Cordina and C. J. Debono, "Increasing wireless sensor network lifetime through the application of som neural networks," in Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on, pp. 467–471, IEEE, 2008.

[222] N. Enami and R. A. Moghadam, "Energy based clustering self organizing map protocol for extending wireless sensor networks lifetime and coverage," Canadian Journal on Multimedia and Wireless Networks, vol. 1, no. 4, pp. 42–54, 2010.

[223] L. Dehni, F. Krief, and Y. Bennani, "Power control and clustering in wireless sensor networks," in Challenges in Ad Hoc Networking, pp. 31–40, Springer, 2006.

[224] F. Oldewurtel and P. Mahonen, "Neural wireless sensor networks," in Systems and Networks Communications, 2006. ICSNC'06. International Conference on, pp. 28–28, IEEE, 2006.

[225] G. A. Barreto, J. Mota, L. G. Souza, R. A. Frota, and L. Aguayo, "Condition monitoring of 3G cellular networks through competitive neural models," Neural Networks, IEEE Transactions on, vol. 16, no. 5, pp. 1064–1075, 2005.

[226] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," Instrumentation and Measurement, IEEE Transactions on, vol. 57, no. 5, pp. 981–988, 2008.

[227] D. Hoang, R. Kumar, and S. Panda, "Fuzzy C-Means clustering protocol for wireless sensor networks," in Industrial Electronics (ISIE), 2010 IEEE International Symposium on, pp. 3477–3482, IEEE, 2010.

[228] E. I. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks," in Communications, 2004 IEEE International Conference on, vol. 6, pp. 3663–3667, IEEE, 2004.

[229] W. Zhang, S. K. Das, and Y. Liu, "A trust based framework for secure data aggregation in wireless sensor networks," in Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on, vol. 1, pp. 60–69, IEEE, 2006.

[230] G. Kapoor and K. Rajawat, "Outlier-aware cooperative spectrum sensing in cognitive radio networks," Physical Communication, vol. 17, pp. 118–127, 2015.

[231] T. Ristaniemi and J. Joutsensalo, "Advanced ICA-based receivers for block fading DS-CDMA channels," Signal Processing, vol. 82, no. 3, pp. 417–431, 2002.

[232] M. S. Mushtaq, B. Augustin, and A. Mellouk, "Empirical study based on machine learning approach to assess the QoS/QoE correlation," in Networks and Optical Communications (NOC), 2012 17th European Conference on, pp. 1–7, IEEE, 2012.

[233] M. Alreshoodi and J. Woods, "Survey on QoE\ QoS CORRELATION Models for Multimedia Services," International Journal of Distributed and Parallel Systems, vol. 4, no. 3, p. 53, 2013.

[234] A. Testolin, M. Zanforlin, M. De Filippo De Grazia, D. Munaretto, A. Zanella, and M. Zorzi, "A machine learning approach to QoE-based video admission control and resource allocation in wireless systems," in Ad Hoc Networking Workshop (MED-HOC-NET), 2014 13th Annual Mediterranean, pp. 31–38, IEEE, 2014.

[235] S. Przylucki, "Assessment of the QoE in Voice Services Based on the Self-Organizing Neural Network Structure," in Computer Networks, pp. 144–153, Springer, 2011.

[236] P. Ahammad, B. Kennedy, P. Ganti, and H. Kolam, "QoE-driven Unsupervised Image Categorization for Optimized Web Delivery: Short Paper," in Proceedings of the ACM International Conference on Multimedia, pp. 797–800, ACM, 2014.

[237] D. C. Mocanu, G. Exarchakos, and A. Liotta, "Deep learning for objective quality assessment of 3D images," in Image Processing (ICIP), 2014 IEEE International Conference on, pp. 758–762, IEEE, 2014.

[238] B. Francisco, A. Ramon, P.-R. Jordi, and S. Oriol, "Distributed spectrum management based on reinforcement learning," in 14th International Conference on Cognitive Radio Oriented Wireless Networks and Communications, pp. 1–6, IEEE, 2009.

[239] L. Xuedong, C. Min, X. Yang, B. LLangko, and L. Victo C. M., "MRL-CC: a novel cooperative communication protocol for QoS provisioning in wireless sensor networks," International Journal of Sensor Networks, vol. 8, no. 2, pp. 98–108, 2010.

[240] P. Geurts, I. El Khayat, and G. Leduc, "A machine learning approach to improve congestion control over wireless computer networks," in Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on, pp. 383–386, IEEE, 2004.

[241] K.-S. Hwang, S.-W. Tan, M.-C. Hsiao, and C.-S. Wu, "Cooperative multiagent congestion control for high-speed networks," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 35, no. 2, pp. 255–268, 2005.

[242] K. Winstein and H. Balakrishnan, "Tcp ex machina: computer-generated congestion control," in ACM SIGCOMM Computer Communication Review, vol. 43, pp. 123–134, ACM, 2013.

[243] T. J. O'Shea and J. Hoydis, "An introduction to machine learning communications systems," arXiv preprint arXiv:1702.00832, 2017.

[244] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," arXiv preprint arXiv:1707.07980, 2017.

[245] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Unsupervised representation learning of structured radio communication signals," in Sensing, Processing and Learning for Intelligent Machines (SPLINE), 2016 First International Workshop on, pp. 1–5, IEEE, 2016.

[246] T. Huang, H. Sethu, and N. Kandasamy, "A new approach to dimensionality reduction for anomaly detection in data traffic," IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 651–665, 2016.

[247] A. Zoha, A. Saeed, A. Imran, M. A. Imran, and A. Abu-Dayya, "A learning-based approach for autonomous outage detection and coverage optimization," Transactions on Emerging Telecommunications Technologies, vol. 27, no. 3, pp. 439–450, 2016.

[248] A. Shirazinia and S. Dey, "Power-constrained sparse gaussian linear dimensionality reduction over noisy channels," IEEE Transactions on Signal Processing, vol. 63, no. 21, pp. 5837–5852, 2015.

[249] S. Hou, R. C. Qiu, Z. Chen, and Z. Hu, "Svm and dimensionality reduction in cognitive radio with experimental validation," arXiv preprint arXiv:1106.2325, 2011.

[250] C. Khalid, E. Zyad, and B. Mohammed, "Network intrusion detection system using L1-norm PCA," in Information Assurance and Security (IAS), 2015 11th International Conference on, pp. 118–122, IEEE, 2015.

[251] E. Goodman, J. Ingram, S. Martin, and D. Grunwald, "Using bipartite anomaly features for cyber security applications," in Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on, pp. 301–306, IEEE, 2015.

[252] N. Patwari, A. O. Hero III, and A. Pacholski, "Manifold learning visualization of network traffic data," in Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data, pp. 191–196, ACM, 2005.

[253] D. López-Sánchez, A. G. Arrieta, and J. M. Corchado, "Deep neural networks and transfer learning applied to multimedia web mining," in Distributed Computing and Artificial Intelligence, 14th International Conference, vol. 620, p. 124, Springer, 2018.

[254] T. Ban, S. Pang, M. Eto, D. Inoue, K. Nakao, and R. Huang, "Towards early detection of novel attack patterns through the lens of a large-scale darknet," in Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences, pp. 341–349, IEEE, 2016.

[255] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based DDoS detection system in software-defined networking (SDN)," arXiv preprint arXiv:1611.07400, 2016.

[256] C. G. Cordero, S. Hauke, M. Mühlhäuser, and M. Fischer, "Analyzing flow-based anomaly intrusion detection using replicator neural networks," in Privacy, Security and Trust (PST), 2016 14th Annual Conference on, pp. 317–324, IEEE, 2016.

[257] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "A novel anomaly detection system using feature-based MSPCA with sketch," in Wireless and Optical Communication Conference (WOCC), 2017 26th, pp. 1–6, IEEE, 2017.

[258] T. Matsuda, T. Morita, T. Kudo, and T. Takine, "Traffic anomaly detection based on robust principal component analysis using periodic traffic behavior," IEICE Transactions on Communications, vol. 100, no. 5, pp. 749–761, 2017.

[259] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of PCA and optimized SVM," in Contemporary Computing and Informatics (IC3I), 2014 International Conference on, pp. 879–884, IEEE, 2014.

[260] B. Subba, S. Biswas, and S. Karmakar, "Enhancing performance of anomaly based intrusion detection systems through dimensionality reduction using principal component analysis," in Advanced Networks and Telecommunications Systems (ANTS), 2016 IEEE International Conference on, pp. 1–6, IEEE, 2016.

[261] I. Z. Muttaqien and T. Ahmad, "Increasing performance of IDS by selecting and transforming features," in Communication, Networks and Satellite (COMNETSAT), 2016 IEEE International Conference on, pp. 85–90, IEEE, 2016.

[262] N. Y. Almusallam, Z. Tari, P. Bertok, and A. Y. Zomaya, "Dimensionality reduction for intrusion detection systems in multi-data streamsâĂŤa review and proposal of unsupervised feature selection scheme," in Emergent Computation, pp. 467–487, Springer, 2017.

[263] Y. Kumar, H. Farooq, and A. Imran, "Fault prediction and reliability analysis in a real cellular network," in Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International, pp. 1090–1095, IEEE, 2017.

[264] Z. Nascimento, D. Sadok, S. Fernandes, and J. Kelner, "Multi-objective optimization of a hybrid model for network traffic classification by combining machine learning techniques," in Neural Networks (IJCNN), 2014 International Joint Conference on, pp. 2116–2122, IEEE, 2014.

[265] Z. Ansari, M. Azeem, A. V. Babu, and W. Ahmed, "A fuzzy approach for feature evaluation and dimensionality reduction to improve the quality of web usage mining results," arXiv preprint arXiv:1509.00690, 2015.

[266] M. A. Alsheikh, S. Lin, H.-P. Tan, and D. Niyato, "Toward a robust sparse data representation for wireless sensor networks," in Local Computer Networks (LCN), 2015 IEEE 40th Conference on, pp. 117–124, IEEE, 2015.

[267] K. Labib and V. R. Vemuri, "An application of principal component analysis to the detection and visualization of computer network attacks," Annals of telecommunications, vol. 61, no. 1, pp. 218–234, 2006.

[268] J. Lokoč, J. Kohout, P. Čech, T. Skopal, and T. Pevnỳ, "k-NN classification of malware in HTTPS traffic using the metric space approach," in Pacific-Asia Workshop on Intelligence and Security Informatics, pp. 131–145, Springer, 2016.

[269] M. Ancona, W. Cazzola, S. Drago, and G. Quercini, "Visualizing and managing network topologies via rectangular dualization," in Computers and Communications, 2006. ISCC'06. Proceedings. 11th IEEE Symposium on, pp. 1000–1005, IEEE, 2006.

[270] G. Cherubin, I. Nouretdinov, A. Gammerman, R. Jordaney, Z. Wang, D. Papini, and L. Cavallaro, "Conformal clustering and its application to botnet traffic.," in SLDS, pp. 313–322, 2015.

[271] I. Marsh, "A lightweight measurement platform for home internet monitoring," http://cheesepi.sics.se/Publications/mmsys2017.pdf.

[272] J. M. Lewis, V. R. De Sa, and L. Van Der Maaten, "Divvy: fast and intuitive exploratory data analysis," The Journal of Machine Learning Research, vol. 14, no. 1, pp. 3159–3163, 2013.

[273] G. Holmes, A. Donkin, and I. H. Witten, "Weka: A machine learning workbench," in Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on, pp. 357–361, IEEE, 1994.

[274] Q. Liao and S. Stanczak, "Network state awareness and proactive anomaly detection in self-organizing networks," in Globecom Workshops (GC Wkshps), 2015 IEEE, pp. 1–6, IEEE, 2015.

[275] S. Chernov, D. Petrov, and T. Ristaniemi, "Location accuracy impact on cell outage detection in lte-a networks," in Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International, pp. 1162–1167, IEEE, 2015.

[276] A. Zoha, A. Saeed, A. Imran, M. A. Imran, and A. Abu-Dayya, "Data-driven analytics for automated cell outage detection in self-organizing networks," in Design of Reliable Communication Networks (DRCN), 2015 11th International Conference on the, pp. 203–210, IEEE, 2015.

[277] J. Turkka, F. Chernogorov, K. Brigatti, T. Ristaniemi, and J. Lempiäinen, "An approach for network outage detection from drive-testing databases," Journal of Computer Networks and Communications, vol. 2012, 2012.

[278] N. Kato, Z. M. Fadlullah, B. Mao, F. Tang, O. Akashi, T. Inoue, and K. Mizutani, "The deep learning vision for heterogeneous network traffic control: proposal, challenges, and future perspective," IEEE Wireless Communications, vol. 24, no. 3, pp. 146–153, 2017.

[279] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "Routing or computing? the paradigm shift towards intelligent computer network packet transmission based on deep learning," IEEE Transactions on Computers, 2017.

[280] J. Qadir, N. Ahad, E. Mushtaq, and M. Bilal, "SDNs, clouds, and big data: new opportunities," in Frontiers of Information Technology (FIT), 2014 12th International Conference on, pp. 28–33, IEEE, 2014.

[281] B. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," Communications Surveys & Tutorials, IEEE, vol. 16, no. 3, pp. 1617–1634, 2014.

[282] H. Kim and N. Feamster, "Improving network management with software defined networking," Communications Magazine, IEEE, vol. 51, no. 2, pp. 114–119, 2013.

[283] J. Ashraf and S. Latif, "Handling Intrusion and DDoS attacks in Software Defined Networks using Machine Learning Techniques," in Software Engineering Conference (NSEC), 2014 National, pp. 55–60, IEEE, 2014.

[284] D. J. Dean, H. Nguyen, and X. Gu, "Ubl: unsupervised behavior learning for predicting performance anomalies in virtualized cloud systems," in Proceedings of the 9th international conference on Autonomic computing, pp. 191–200, ACM, 2012.

[285] D. He, S. Chan, X. Ni, and M. Guizani, "Software-defined-networking-enabled traffic anomaly detection and mitigation," IEEE Internet of Things Journal, 2017.

[286] K. K. Goswami, "Intelligent threat-aware response system in software-defined networks," http://scholarworks.sjsu.edu/etd_theses/4801/, 2017.

[287] A. S. da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, "ATLANTIC: a framework for anomaly traffic detection, classification, and mitigation in SDN," in Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP, pp. 27–35, IEEE, 2016.

[288] S.-h. Zhang, X.-x. Meng, and L.-h. Wang, "SDNForensics: A comprehensive forensics framework for software defined network," Development, vol. 3, no. 4, p. 5, 2017.

[289] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, "Machine learning in software defined networks: Data collection and traffic classification," in Network Protocols (ICNP), 2016 IEEE 24th International Conference on, pp. 1–5, IEEE, 2016.

[290] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 336–361, 2013.

[291] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5g: how to empower son with big data for enabling 5g," IEEE Network, vol. 28, no. 6, pp. 27–33, 2014.

[292] X. Wang, X. Li, and V. C. Leung, "Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges," IEEE Access, vol. 3, pp. 1379–1391, 2015.

[293] A. Misra and K. K. Sarma, "Self-organization and optimization in heterogenous networks," in Interference Mitigation and Energy Management in 5G Heterogeneous Cellular Networks, pp. 246–268, IGI Global, 2017.

[294] Z. Zhang, K. Long, J. Wang, and F. Dressler, "On swarm intelligence inspired self-organized networking: its bionic mechanisms, designing principles and optimization approaches," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 513–537, 2014.

[295] S. Latif, J. Qadir, S. Farooq, and M. A. Imran, "How 5g wireless (and concomitant technologies) will revolutionize healthcare?," Future Internet, vol. 9, no. 4, p. 93, 2017.

[296] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for Internet of things services," IEEE Internet Computing, vol. 21, no. 2, pp. 16–24, 2017.

[297] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[298] H.-Y. Kim and J.-M. Kim, "A load balancing scheme based on deep-learning in IoT," Cluster Computing, vol. 20, no. 1, pp. 873–878, 2017.

[299] D. Puschmann, P. Barnaghi, and R. Tafazolli, "Adaptive clustering for dynamic IoT data streams," IEEE Internet of Things Journal, vol. 4, no. 1, pp. 64–74, 2017.

[300] H. Assem, L. Xu, T. S. Buda, and D. OâĂŹSullivan, "Machine learning as a service for enabling internet of things and people," Personal and Ubiquitous Computing, vol. 20, no. 6, pp. 899–914, 2016.

[301] J. Lee, M. Stanley, A. Spanias, and C. Tepedelenlioglu, "Integrating machine learning in embedded sensor systems for internet-of-things applications," in Signal Processing and Information Technology (ISSPIT), 2016 IEEE International Symposium on, pp. 290–294, IEEE, 2016.

[302] P. Lin, D.-C. Lyu, F. Chen, S.-S. Wang, and Y. Tsao, "Multi-style learning with denoising autoencoders for acoustic modeling in the internet of things (IoT)," Computer Speech & Language, 2017.
[303] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Security and Privacy (SP), 2010 IEEE Symposium on, pp. 305–316, IEEE, 2010.
[304] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," Information Sciences, vol. 378, pp. 484–497, 2017.
[305] L. Watkins, S. Beck, J. Zook, A. Buczak, J. Chavis, W. H. Robinson, J. A. Morales, and S. Mishra, "Using semi-supervised machine learning to address the big data problem in DNS networks," in Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual, pp. 1–6, IEEE, 2017.
[306] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2010.
[307] E. Başтuğ, M. Bennis, and M. Debbah, "A transfer learning approach for cache-enabled wireless networks," in Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on, pp. 161–166, IEEE, 2015.
[308] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," arXiv preprint arXiv:1610.05492, 2016.
[309] A. Gokhale and A. Bhagwat, "System and method for network address administration and management in federated cloud computing networks," May 30 2017. US Patent 9,667,486.
[310] J. H. Abawajy and M. M. Hassan, "Federated internet of things and cloud computing pervasive patient health monitoring system," IEEE Communications Magazine, vol. 55, no. 1, pp. 48–53, 2017.
[311] P. Massonet, L. Deru, A. Achour, S. Dupont, A. Levin, and M. Villari, "End-to-end security architecture for federated cloud and IoT networks," in Smart Computing (SMARTCOMP), 2017 IEEE International Conference on, pp. 1–6, IEEE, 2017.
[312] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, pp. 2672–2680, 2014.
[313] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," arXiv preprint arXiv:1702.05983, 2017.
[314] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," arXiv preprint arXiv:1606.04435, 2016.
[315] L. Breiman, "Statistical modeling: The two cultures (with comments and a rejoinder by the author)," Statistical science, vol. 16, no. 3, pp. 199–231, 2001.
[316] I. Sturm, S. Lapuschkin, W. Samek, and K.-R. Müller, "Interpretable deep neural networks for single-trial eeg classification," Journal of neuroscience methods, vol. 274, pp. 141–145, 2016.
[317] X. Zhu, C. Vondrick, C. C. Fowlkes, and D. Ramanan, "Do we need more training data?," International Journal of Computer Vision, vol. 119, no. 1, pp. 76–92, 2016.
[318] P. Domingos, "A few useful things to know about machine learning," Communications of the ACM, vol. 55, no. 10, pp. 78–87, 2012.
[319] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, "Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study," IEEE Access, vol. 4, pp. 7940–7957, 2016.
[320] G. P. Zhang, "Avoiding pitfalls in neural network research," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 37, no. 1, pp. 3–16, 2007.
[321] A. Ng, "Advice for applying machine learning," Stanford University, http://cs229.stanford.edu/materials/ML-advice.pdf, 2011.